RESEARCH ON SERVICE DISCOVERY BASED ON DHT FOR CLOUD MANUFACTURING SYSTEM

Yifei TONG[1], Xinye WU, Qian WANG, Dongbo LI

In order to achieve resource sharing of cloud manufacturing system, it is very important for users to discover the resource service connected into the cloud system through different infrastructures. To aim this, the service discovery mechanism of cloud manufacturing is studied. Firstly, cloud services/resources are classified and analyzed. Secondly, to efficiently discover the cloud service according to the users' requirements of cloud manufacturing system, a layered resource service model for cloud manufacturing with convenient, efficient and intelligent resource discovery mechanism, is proposed based on distributed hash table. The proposed mechanism deploys the idea of "grouping resource service" by the similarity of resource attributes. Case study and analysis demonstrate that the proposed service discovery approach for cloud manufacturing is feasible and effective. The research can favor the location and discovery of resource services form the distributed cloud manufacturing system.

Keywords: Cloud manufacturing, Resource service, Distributed hash table, Service discovery algorithm, Resources grouping

1. Introduction

In order to achieve the resource sharing of cloud manufacturing system, it is very important for users to discover the resource service [1]. Due to the autonomy, distributed and dynamic manufacturing resources, the service location and discovery of service resources in the cloud manufacturing environment are facing the challenges. At present, great deals of researches have made to focus on this issue. A solution by applying Case-based Reasoning was proposed for web service discovery and selection, in which the resemblance between a pair of cases is quantified through a similarity function by introducing a novel case representation, learning heuristics and three different similarity functions [2]. Another method of service discovery for cloud manufacturing was proposed based on Ontology Web

[1] Nanjing University of Science and Technology, China, email: tyf51129@aliyun.com

Language for Service (OWL-S) with semantic elements integration and tree-like structured basic attributes and quality attributes of web services [3]. Gaur et al. proposed a heuristic based approach using hybrid Genetic and Tabu algorithm to determine the Optimal Quality of Service (QoS) based service composition [4]. An agility-oriented and fuzziness-embedded ontology model, adopting agility-centric design along with OWL2 (Web Ontology Language) fuzzy extensions was proposed with enabling comprehensive service specification and a service recommendation system was developed based on the proposed model as a knowledge base [5]. Ma et al. proposed a Contextual Service Discovery (CSD) approach to help find out qualified services in accordance with binding context on the user side, considering the descriptions and binding information as a set of meaningful terms [6]. With the proliferation of Software-as-a-Service (SaaS) in the cloud environment, it is difficult for users to search for the right service that satisfies all their needs. Afify et al. presented a comprehensive survey on cloud services discovery and selection research approaches, based on which proposed a semantic-based SaaS publication, discovery, and selection system, which combines services domain knowledge, SaaS characteristics, QoS metrics, and real SaaS offers [7].

Considering the requirements of the autonomy, robust and dynamic aspects of the cloud manufacturing system, the P2P (peer-to-peer) structure is used to discover the cloud resources [8]. DHT (distributed hash table) has been widely studied and applied for P2P. As for resource service location and discovery, it is found that the research on DHT is generally based on the matching of key words. However, considering the characteristics of cloud manufacturing, the location and discovery of cloud services should be based on resource attributes. In order to better organize cloud resource service and support the users' complex requirements, a hierarchical cloud manufacturing resource service organization model based on DHT is put forward as well as multi-attributes based cloud manufacturing resource service discovery algorithm.

2. Description of cloud manufacturing resource

2.1. Definition and classification of manufacturing resources

In the cloud manufacturing environment, the manufacturing resources connected into cloud system through different infrastructures can be classified into 7 types, as shown in Fig. 1:

- Software resources.
- Equipment resources: physical equipments with specific processing capacity during the product life cycle.
- Industry resources: the field knowledge related to manufacturing industry or products manufacturing.
- Manufacturing capacity.
- Material resources: the materials needed to produce a manufacturing.
- Human resources: the personnel throughout the entire life cycle of manufacturing products.
- Logistics resources: the transportation of the geographically distributed manufacturing resources in the cloud manufacturing environment.

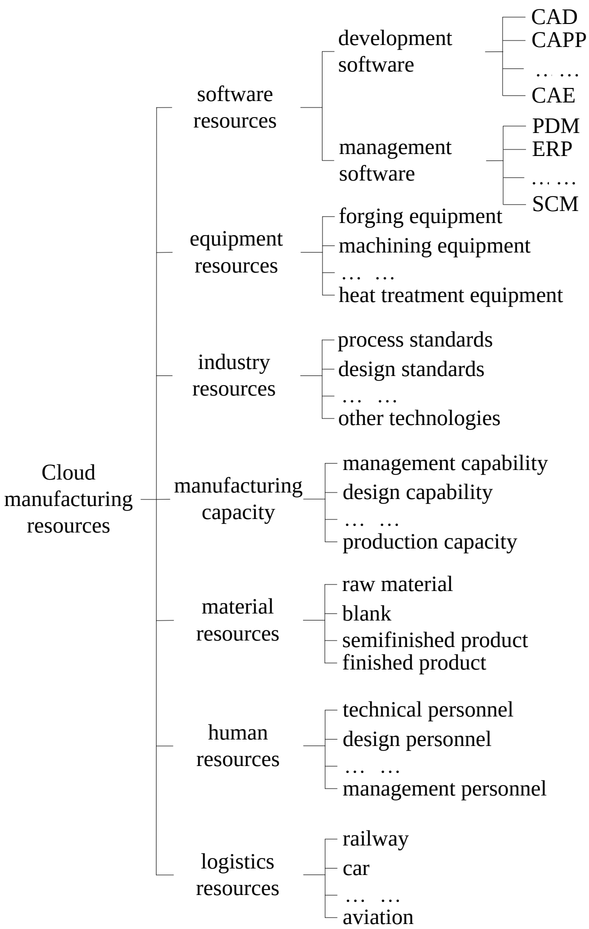

Fig. 1 Manufacturing resource classification

2.2. Manufacturing resource description

The description of the manufacturing resources provides important information for the discovery of cloud manufacturing services. In the cloud manufacturing environment, the manufacturing resources are massive, highly integrated, dynamic and transparent. So, a resource description mechanism is necessary to simplify and standardize manufacturing resource description for unified representation.

The unified attribute description model of the manufacturing resources in cloud manufacturing system can be abstracted as shown in Fig. 2. The description model can be extended to describe the attributes of the specific manufacturing resources.

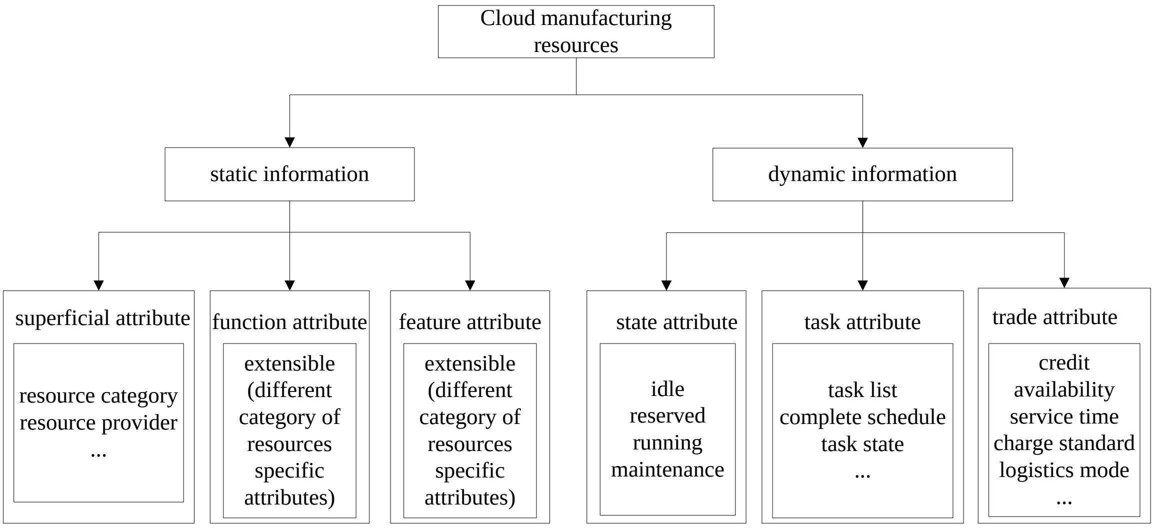

Fig. 2 Manufacturing resource description model

The description information model mainly includes the following attributes:

- Superficial attribute.
- Function attribute: functional parameters of the manufacturing resources, one of the important references for resources/services selection.
- Feature attribute: most concerned by users of cloud manufacturing system about the ability of forge resource services, as the key of resource discovery.
- State attribute: the status of cloud resources/services, an important reference for manufacturing resource optimization and scheduling.

- Task attribute: a detailed description of the task information submitted to the cloud manufacturing system.
- Trade attribute: transaction index of resources/services in cloud manufacturing system.

The users of cloud manufacturing system should package the description information of manufacturing resources according to the specification, which will be abstracted as a service description template based on the description of XML Schema. The manufacturing resources/services description template is as follows:

```
<xsd:element name="CFgResource">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="SuperficialAttribute" type="fr:SuperficialAttributeType"/>
<xsd:element name="FunctionAttribute" type="fr:FunctionAttributeType"/>
<xsd:element name="FeatureAttribute" type="fr:FeatureAttributeType"/>
<xsd:element name="StateAttribute" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:complexType name="SuperficialAttributeType">
<xsd:sequence>
<xsd:element name="identifyID" type="xsd:string"/>
<xsd:element name="name" type="xsd:string"/>
<xsd:element name="provider" type="xsd:string"/>
<xsd:element name="type" type="xsd:string"/>
<xsd:element name="location" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="FunctionAttributeType">
<xsd:sequence>
<!--extended here-- >
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name=" Feature AttributeType">
```

```
<xsd:sequence>
<!--extended here-- >
</xsd:sequence>
</xsd:complexType>.
```

3. Cloud manufacturing resource/service discovery based on DHT

3.1. Definitions and the organizational model

Definition 1: Manufacturing Resource Service (MRS)–logic manufacturing resources published to cloud manufacturing system after virtualization and encapsulation.

Definition 2: Manufacturing Service Host Node (MSHN)–the basic unit of cloud manufacturing system

Definition 3: Manufacturing Service Login Node (MSLN)–the entrance of manufacturing resources registration and the query to manage multiple MSHN by communicating with other MSLNs to realize the registration, discovery and maintenance of MRS, et al.

Definition 4: Autonomous Service Domain (ASD)–the manufacturing resource service domain, which is composed of every MSLN and MSHN under the MSLN.

The organization model of cloud manufacturing resources service is proposed. It is a three–layer hierarchical architecture as shown in Fig.3. The bottom layer is manufacturing resources service layer, which is made up of all kinds of MRS. The middle layer is composed of some ASD. Every Service Domain's MSLN and backup nodes are selected according to the load capacity, the availability and other index. In case of MSLN failure, the backup nodes will monitor the MSLN during the service life cycle and record its operations. When the MSLN fails, the backup node will take place of the MSLN and synchronously update the data. Meanwhile, a new backup node will be chosen. The upper layer is the overlay network consisting of many MSLNs by interconnection based on DHT [9]. In this paper the Chord agreement is adopted [10].

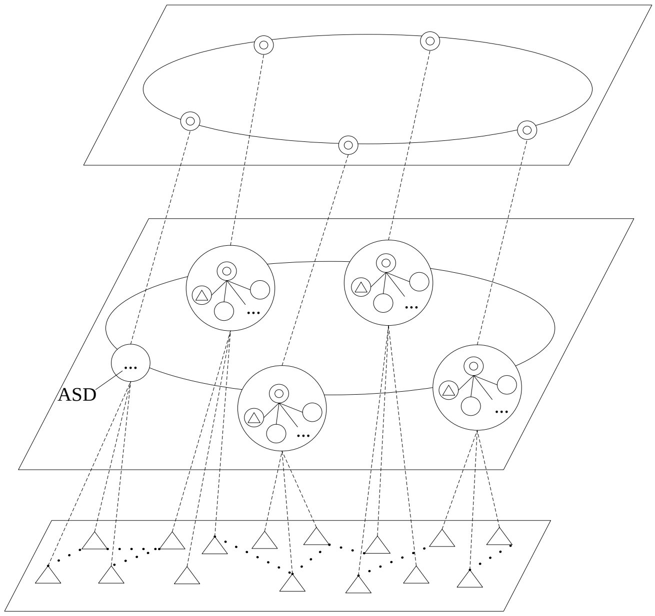

Fig. 3 The organization model of cloud manufacturing resources service based on DHT

3.2. Manufacturing resource service grouping based on DHT

3.2.1. Service grouping based on DHT

Under the cloud manufacturing environment, the manufacturing resources service with similar type should be stored in the same autonomous service domain as much as possible for convenient and efficient service discovery. Traditionally, the similarity between two attributes is determined by the distance between the two attribute vectors, which is feasible for small amounts of attributes, but not efficient for manufacturing resource with massive attributes under the cloud manufacturing environment. Another method based on traditional hash, cannot output the same or similar hash value with input of the same or similar contents. Therefore, the locality sensitive hash (LSH) is adopted here [11]. LSH can map the similar MRS to the same or similar resource service domain. Simhash is one of the LSH, which can output the similar hash mapping signature for input of similar texts.

Fig. 4 shows an annular identity space whose size is 2P (P = 160 in this paper). The identity space is generated by the mapping of nodes and manufacturing resource service. The SHA-1 hash function is utilized to map the IP address of MSLN to the identity space. Every MSLN manages a routing table which has p items to locate the manufacturing resources. The similar MRS will be mapped to the same identity space through Simhash. In this research, Chord agreement is adopted to make mapping between resource services and nodes. The nodes are sorted by the clockwise descending [12].

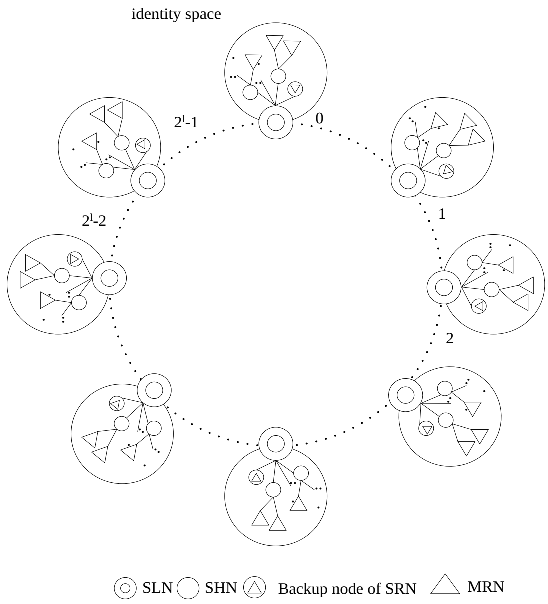

Fig. 4 The grouping structure of DHT

3.2.2. Manufacturing resource services grouping based on DHT

Each attribute of MRS is described as <*attribute category*, *attribute value*>. Take "Heat treatment services" as example to be described by the attributes set as:{*service name*=heat treatment, *craft type*=quenching, *single weight*=5T, *service Duration*=one month, …}. There are plenty of manufacturing resource categories in the cloud manufacturing system and each resource service has a lot of properties. If all attributes of every manufacturing resource are included, it will increase the index and maintenance cost of cloud manufacturing services. However, users usually focus on part of the attributes as the most typical feature attributes of manufacturing resource service. For example, as for heat treatment outsourcing service, the name of the resource service, the technological types, service quality and single weight are often selected as the feature attributes. In the actual situation, the feature attributes of manufacturing resource service can be flexibly selected and defined. The manufacturing resource service grouping is the process of MRS registering into the MSLN.

Provided that *n* selected attributes of *MRS Attr*[ ]={*attr*1, *attr*2,…, *attrn*}, the grouping algorithm of MRS is described as below:

Step1. The MRS of node m initiates request registration and traverse the n attributes of MRS. Then calculate the Simhash values of the *Attr*[ ]. Next search the MSLN to deposit the MRS index and deploy the MRS.

Step1.1. Locate the MSLN which can deposit the i^{th} feature attribute. Check the Simhash(*attr*[*i*]) if it is within SHA-1(*m*) and SHA-1(*m*. *successor*). If so, end the search and the *successor* of *m* is the target MSLN to deposit the feature attribute, and then go to step1.2; Otherwise, according to the Chord agreement, find out the *successor* of *m* nearest to the Simhash(*attr*[*i*]) and less than Simhash(*attr*[*i*]) in the finger table of node *m*. Then forward the search request to the node. Repeat the above process to recursive query until the node of *attr*[*i*].*successor* is found.

Step1.2. Locate the MSLN which can deploy the i^{th} feature attribute. The condition is that the number of marked MSHN's in the service domain in MSLN's charge is less than the number (*w*) of MSHN not marked. If in Step1.1 the MSLN which can deposit the *attr*[*i*] has been found can satisfy the above conditions, mark this MSLN as the deployable node.

Step2. Deploy this feature attribute to the MSHN of ASD, where the deployable MSHN locates.

Step3. Forward the registration information of this feature attribute to the MSLN and then update the MRS index table of the MSLN.

The core idea of the MRS grouping are that:①the index of every MRS will exist in n registered MSLNs.②A MRS can be deployed to w MSHNs at most, and the MSHN still has the ability to accept new MRS deployment.

3.3. Resource service discovery based on DHT for cloud manufacturing

The overlay network based on DHT is a dynamic organization, in which the nodes and manufacturing resources can join and exit at any time as a dynamic process. In this case, the nodes must dynamically update the hash table. The nodes for connection into the overlay network can be divided into MSLN and MSHN. The node joining algorithm of Chord protocol is used for MSLN into the overlay network. As for MSHN, it need send a joining request the nearest MSLN. After successful connection to MSLN, MSLN update MSHN table in its charge.

Similarly, the node exit is divided into proactive or forced MSLN and MSHN. When the MSHN is proactive to exit, the MSLN of its group updates the management table and the MRS index table, and the MRS deployed in the group re-register. This will have no effect on the routing information of the MSLN in the group, and can also significantly reduce the impact of frequent node exit on the network. MSHN can be monitored and updated when the MSHN is forced to exit by the abnormal node. When the MSLN is proactive to exit, the FSHN in the MSLN service domain and the service deployed in the MSLN service domain are delivered to the MSLN's backup node, and update the MSHN management table and the MRS index table. The state of MSLN will be monitored and recorded by its backup nodes. If abnormal MSLN is detected, the backup node can take the place of MSLN and synchronize the data. At the same time, the new backup node is selected. This maintenance process not only has no effect on the topology of the overlay network, but also limits the impact within the service domain.

The maintenance of the manufacturing resources service is also divided into the registration and exit of MRS. Wherein, the registration of MRS has been mentioned above. When exiting MRS, send the exit request to its MSLN of the ASD. After receiving this request, the MSLN respond with the "deleting the MRS's index" information and the deployment information in the MSHN. Meanwhile, notify the other MSLN to update.

Based on the layered organization of cloud resource service and the resources grouping, a discovery algorithm based on multi-attributes is proposed as follows:

1)func search(*MRS*，*m*) // the searching request and algorithm of MRS on node *M*

2)for *i*=1;*i*<=*n*;*i*++ //*n* feature attributes for searching

3)set[*i*] = chord_search(*MRS_attr*[*i*],*m*)； //A set of *n* feature attributes searching

4)end func

5)func chord_search(*MRS_attr*,*m*)：Result //According to the hash value of *attr* to locate the target node MSLN

6)if SHA-1(*m.successor*)>=Simhash(*MRS_ attr*)>SHA-1(*m*)

7)*targetNode*=*m.successor*； // *m.successor* is the target node

8)if *attr* exist in *targetNode*'s cache //Search the MRS index in the target node cache

9)return *MR_index*； // Return resource services index

10)else

Find the *MR_index* by hamming_distance(Simhash(*MRS_ attr*), Simhash(*MSHN_ attr*))； //Hamming distance is calculated by the Simhash value of MRS feature attribute and target node management of resource service

11)if hamming_distance(Simhash(*MRS_ attr*), Simhash(*MSHN_ attr*)) <=*q* //Similarity matching

Return *MR_index* //Return MRS index

12)else

return can't search suitable *MR_index*!；

13)else

according to chord routing protocol, search *m_next* which is the nearest and less than

simhash(*MRS_attr*) from *m*'s finger_table；

chord_search(*MRS*_attr, *m_next*)； //Recursive query

14)end func

The above algorithm can be considered as two steps to complete. Firstly locate the target MSLN for the MRS on Chord ring rapidly, and then carry out MRS matching based on the similarity within the resource service domain of the target MSLN. In the actual manufacturing business, with the frequent business cooperation between enterprises and their dynamic alliance partners, the cooperation will gradually stabilize. Therefore, if a cloud manufacturing business is successfully completed, the system will retrieve the information. If a same manufacturing business next time is issued, the cache history can be directly called for user with reduced retrieve time. So, each MSLN is not only responsible for the registration and retrieval of MRS , but also need to maintain the related cache of its node.

3.4. Case study

Due to the limited device capabilities and production capacity, a manufacturing company(DVR) in Nanjing often need to complete manufacturing tasks by outsourcing. The outsourcing task can be issued through the cloud manufacturing system, and the most concerned feature properties are: *service name*=heat treatment, *craft type*=quenching, *single weight*=5T, *service duration*=one month.

Here, take the feature attributes search of "*service name* = heat treatment", denoted as attr, as example to illustrate the discovery process of manufacturing resource services. The search of other features attributes are the same. If the target cloud manufacturing service is in the N56 group, the specific searching process is as follows:

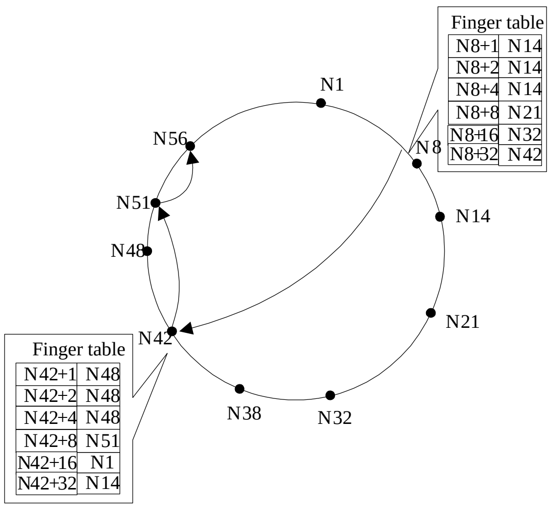

Fig. 5 The search process of cloud manufacturing resource services

- Fig. 5 shows a ring composed of p=6, 10 nodes, and there are 10 nodes on the ring. The node N8 sends the searching request of manufacturing resource service, and locates the service domain of the attribute according to the routing table on the node N8.
- Firstly judge whether the successor node of N8 has the resources. N14 is the successor node of N8, whose Simhash (*attr*) does not conform to Simhash(*attr*)ϵ(SHA-1(N8); SHA-1(N14)).Then, start the searching from far to near in the routing table of N8.
- N42 is farthest in the routing table of N8, which can meet the SHA-1(N14) ϵ (SHA-1(N8); Simhash(*attr*)). It indicates that the N42 node is closest to the location of the service domain. Then jump to N42 to continue searching.
- N48 is the successor node of N42, whose Simhash(*attr*) does not conform to Simhash(*attr*)ϵ(SHA-1(N42); SHA-1(N48)). It indicates that there is no satisfactory resource service in the N48 group. So, continue to search in the routing table of N42.
- Then search from far and near, and find the node N51 can satisfy SHA-1(N51)ϵ(SHA-1(N42); Simhash(*attr*)),which indicates that the N51 node is closest to the location of the service domain. Then jump to N51 to continue searching.
- N56 as the successor node of N51, can satisfy Simhash(*attr*)ϵ(SHA-1(N51); SHA-1(N56)]. Until now, the location is finished, and the service domain in N56's charge is the service domain that DVR wants.
- After locating the resources service domain, search the similar cloud manufacturing services in the manufacturing resource service table in N56's charge. Make XOR operation of the Simhash(*attr*) and the Simhash (*attr*') of the manufacturing resource service registered in the node N56, and the number of 1 is the Hamming distance. The Hamming distance indicates the similarity degree of the two resources services. If the Hamming distance is less than a threshold q, it is determined that the two are similar and can satisfy the search requirements.
- N56 returns to the satisfactory resource service index, to obtain a set of alternative manufacturing resource services.

4. Algorithm performance analysis and comparison

According to references [13]and[14], the performance of the manufacturing resources service discovery algorithm is analyzed from the view of the search delay and message overhead without considerations for resource services failure: (1) search delay: in the worst cases, the hops from sending the request to finding the first manufacturing resource needed; (2) message overhead: in the worst case, the total number of messages generated during the period of from sending the request to finding the first manufacturing resource needed. Below are comparisons with some Resource Discovery Algorithms.

(1)Flooding: Flooding method is of great blindness, especially when determining the forwarding destination for resource service search request. Let Pflooding be the network diameter of two nodes in the overlay, and *n* be the number of information node. In the worst case, the flooding resource service discovery will traverse all the nodes of the overlay, so the search delay is $O(P_{flooding})$, and the message overhead is $O(|n|)$.

(2)Routing-Transfer(RT): With the implementation and status changing of resource services, the network overhead increases. Let PRT be the length of the acyclic longest path from sending the request to locating the target resource services. In the worst case, both the search latency and message overhead are $O(P_{RT})$.

(3)NEVRLATE: Resource information nodes are randomly grouped with the same number of nodes in each group. Resource services also randomly select nodes in certain group to register. In the worst case, the resource service request traverse all the nodes in all the groups. Let P'flooding be the network diameter of resource information node, and the search delay is $O(P'_{flooding})$, and the message overhead is $O(|n|/h)$.

(4)The proposed method proposed in this paper: All the manufacturing resource information nodes are arranged in a Chord ring from small to large to construct the DHT structure. Each resource is traversed and registered on the information node based on the Simhash value, and the manufacturing resource services are divided into multiple service domains. Assuming the DHT overlay network consists of n information nodes, and then in the worst case, the search delay of the algorithm is $O(log_2 n)$ of DHT search delay, and message overhead is $O(qlog_2 n)$ of the total number of messages for the q attributes.

Table 1

Comparison of the resource discovery methods

method	search latency	the message overhead
flooding	$O(P_{flooding})$	$O(\|n\|)$
RT	$O(P_{RT})$	$O(P_{RT})$
NEVRLATE	$O(P'_{flooding})$	$O(\|n\|/h)$
the proposed algorithm	$O(log_2 n)$	$O(qlog_2 n)$

PRT represents the length of the length of the acyclic longest path in the overlay network, so $P_{flooding}<P_{RT}$. The number of nodes in a group of NEVRLATE is certainly not greater than the total number of nodes. Therefore, under the same topology organizational protocol, $P'_{flooding}<P_{flooding}$. The proposed resource service discovery algorithm based on DHT is equivalent to binary search in the sequence table, so $O(log_2 n)<O(P'_{flooding}) <O(P_{flooding})<O(P_{RT})$ and $O(qlog_2 n)<O(|n|/h)<O(|n|)<O(P_{RT})$.

5. Conclusions

Based on the above analysis, it can be concluded that with the increase of resource services, resource service discovery performance will be better, but the overhead of the resource information update will increase. Different from the random grouping of NEVRLATE, the proposed resource service discovery algorithm proposes the grouping based on the Simhash value of the manufacturing resource service with semantic information integration, and can narrow down the service search range and the spread range of the manufacturing resource service update information. Therefore the performance of the proposed algorithm is superior than the other algorithms.

Acknowledgments

This work was financially supported by the National Defense Science and Technology Project Foundation (No. 0106142), and MOE (Ministry of Education in China) Youth Project of Humanities and Social Sciences (No. 17YJC630139). The supports are gratefully acknowledged.

Conflicts of Interest: The authors declare no conflict of interest.

REFERENCES

[1]. *B. H. Li*, *L. Zhang*, *L. Ren*, *et al.*, "Further discussion on cloud manufacturing", Computer Integrated Manufacturing Systems, vol. 17, no. 3, 2011, pp. 450-457.

[2]. *A. D. Renzis*, *M. Garriga*, *A. Flores*, *et al.*, "Case-based Reasoning for Web Service Discovery and Selection," Electronic Notes in Theoretical Computer Science, vol. 321, 2016, pp. 89-112, DOI:10.1016/j.entcs.2016.02.006

[3]. *H. Jiao*, *J. Zhang*, *J. H. Li*, *et al.*, "Research on cloud manufacturing service discovery based on latent semantic preference about OWL-S", International Journal of Computer Integrated Manufacturing, 2015, pp. 1-9, DOI: 10.1080/0951192X.2014.1002812

[4]. *V. Gaur*, *P. Dhyani*, *O. P. Rishi*, "A QoS-based service discovery using meta-heuristic optimization in federated cloud," International Conference on Image Information Processing, Solan, India, 21-25 Dec., 2015, IEEE Computer Society, pp. 174-178.

[5]. *D. Fang*, *X. Liu*, *I. Romdhani*, *et al.*, "An agility-oriented and fuzziness-embedded semantic model for collaborative cloud service search, retrieval and recommendation", Future Generation Computer Systems, vol. 56, 2016, pp. 11-26, DOI: 10.1016/j.future.2015.09.025

[6]. *S. P. Ma*, *C. W. Lan*, *C. H. Li*, "Contextual service discovery using term expansion and binding coverage analysis", Future Generation Computer Systems, vol. 48, no. C, 2015, pp. 73-81, DOI: 10.1016/j.future.2014.09.013

[7]. *Y. M. Afify*, *I. F. Moawad*, *N. L. Badr*, *et al.*, "Cloud Services Discovery and Selection: Survey and New Semantic-Based System", Intelligent Systems Reference Library, vol, 70, 2014, pp. 449-477, DOI: 10.1007/978-3-662-43616-5_17

[8]. *S. H. Chang*, *Z. R. Chen*, "Protecting Mobile Crowd Sensing against Sybil Attacks Using Cloud Based Trust Management System", Mobile Information Systems, vol. 2016, 2016, pp. 10, http://dx.doi.org/10.1155/2016/6506341

[9]. *P. Sendín-Raña*, *F. J. Gonzálezcastaño*, *F. Gómezcuba*, *et al.*, "Improving management performance of p2psip for mobile sensing in wireless overlays", Sensors, vol. 13, no. 11, 2013, pp. 15364-84, DOI:10.3390/s131115364

[10]. *I. Stoica*, *R. Morris*, *D. Karger*, *et al.*, "Chord: a scalable peer-to-peer lookup service for Internet applications", Proceeding of the 2001 Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, New York, USA, 27-31 Aug., 2001, ACM, pp. 149-160.

[11]. *N. Sundaram*, *A. Turmukhametova*, *N. Satish*, *et al.*, "Streaming Similarity Search over one Billion Tweets using Parallel Locality-Sensitive Hashing", Proceedings of the Vldb Endowment, vol. 6, no. 14, 2013, pp. 1930-1941, DOI: 10.14778/2556549.2556574

[12]. *Q. Zhang*, Research on Several Key Techniques of Cloud Manufacturing and its Applications, Master Thesis, South China University of Technology, Guangzhou, 2013.

[13]. *D. Q. Zhang*, *S. C. Chen*, "A comment on 'Alternative C-means clustering algorithms'", Pattern Recognition, vol. 37, no. 2, 2004, pp. 173-174, DOI: 10.1016/j.patcog.2003.08.001

[14]. *K. Vanthournout*, *G. Deconinck*, *R. Belmans*, "A Taxnomy for Resource Discovery", Personal & Ubiquitous Computing, vol. 9, no. 2, 2004, pp. 78-91, DOI: 10.1007/s00779-004-0312-9