PROCEEDINGS OF SCIENCE

Grid Resources Search Engine based on Ontology

Zhuang Miao[12]
College of Command Information Systems, PLA University of Science and Technology, 210007, Nanjing, China
E-mail: emiao_beyond@163.com

Yang Li[3]
College of Command Information Systems, PLA University of Science and Technology, 210007, Nanjing, China
E-mail: miipl606@163.com

Weiguang Xu
College of Command Information Systems, PLA University of Science and Technology, 210007, Nanjing, China
E-mail: miipl606@163.com

Jiabao Wang
College of Command Information Systems, PLA University of Science and Technology, 210007, Nanjing, China
E-mail: miipl606@163.com

Lei Song
College of Command Information Systems, PLA University of Science and Technology, 210007, Nanjing, China
E-mail: songlei@nudt.edu.cn

Jiang Xiao
College of Command Information Systems, PLA University of Science and Technology, 210007, Nanjing, China
E-mail: emiao_beyond@163.com

In grid condition, the traditional search engine can find few resources. It cannot meet the users' requirements. The grid resources cannot be utilized efficiently. A search engine system based on ontology of grid resources is hereby proposed. In this system, a database with ontology knowledge is designed to store all related conceptions, the relationship of grid domains and instances. Upon analyzing the semantic of users' queries, the expected resources can be retrieved more precisely and completely to satisfy their intentions.

CENet2015
12-13 September 2015
Shanghai, China

[1]Speaker
[2]Corresponding Author
[3]The authors are supported by the Provincial Nature Science Foundation of Jiangsu China BK2012512.

PoS(CENet2015)047

1. Introduction

The grid is an emerging technique for resources sharing and coordination in dynamic multi-institutional virtual organizations [1, 2]. The grids are used to join various geographically distributed computational resources and data resources while delivering such resources to heterogeneous user communities. These resources, may belonging to different institutions, have different usages and pose different requirements; however, the applications based on grid may have different constraints about resources, which can only be satisfied by certain type of resources with specific capabilities. After the users or agents have selected the resources they want, the application can run. Under the grid environment with resources coming and leaving, it is important to realize the matching between the users and the resources providers.

Commonly, resources in grid are described in different description languages and forms; however, when users want to inquire certain resources, what they get may have much difference with what the providers tend to imply. There are two serious disadvantages as a result. On the one hand, only a few of resources are utilized so that applications cannot run in a perfect condition; on the other hand, abundant resources cannot be utilized efficiently. In fact, the shortage of semantics leads to these two problems. Ontology is the explicit standard explanation about conceptual model [3]. It consists of vocabulary and a set of constraints and terms that can be combined to model a domain [4]. Since the ontology describes the semantic structure of a domain, it may be helpful for the retrieval of grid resources.

In order to solve these problems, a search engine based on ontology is proposed in this paper. Its advantages are shown as follows. Firstly, it can understand the real meaning of the query terms and improve the precision; secondly, the database of the engine not only includes the resources information but also the semantic relationships. Finally, the feedback results are relevant to what the users want even when it doesn't contain the query keywords.

The rest of this paper is organized as follows: Section 2 introduces related work; Section 3 describes the design of search engine based on ontology; Section 4 discusses the implementation of this search engine and Section 5 makes the conclusion.

2. Related Works

Several researchers have carried out some work to solve the problem. Pan proposed a retrieval system based on ontology [5]; however, the system is so simple that it does not design the storage module. Thus it can not deal with queries if there is a huge amount of classes or instances in ontology. Dai proposed a search engine based on the ontology of technological resources [6]. It only ran on web-pages and also had no storage module. Kang proposed a search engine for cloud computing system [7], which was too much focusing on the similarity between two ontologies: CO1 and CO2. Only a few of efforts can explain the structure or architecture.

Generating ontology from relational database can build ontology fast and automatically. Existing solutions build ontology from database by using certain languages, which may lack evaluation of semantics for database. To overcome the problem, we proved that E-R model equals to the description logic *ALUIN* and adopted First Order Logic to match the relationship between E-R model and the description logic *ALUIN* [8].

In order to discover a variety of network resources of structured P2P, a resource information organization method is required, which should have scalability and robustness. Structured P2P is a kind of basic network form and can be used to build the grid; however, the structured P2P features bad performance because of churn, which makes it not widely used currently. We put forward a resource information organization mode based on the node encoding and presented a resource distribution algorithm based on the node encoding [9]. The method is tolerated to churn.

To achieve multi-attribute resource discovery, we presented a framework [10]. In this framework, the ontology-based encoding method is adopted and an extended routing protocol of Pastry is used in the framework, which is flexible enough to adapt to the unstable inheritance structure of ontology; and in the frame, each query is resolved in logarithm routing hops with

high recall ratio. Furthermore, traditional structured P2P network can adapt to the framework by maintaining only a single one-dimensional code space.

3. Design of the Search Engine based on Ontology

3.1 Workflow of the Search Engine

The search engine is a kind of search tool which may help users search information in network. Generally, its workflow is described as follows: web crawling, indexing and searching. What the search engine searches doesn't cover the entire network but the established database, which stores lots of available data. There are three modules in the framework of search engine system: the collecting module, the indexing module and the retrieval module [11].

Our system includes the collecting module to collect resources in the grid environment rather in Web, the indexing module to index information of instances, and the retrieval module to search not only the database but also the ontology.

The collecting module, consisting of instance database updating, collects resource in the grid environment. Once a resource is found, some measures are taken to match resources with the concepts in the ontology. The new resource ends up with an instance of a certain concept. The indexing module, consisting of the database indexing, indexes the information of instances in the database and prepares for the coming search.

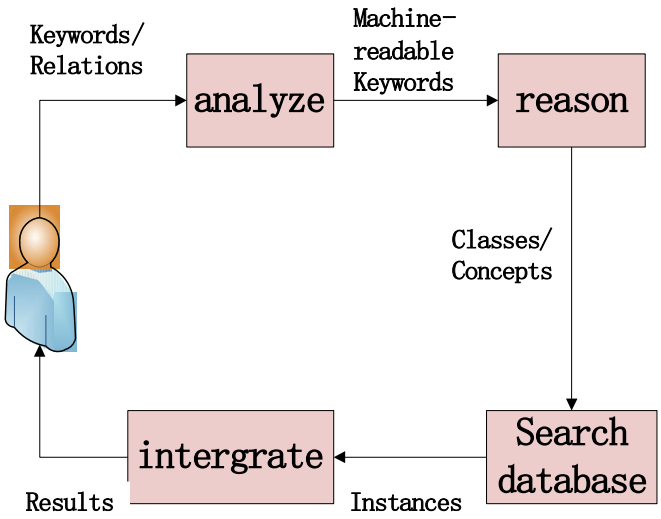

Figure 1: Workflow of the Search Engine

The retrieval module consists of query analysis, ontology reasoning and database retrieval. The module analyzes the entered term, converts nature language to machine-readable language, reason in the ontology, finally searches relevant instances in the database.

After the user has submitted the query consisting of keywords and relations, the workflow of the search engine can be described as follows:

Firstly, the query is transformed to the machine-readable keywords semantically; secondly, these machine-readable keywords are used in ontology as input to reason more concepts. Thirdly, a search of database is taken by using the result of the second step. Finally, instances will be integrated as results and returned to the users. The workflow of the search engine is shown in Fig. 1.

3.2 Structure and Function

The structure of the system is shown in Fig. 2. The structure consists of seven parts and their functions are described as follows.

1) *Ontology establishment*: it is to establish the grid resources ontology. How to build the ontology is the key point of the system. Considering the rate of recall and precision, definitions and relations in respect of the ontology should be described as detailed as possible.

2) *Instances database updating*: when a new instance comes, it should be matched with concepts in the ontology and finally stored into the database. As a result, the index of the database should be updated to make the new information to be searched.

3) *User interface*: users enter keywords and relations that they want to know in natural language, and results of the search process are also shown to the users.

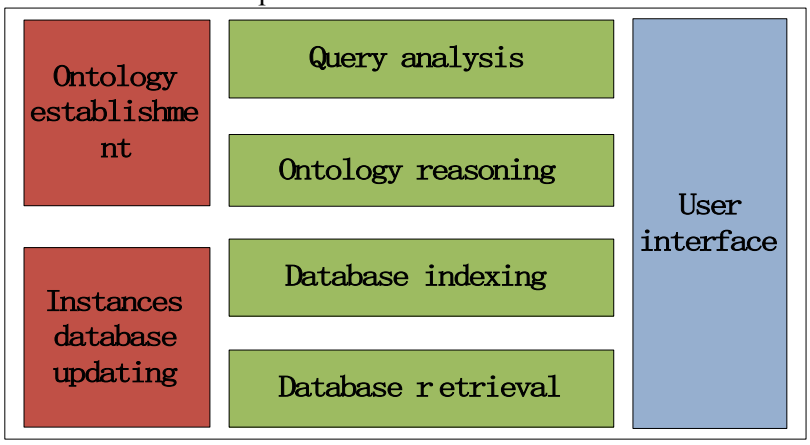

Figure 2: Structure of System

4) *Query analysis*: the keywords that users have entered are analyzed. These keywords are converted into semantic items.

5) *Ontology reasoning*: after the query analysis, the converted semantic items are used as input of certain reasoning machine to get more relevant concepts.

6) *Database indexing*: it works to index information stored in the database. The method it adopts may affect the query efficiency.

7) *Database retrieval*: its function is to find the instances in database. By using the SQL (Structured Query Language), it is easy to find all instances of a certain concept.

The structure is flexible enough to meet various demands of the grid resource retrieval. Most of all, it utilizes the semantic feature of ontology and can offer better query precision.

4. Implementation of the System

Our ontology is established in the description language RDF (Resource Description Framework). Jena, a java API (Application Programming Interface), is used to implement the query analysis and the ontology reasoning.

4.1 Ontology Establishment and Cluster Coding

Three points are considered when we establish the grid ontology. Firstly, the ontology should be unique; secondly, the resources contain the data resources, the computational resources and the services; finally, the relations among concepts should be clear and explicit. We adopt the method of building ontology from the relational database [8].

In order to make the search engine run fast and more precisely, the cluster coding is adopted, which is a method aiming at coding the concepts in ontology. This method makes each concept in ontology have its own unique ID [10]. When we enter keywords, upon analysis and reasoning, the content of query in database is IDs instead of keywords. The query costs to is

much less than the method of using keywords. Computers can analyze and reason much closely to what the users want.

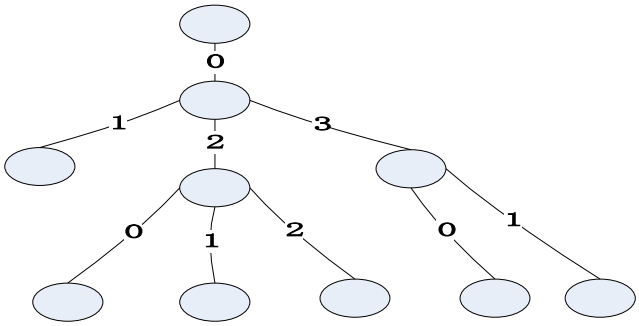

Figure 3: Cluster Coding

The ontology is treated as a tree with many branches, and the nodes in this tree are concepts of ontology. According to the tree structure, every node in this tree has a unique code as shown in Fig. 3.

Upon the cluster coding, concepts in the ontology are coded and the storage space is less; thus the searching efficiency in the database is increased.

4.2 Storage of Ontology

Considering the huge amount of resources, the ontology and instances should be stored in database. Jena provides a convenient API for database such as MySQL, Oracle and SQLServer, etc. We use Jena to store ontology schema and use cluster coding method to store instances. This separate storage in MySQL could improve the query efficiency. With Jena, the concepts in the Ontology are stored in seven tables; however, only one table stores the main information in the form of triples.

The other tables store URI (Uniform Resource Identifier) and other information, which are related to the concepts. This kind of storage schema adopts vertical schema which benefits the query efficiency.

Instances are also stored in the database. There are more instances at the bottom of ontology tree than those on the top or middle. Two kinds of table structures are purposed to make storage schema to fit this feature. One is used to store all the instances on the top and middle and it contains all the properties that have occurred. The other one is used to store instances at the bottom and it contains properties that only a kind of instances has.

```
CREATE TABLE `ontology`.`Instance_middle` (
  `Cluster code` INTEGER UNSIGNED NOT NULL,
  `Instance_name` VARCHAR(45) NOT NULL,
  `MAC` VARCHAR(45),
  `Memory` INTEGER(20) UNSIGNED,
  `Available` BOOLEAN,
  `System` VARCHAR(45),
  `Protocol_supp` VARCHAR(45),
  `Storage` INTEGER(20) UNSIGNED,
  `App_type` VARCHAR(45),
  `Version` VARCHAR(45),
```

Figure 4: Top and the Middle Table to Store All Instances

```
CREATE TABLE `ontology`.`031111` (
   `Instance_name` VARCHAR(45) NOT NULL,
  `Mem_Bus_Speed` INTEGER(20) ,
  'Clock_Speed'  INTEGER(20),
  `Exp_Bus_Speed` INTEGER(20) ,
  `Available` BOOLEAN,
  `Version` VARCHAR(45),
```

Figure 5: Bottom Table to Store Instances

Fig.4 and Fig.5 show how to create two kinds of tables. Fig. 4 treats its cluster code as its primary key. Fig.4, which is the table of CPU (Central Processing Unit), treats the cluster code as its table name. This method of storage may take much space, but its respond time is longer.

4.3 Query

According to the users' query keywords, indexing domain and the relationships the search engine can query the ontology schema, as described in RDF. Here, the new query language SPARQL(SPARQL Protocol and RDF Query Language) [12], as recommended by W3C (World Wide Web Consortium), is adopted. Compared to other languages, SPARQL has many advantages. SPARQL can be used to express queries across diverse data sources no matter the data is stored natively as RDF or viewed as RDF via middle-ware. SPARQL contains capabilities for querying as required and optional graph patterns along with their conjunctions and disjunctions. It also supports extensible value testing and constraining queries by source RDF graph. The results of SPARQL queries are the results sets or RDF graphs [12].

When the query is executed, the system must find the cluster code of the result and search database for the instances. With SPARQL, we can query the related instances according to properties. An example of querying word 'Unix' is shown in Fig.6.

The experiment environment is Inter core2 duo CPU 1.8GHz, 3GB Rom and Microsoft Windows XP. The compared test data is a set of 100 query words collected from web. Experiment results show that our ontology-based query can find all the wanted or related instances. Keyword-based query can find only 68 desired instances.

```
String prefix =
"PREFIX base:<http://www.w3.org/TR/2003/PR-owl-guide-20031209/Grid#>"+
"PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>"+
"PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>"+
"PREFIX owl: <http://www.w3.org/2002/07/owl#>";
 String k="base:Unix";
 String select = "SELECT ?s";
 String where = "WHERE { ?s rdfs:subClassOf k; ?s ObjectProperty:cc ?c
```

Figure 6: Query of Term 'Unix'

5. Conclusion

Ontology has the ability of data sharing and reusing, and it can provide a shared understanding to eliminate heterogeneity between clients and providers. A system of grid resources retrieval based on ontology is hereby presented and several key techniques of the system are discussed here. Taking advantage of RDBMS (Relational Database Management System) and the SQL's efficient match speed and SPARQL's ability to search RDF, we will make up the defects of low efficiency of the ontology data query.

The system can improve the precision and efficiency of search of resources in grid. The return results are as many as possible to make the application run to be perfect. It can also deal with ontologies with a huge amount of instances. The system also has some defect that the query optimization problem has not been taken into consideration.

References

[1] I. Foster, C. Kesselman, S. Tuecke. *The anatomy of the grid: Enabling scalable virtual organizations*. International J. Supercomputer Applications, 15(3): 200 - 222(2001).

[2] I. Foster, C. Kesselman, editors. *The Grid: Blueprint for A New Computing Infrastructure.* Morgan Kaufmann Publishers, San Francisco, USA, 1-16(1999).

[3] T. R. Gruber. *A Translation Approach to Portable Ontology Specifications*. Knowledge Acquisition, 5(2):199 - 200(1993).

[4] Z. Q. Du, J. Hu, H. X. Yi, J. Z. Hu. *The research of the semantic search engine based on the ontology*, International Conference on Wireless Communications, Networking and Mobile Computing, IEEE Computer Science, Washington D.C. USA, 5403-5406(2007).

[5] Y. Pan, T. J. Wang, X. L. Jiang. *Building intelligent information retrieval system based on ontology*, Proceedings of 8th International Conference on Electronic Measurement and Instruments, IEEE Computer Science, Washington D.C. USA, 612 - 615(2007).

[6] W. H. Dai, Y. You, W. J. Wang, Y. M. Sun. *Search engine system based on ontology of technological resources,* Journal of Software, 6(9): 1729-1736(2011).

[7] J. Kang, K. M. Sim. *Ontology and search engine for cloud computing system,* 2011 International Conference on System Science and Engineering, IEEE Computer Science, Washington D.C. USA, 276-281(2011).

[8] Y. P. Du, Z. Miao, Y. F. Zhang, W. G. Xu, Q. Q. Zhang. *Evaluation of semantics ability of E-R model.* The 9th international Symposium on Linear Drives for Industry Applications(LDIA 2013), Springer Publishing Company. New York USA, 713-730(2013).

[9] Z. Miao, Q. Q. Zhang, S. Q. Wang, Y. Li, W. G. Xu, J. Xiao. *A resource information organization method based on node encoding for resource discovering*. In: the 2013 International Conference on Computer Engineering and Network(CENet 2013), Springer Publishing Company. New York. 1263-1270 (2013).

[10] Q. Q. Zhang, Z. Miao, Y. F. Zhang, W. G. Xu, Y. P. Du. *Multi-attribute resource discovery in structured P2P networks.* In: the 9th international Symposium on Linear Drives for Industry Applications(LDIA 2013), Springer Publishing Company. New York. USA, 501-508(2013).

[11] Y. Khopkar, A. Spink, C. L. Giles, P. Shah, S. Debnath. *Search engine personalization: an exploratory study*, First Monday, 8(7):1-23(2003).

[12] E. Prud'hommeaux, A. Seaborne. *SPARQL Query Language for RDF.* http://www.w3.org/TR/rdf-sparql-query/(2008)