



Journal of
**Software
Engineering**

ISSN 1819-4311



Academic
Journals Inc.

www.academicjournals.com

Ontology-Based Scientific Service Instant Response System Design

Xiaoning Wang, Xing Fang and Ke Hou

School of Economic and Management, Xi'an Shiyu University, Xi'an, 710065, China

Corresponding Author: Xiaoning Wang, School of Economic and Management, Xi'an Shiyu University, Xi'an, 710065, China Tel: +86 18066545572

ABSTRACT

With the trend of scientific resources getting more important and the resource amount turning to massive, the precision and scalability become key factor of the scientific service system. This study builds the Scientific Service Instant Response System (SSIRS) with the technology of semantic web, cloud computing and parallel processing. For OWL data reasoning, the method which is used to transform scientific data into OWL data was proposed and the composite suggestion algorithm was achieved. MapReduce programming model was used to process data and the task abstract on multi-core with TBB was done. The experiment result showed that the SSIRS improves the searching efficiency, precision and scalability.

Key words: Scientific service, suggestion algorithm, OWL

INTRODUCTION

With the increasing and networking of scientific resources, building the scientific service system has been planned and implemented by many countries. The popularity of Open Access, document digitalization, electronic government and electronic commerce made the scientific service system face the challenge of massive data. To deal with massive data problem, one is the system architecture solutions (e.g., cloud computing) which change the way of data processing and storing. The other is data mining researches represented by the semantic web and Linked data by optimize data type to respond user requirements quickly. The cooperation and supplement of these two aspects offer users more precise and wide search results about scientific service information.

Being an important support of national infrastructure, many national scientific resource integration platform has been built, such as the U.S. Science and Technology Portal Website, the National Research Council of Canada, the Government Science and Technology Portal Website of French, the Engl and Intute and the National Science and Technology foundation platform of China. Although, these platforms contain many databases and websites already, there are still many problems need to be improved, for example, difficult to query, slow responses and mismatching results.

The definition of ontology comes from the semantic web. It is a set of data definition specifications. The research on ontology is developing from basic page tags to Linked Data which pays more attention to a precision result through open data creation. As an advanced tagging grammar in semantic web, the OWL has more attribution tags than RDF (Resource Description Framework) which can expand the semantic data to reasoning. MapReduce is the basic programming model of current popular cloud computing platforms (e.g., Hadoop, Yarn, Google Cloud). Now there are many researches on RDF processing with MapReduce but few about the OWL data ontology processing with Mapreduce. Urbani *et al.* (2012) proposed the WebPIE system which encodes the OWL reasoning with a set of map and reduce operations. Based on OWL Horst

rules, Shi and Rao (2013) implemented the msiPIE system, also achieved a MapReduce processing. Chen *et al.* (2014) presented a general OWL reasoning framework to study the implicit relationships among biological entities. Liu *et al.* (2012) investigated how MapReduce can be applied to solve the scalability issue of fuzzy reasoning in OWL.

This study designed the Scientific Service Instant Response System (SSIRS) which is going to solve the related problems of scientific service massive data processing. It combined the OWL (Web Ontology Language) ontology technology in semantic web with MapReduce which is a classical/programming model in cloud computing. To achieve instant response, the multi-core physical platform was used to optimize parallel processing.

MATERIALS AND METHODS

SSIRS provides an access interface for multiple scientific resources, its architecture includes the end-user layer, the presentation layer, the processing layer, the data layer and the infrastructure layer (Fig. 1). From bottom to above, the detail of each layer are as follow.

Taking advantage of cloud computing technology, the infrastructure layer provides a fundamental environment which is multiplexing, massive scalability, elasticity, pay as you go and self-provisioning of resources (Rana *et al.*, 2014). It virtualizes the physical resources into resource pool and fuse the hardware devices (e.g., computing, storage and network) together. As one of the development factors of cloud computing, the advent of multi-core architecture create more possibility to it (Foster *et al.*, 2008). Especially, we take the multi-core processor as important computing resources, to explore the concurrent processing ability of physical resources, making the SSIRS has the scalability oriented to multi-core architecture and a more powerful computing performance.

The data layer based on semantic web technology, using the OWL method to provide scientific resources data to upper layer which come from the corresponding equipment, document, Q and A and specialists database and other scientific resources sub-systems' data. In the former scientific resources system, these data is stored and published in the form of the raw data. The data volume increasing in scientific resources system bring big challenges to searching results efficiency and effectiveness. Semantic data can explain and interpret raw data, provide methods and rules for computer to automatic recognize knowledge definition and describe data resources. Being a part of semantic web, the OWL data based on the XML to define customized tagging and RDF to representing data, described the meaning of terminology used in Web documents (McGuinness and van Harmelen, 2004). For different using purpose, OWL provided three sublanguages, including OWL Lite, OWL DL and OWL Full. The scientific service OWL ontology building method was designed to convert the raw data into OWL form.

The processing layer uses the semantic search technology, in order to provide users with more intelligent search processing results. Its implementation is through the MapReduce programming model and TBB tools, according to the semantic search rules. We designed for the SSIRS with OWL P rules and similarity rules which will introduce in detail later. MapReduce was first developed by Google to process massive data (Dean and Ghemawat, 2008). A MapReduce program include two user define functions, map function and reduce function. When the input data is assigned, map function will scanning it and produce the intermediate key/value pair result. All the key/value pairs will distributed into corresponding partitions and processed by one reduce function. For the multi-core processor resources in infrastructure layer, MapReduce cannot take advantages of it sufficiently. To solve this problem, we use TBB on map thread to core and schedule them. TBB has many advantages, one is the task schedule program which can achieve load balance crossing multiple logic and physical core. We use TBB to supply the shortage of MapReduce invoking multi-core in order to get better performance.

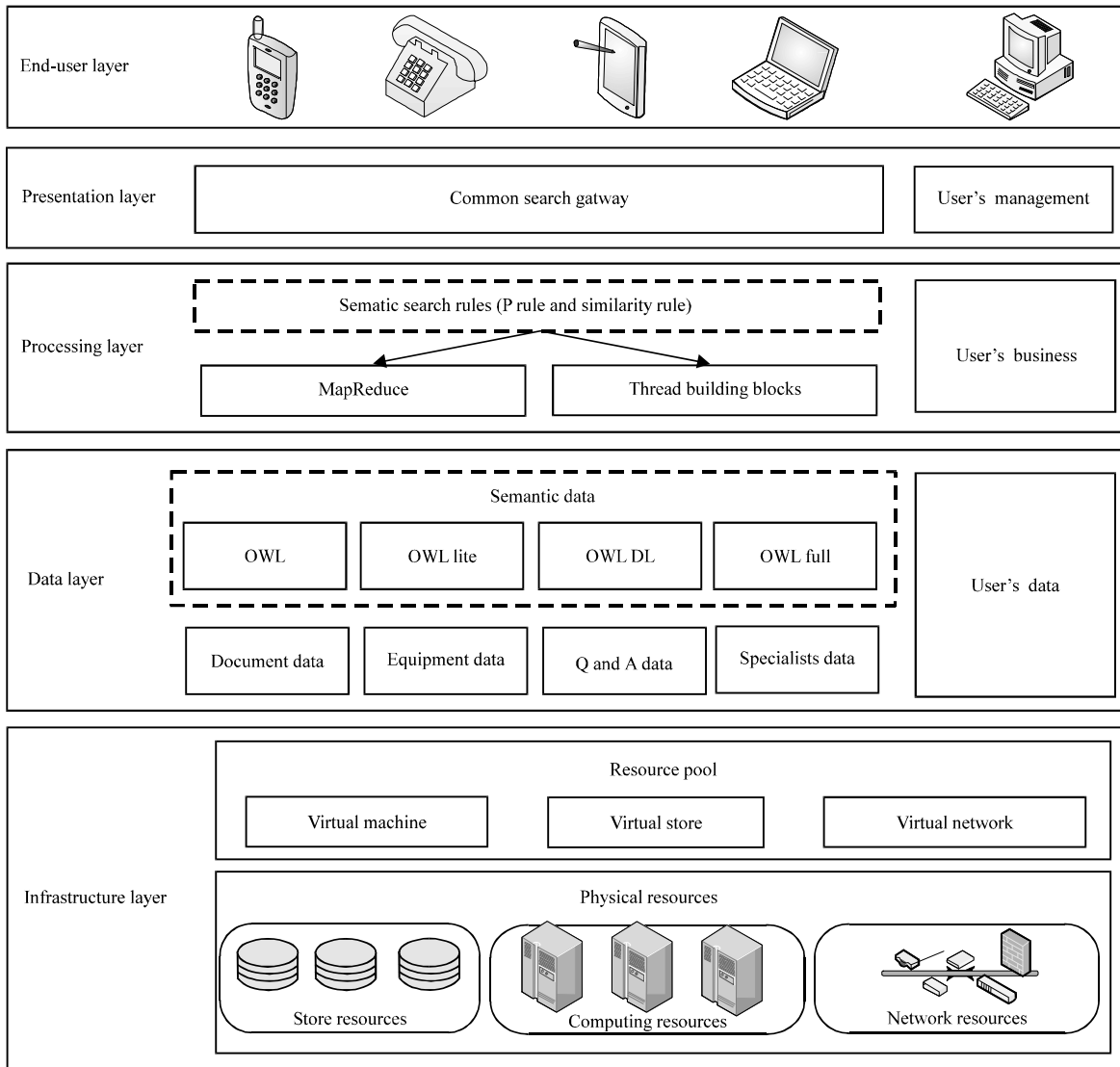


Fig. 1: Structure of SSIRS

The presentation layer provides the common search gateway to users and the registered users can have their own management interface. This layer is close to the end-user layer and deals with user interaction logic, including data model, testability and so on. The above end-user layer including all kinds of equipment that can visit SSIRS, it is the interface to display data and present outer style. To enrich the access mode of system, the call center is added to SSIRS.

Key technologies: When using the SSIRS, users can enter keywords and the system will analysis them according to response rules. The processing is achieved through the MapReduce method and using the Intel Treading Building Blocks (TBB) to call underlying multi-core resources. At the end of the procedure, the system will combine the suggestion results together and feedback to the user. To insure this procedure execute successfully, the following three key technologies should be achieved.

Scientific service OWL ontology building: The first key technology is building the scientific service OWL ontology which is used in the data layer. Because the scientific resources data is in the form of raw data, to rebuild these data by OWL form make the semantic web recognize and process them be possible. Building scientific service OWL ontology needs to make a statement about its attributes and algorithms. After stating the classes, the method will built the Data Type properties and state object properties. Looking at the scientific equipment, for example, it contains many kinds of properties which include the equipment name (i.e., Name), the instrument type (i.e., Model), the purchase date of equipment (i.e., purchase date), the equipment status (i.e., status), the affiliated unit (i.e., owner) and the equipment function (i.e., function label). The constructed ontology instance is shown in Fig. 2.

Composite suggestion algorithm: The second key technology is the composite suggestion algorithm which is used in the processing layer. This algorithm is consist of two parts, the OWL P rules and similarity suggestion rules. By using the P rules results as the input of similarity suggestion rules, the algorithm can not only take advantages of semantic reasoning but also getting more precision.

P reasoning rules of OWL: The P reasoning rules of OWL is presented by Ter Horst (2005) who combined the OWL and pD* rules, it is a common standard of OWL reasoning. Being a subset of OWL DL, the P reasoning rules of OWL had been implemented in the industrialized triple store (e.g., OWL LIM). In the semantic tagging collection, the OWL Full is hard to calculate and the RDFS only has limited expression ability, so the P rules provides a choice among them. The P rules is shown in Fig. 3.

Similarity suggestion rules: There are a lot of unstructured data in the scientific service system, this study uses the data label to process these unstructured data in accordance with the similarity suggestion rules. Taking the nano materials' characterization equipment information as an example, the similarity suggestion rules has the following procedures. Firstly, the label tree should be built to know the relationship of these equipment. As Fig. 4 shows, the depth of label "Characterization of nano materials" is "0", the "Laser Raman analysis" label's depth is "2" and so on.

```
<State rdf: ID = "name">
<rdfs: label xml: lang = "en">name</rdfs:label>
  <Abbreviation>model</abbreviation>
  <Purchase date>date</purchase date>
  <Equipstatus>status</Equipstatus>
  <Equipowner>owner</Equipowner>
  <Functions rdf:resource = "#function lable"/>...
</Function lable>
```

Fig. 2: Scientific service OWL ontology instance

Rule 1	(Prdf:type,owl:FunctionalProperty)(u, p, v)(u, p, w)	⇒	(v,owl:sameAs,w)
Rule 2	(p,rdf:type,owl:InverseFunctionalProperty)(v,p,u)(w,p,u)	⇒	(v,owl:sameAs,w)
Rule 3	(p,rdf:type,owl:SymmetricProperty)(v,p,u)	⇒	(u,p,v)
Rule 4	(p,rdf:type,owl:TransitiveProperty)(u,p,w)(w,p,v)	⇒	(u,p,v)
Rule 5ab	(u,p,v)(u,p,v)	⇒	(u,owl:sameAs,u)(v,owl:sameAs,v)
Rule 6	(v,owl:sameAs,v)	⇒	(v,owl:sameAs,v)
Rule 7	(v,owl:sameAs,w)(w,owl:sameAs,u)	⇒	(v,owl:sameAs,u)
Rule 8a	(p,owl:inverseOf,q), (v,p,w)	⇒	(w,q,v)
Rule 8b	(p,owl:inverseOf,q), (v,q,w)	⇒	(w,p,v)
Rule 9	(v,rdf:type,owl:Class)(v,owl:sameAs,w)	⇒	(v,rdfs:subClassOf,w)
Rule 10	(p,rdf:type,owl:Property)(p,owl:sameAs,q)	⇒	(p,rdfs:subPropertyOf,q)
Rule 11	(u,p,v)(u,owl:sameAs,x)(v,owl:sameAs,y)	⇒	(x,p,y)
Rule 12a	(v,owl:equivalentClass,w)	⇒	(v,rdfs:subClassOf,w)
Rule 12b	(v,owl:equivalentClass,w)	⇒	(w,rdfs:subClassOf,v)
Rule 12c	(v,rdfs:subClassOf,w)(w,rdfs:subClassOf,v)	⇒	(v,rdf:equivalentClass,w)
Rule 13a	(v,rdfs:equivalentProperty,w)	⇒	(v,rdf:subPropertyOf,w)
Rule 13b	(v,rdfs:equivalentProperty,w)	⇒	(w,rdf:subPropertyOf,v)
Rule 13c	(v,rdfs:subPropertyOf,w)(w,rdfs:subPropertyOf,v)	⇒	(v,rdf:equivalentProperty,w)
Rule 14a	(v,owl:hasValue,w)(v,owl:onProperty,p)(u,p,v)	⇒	(u,rdf:type,v)
Rule 14b	(v,owl:hasValue,w)(v,owl:onProperty,p)(u,rdf:type,v)	⇒	(u,p,v)
Rule 15	(v,owl:someValueFrom,w)(v,owl:onProperty,p)(u,p,x)(x,rdf:type,w)	⇒	(u,rdf:type,v)
Rule 16	(v,owl:allValueFrom,w)(v,owl:onProperty,p)(w,rdf:type,v)(w,p,x)	⇒	(x,rdf:type,u)

Fig. 3: P rules of OWL

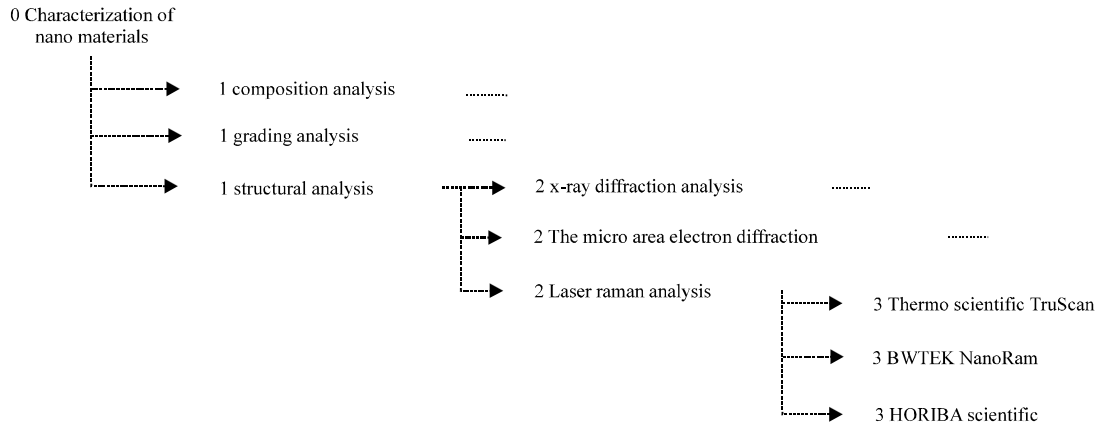


Fig. 4: Depth of label

Secondly, the distance of searching keyword and each equipment label should be calculated, as shown in the Eq. 1:

$$\text{Sim}_{\text{sub}}(\text{label}_1, \text{label}_2) = \frac{2 \times |f_{\text{common}}(\text{label}_1, \text{label}_2)|}{|\text{label}_1| + |\text{label}_2|} \quad (1)$$

where, label_1 is the depth of user's searching keyword label, label_2 is other equipment labels' depth and $f_{\text{common}}(\text{label}_1, \text{label}_2)$ is the common information function of these two labels.

Thirdly, calculate the average of distance, assuming there is a set of labels C_s and C_T , the similarity common average of the k-th label, as shown in Eq. 2:

$$\text{Sim}_{\text{avg}}^k(C_s(k), C_T(k)) = \frac{1}{n \times m} \sum_{i=1}^n \sum_{j=1}^m \text{sim}_{\text{sub}}(l_{C_s}^i, l_{C_T}^j) \quad (2)$$

Finally, compare the value of distance and average and suggest the result within the average value.

Composite suggestion algorithm of scientific service system: In order to achieve more precise search results for SSIRS, this study designed an algorithm by combining the P reasoning rules with similarity suggestion rules together. The execution procedure is shown in Fig. 5.

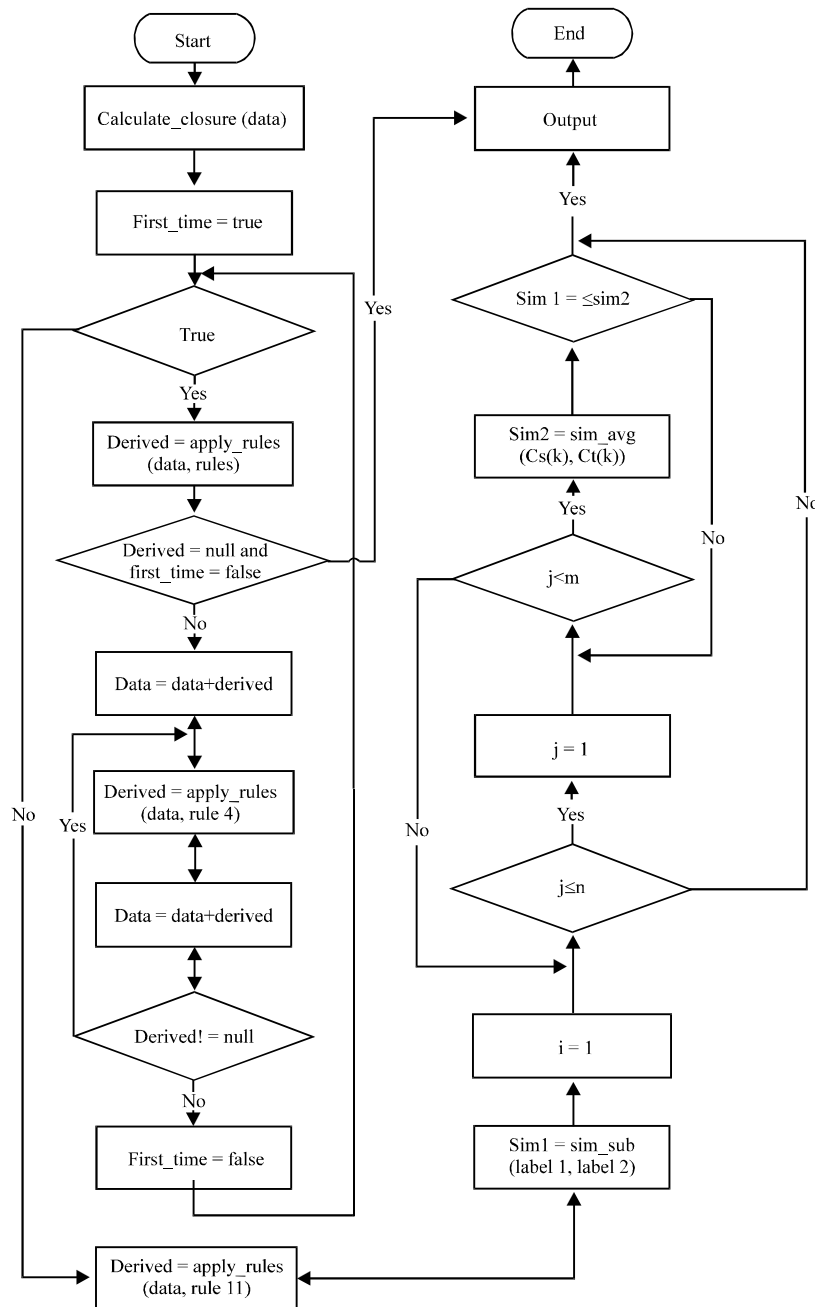


Fig. 5: Composite algorithm execution procedure

The execution steps of OWL P rules can be divided into three parts. First, analyzing the OWL P rules can know that the rule 4 and rule 11 had much iterations and other rules can be realized simply. To solve massive iteration calculation, in this study, the rule 4 and rule 11 were independent from rules.

Second, when applying other rules, many intermediate triples will be created. To decrease overhead of processing repeat triples, outputs of some rules can be used as inputs of other rules with MapReduce model.

In the end, to do a further optimization according to the new triples gotten by OWL P rule's reasoning, the label of results is used to execute the similarity suggestion algorithm and the final related results will be suggested to the user.

TBB multi-core optimization: The third key technology is the TBB multi-core optimization which is used in the processing layer. The TBB developed by Intel Corporation is a task-level parallel programming model. It integrates the advantages of many other types of parallel basic libraries and supports multiple parallel programming modes such as task parallel and data parallel. TBB makes multi-core programming easier. Figure 6 shows that TBB was called to executing Map task and Reduce task of Map Reduce job so as to reach multi-core's full potential.

- Step 1:** Map phrase of composite suggestion algorithm is copied $N \times 1$ times, in order to assign every Map task to corresponding CPU thread
- Step 2:** In the pre-process stage, building the assign index which is used to sort tasks, the process relations with each CPU task is recorded
- Step 3:** Implementing the partitioned Map task in thread, TBB is called to achieve a task abstract at the CPU thread level
- Step 4:** Executing task, results are merged and passed to Reduce task. The following procedure is similar to Map phase

Experiment

Data set and environment: In this study's experiment, dataset comes from Oxford open semantic database, the content of which is the metadata about research equipment and facilities.

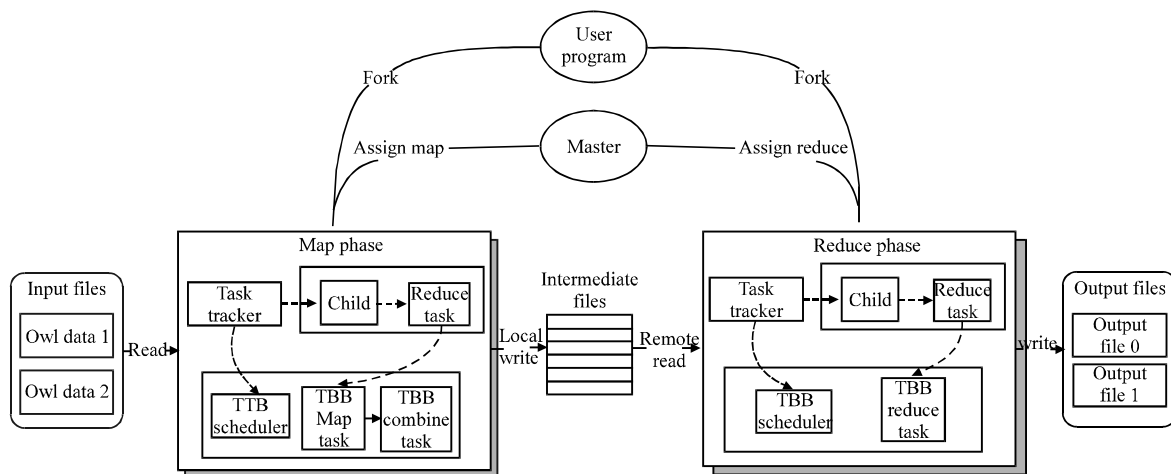


Fig. 6: Procedure of calling TBB in map reduce

The original data is in RDF type and has 19 classes (University of Oxford, 2014), before doing experiment, this data was converted into the OWL type by Protegetools (Noy *et al.*, 2001). The experimental environment is a cluster of 6 servers: One of the machines is master, the other 5 machines are slave. Each machines have one quad-core and install Ubuntu 14.04, Hadoop 0.20.203 and Intel TBB 4.3.

RESULTS

Precision: To test the precision of the composite suggestion algorithm which was evaluated by Precision-Recall method (Davis and Goadrich, 2006), five experiments were run with searching questions, for example, What equipment can process nano structural analysis? Where can I find Laser Raman equipment? The results are listed in Table 1. Since the P reasoning algorithm was employed as a subroutine of the composite suggestion algorithm, it increases the complexity of our composite suggestion algorithm, so its runtime is inevitably smaller than that of our algorithm, as shown in Fig. 7 but Table 1 shows that the precision of our algorithm has improved greatly.

Scalability: Figure 8 shows the speedup of the system as the number of nodes increases from 1-5 and the number of cores increases from 1-4. Although, our algorithm calls multi-core, the ideal linear speedup is not achieved. The reason is that there is extraoverhead such as the communication cost between nodes.

Table 1: Precision comparison

Searching question No.	P rules reasoning	Composite suggestion
1	60.16	70.64
2	70.29	85.41
3	75.60	88.32
4	77.81	90.41
5	79.47	91.36

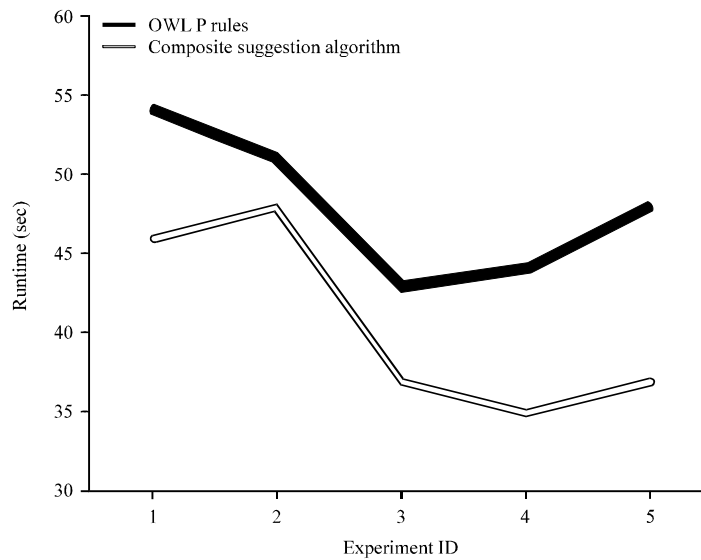


Fig. 7: Runtime comparison

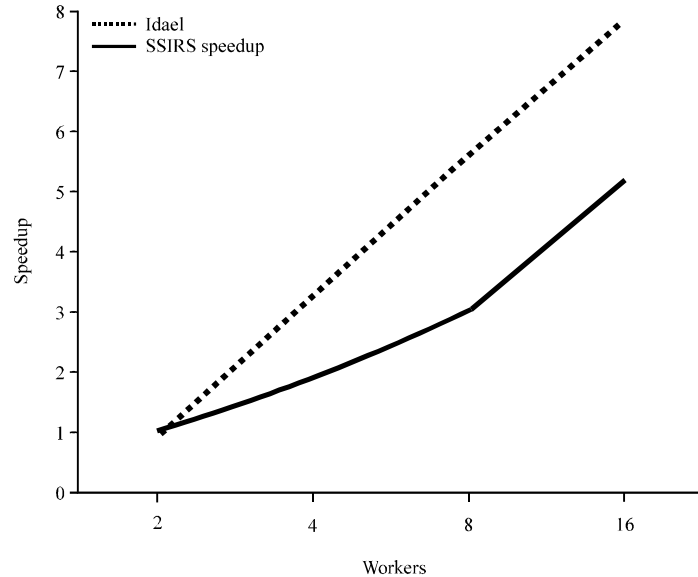


Fig. 8: Scaling performance of speedup

DISCUSSION

For the searching performance, this study presented the scientific resource semantic reasoning rules based on OWL ontology which can benefit to solve two problems. One is that the instant response system is lack of semantic reasoning, the other is searching result cannot achieve fuzzy matching. The OWL ontology rules language makes the scientific resource ontology data reasoning more intelligent which enriched the searching results. Meanwhile, to further optimize the semantic system’s searching performance, the tag similarity is calculated and the searching results are compared with the average value of similarity. Thus the selected retrieval outcomes which meet user’s requirement is filtered and the priority of the suitable results is increased.

Refer to the system performance, first, some inference engines (e.g., the ELK Reasoner by Kazakov *et al.* (2012) and Hermit by Shearer *et al.* (2008)) which can only run in stand-alone environment are hard to complete the massive ontology data reasoning task, they often encounter memory overflow and poor scalability. In our system, those problems will not appear. Dealing with massive scientific service data ontology reasoning, our system has obvious advantage in efficiency, validity and scalability.

Secondly, three kinds of parallel reasoning technology are proposed. One is the reasoning technology based on distributed hash such as the DHT based RDF store (Battre *et al.*, 2007) and RDFPeer (Cai and Frank, 2004). The other is P2P-based method (McGuinness and van Harmelen, 2004; Soma and Prasanna, 2008). The third parallel reasoning technology is based on Hadoop, the representing of which is WebPIE by Urbani *et al.* (2010), Yars2 by Harth *et al.* (2007) and Marvin by Oren *et al.* (2009). Among them, the third reasoning method can put up with better comprehensive performance and our system is belonging to this kind. In view of OWL ontology reasoning, the fuzzy pD* reasoning algorithm (fuzzy pD* for short) (Liu *et al.*, 2012) and the Distributed Reasoning Framework for Big Semantic Data based on Hadoop (DRF for short) (Chen *et al.*, 2014) are chosen in order to make the directly comparison and our system was compared indirectly with WebPIE. In the comparison, Fuzzy pD* is a typical P rules reasoning

Table 2: Speedup scalability

Workers	2	4	8	16
Speedup				
Fuzzy pD*	1	1.84	2.91	4.81
DRF	1	2.08	3.93	
SSIRS	1	1.91	3.05	5.22

Table 3: No. of Nodes versus workers

Nodes	1	2	4	6	8
Workers					
DRF	1	2	4	6	8
SSIRS	4	8	16	24	32

algorithm and DRF studies the data relation between traditional Chinese medicine and western medicine, its' iteration process is analogous to our similarity reasoning sub-routine. Table 2 shows the speedup scalability of our system and the former algorithms.

Our algorithm is more complicated than Fuzzy pD*, the executing time increases and the speedup reduces. But, Fuzzy pD* assigned three processes to run map tasks in each node and three processes to run reduce tasks, our algorithm only allocated two processes to map task and reduce task separately. Our method make every process occupy one kernel by itself, so the execution efficiency is higher and the speedup is increased. Moreover, TBB was used to take charge of scheduling and managing the multi-core resources instead of Hadoop which further optimized the system performance but Fuzzy pD* had not done so. Thus our system has a higher speedup. In addition, the Fuzzy pD* had made a quantitative comparison to WebPIE, so it can be indirectly proofed that our system's performance is equivalent with WebPIE.

Similar to Fuzzy pD*, DRF did not do any optimization in multi-core resources scheduling. Table 3 shows that our system's speedup is a bit lower than DRF but our system has a shorter runtime. For example while the number of workers is 16, our speedup is 27% off DRF. The reason is that the worker number of DRF has the same growth rate with the node number but our worker number has four times growth rate. For example, when the node number is 8 and worker number of DRF is the same but our system's worker number reaches 32.

CONCLUSION

This study designed the Scientific Service Instant Response System (SSIRS), using semanticweb, cloud computing and parallel computing technology. One of the innovative point is that the composite suggestion algorithm was proposed, by combining the OWL P rules with the similarity suggestion algorithm which improved the precision of algorithm.

The other innovative point is, to support the system searching request of massive instant response scientific resource data, MapReduce was used as the main programming model and computing framework to achieve high retrieval efficiency. Meanwhile, using TBB to get the task abstract on multi-core can let MapReduce took full advantage of multi-core resources which also get the system stability, reliability and increased the execution speed. The experiment result shows with MapReduce and TBB, the SSIRS has better efficiency and scalability.

In this study, our system only processed one Data set, we did not do further discuss about scientific resources database itself. In the following works, we will focus on transform more scientific

resources dataset to OWL data. Building different types of scientific resources' domain database and constructing more scientific resource's basic data ontology and improving the data transfer efficiency will provide better data foundation for SSIRS.

ACKNOWLEDGMENT

This study is supported by the Xi'an Technology Resource Revolution Strengthening and Plan Project "Xi'an technology resources connection form research" (No. SF1307-3), the Excellent Master Dissertation Research Paper Cultivate Project of Xi'an Shiyou University (No. 20130901).

REFERENCES

- Battre, D., F. Heine, A. Hoing and O. Kao, 2007. On Triple Dissemination, Forward-Chaining and Load Balancing in DHT Based RDF Stores. In: Databases, Information Systems and Peer-to-Peer Computing, Moro, G., S. Bergamaschi, S. Joseph, J.H. Morin and A.M. Ouksel (Eds.). Springer, Berlin, Heidelberg, pp: 343-354.
- Cai, M. and M. Frank, 2004. RDF peers: A scalable distributed RDF repository based on a structured peer-to-peer network. Proceedings of the 13th International Conference on World Wide Web, May 17-22, 2004, New York, USA., pp: 650-657.
- Chen, H., X. Chen, P. Gu, Z. Wu and T. Yu, 2014. Owl reasoning framework over big biological knowledge network. BioMed Res. Int. 10.1155/2014/272915
- Davis, J. and M. Goadrich, 2006. The relationship between precision-recall and ROC curves. Proceedings of the 23rd International Conference on Machine Learning, June 25-29, 2006, Pittsburgh, PA., USA., pp: 233-240.
- Dean, J. and S. Ghemawat, 2008. MapReduce: Simplified data processing on large clusters. Commun. ACM, 51: 107-113.
- Foster, I., Y. Zhao, I. Raicu and S. Lu, 2008. Cloud computing and grid computing 360-degree compared. Proceedings of the Grid Computing Environments Workshop, November 12-16, 2008, Austin, TX., USA., pp: 1-10.
- Harth, A., J. Umbrich, A. Hogan and S. Decker, 2007. YARS 2: A federated repository for querying graph structured data from the web. Proceedings of the 6th International Semantic Web, 2nd Asian Semantic Web Conference on The Semantic Web, Volume 4825, November 11-15, 2007, Busan, Korea, pp: 211-224.
- Kazakov, Y., M. Krotzsch and F. Simancik, 2012. Elk reasoner: Architecture and evaluation. Proceedings of the 1st International Workshop on OWL Reasoner Evaluation, Volume 858, July 1, 2012, Manchester, UK.
- Liu, C., G. Qi, H. Wang and Y. Yu, 2012. Reasoning with large scale ontologies in fuzzy pD* using MapReduce. IEEE Comput. Intell. Mag., 7: 54-66.
- McGuinness, D.L. and F. van Harmelen, 2004. OWL web ontology language overview. W3C Recommendation. <http://www.w3.org/TR/2004/REC-owl-features-20040210>.
- Noy, N.F., M. Sintek, S. Decker, M. Crubezy, R.W. Fergerson and M.A. Musen, 2001. Creating semantic web contents with protégé-2000. IEEE Intell. Syst., 16: 60-71.
- Oren, E., S. Kotoulas, G. Anadiotis, R. Siebes, A. ten Teije and F. van Harmelen, 2009. Marvin: Distributed reasoning over large-scale semantic web data. Web Semantics: Science, Serv. Agents World Wide Web, 7: 305-316.
- Rana, P., P.K. Gupta and R. Siddavatam, 2014. Combined and improved framework of infrastructure as a service and platform as a service in cloud computing. Proceedings of the Second International Conference on Soft Computing for Problem Solving, Volume 236, December 28-30, 2012, JK Lakshmiipat University, Jaipur, pp: 831-839.

- Shearer, R., B. Motik and I. Horrocks, 2008. Hermit: A highly-efficient owl reasoner. Proceedings of the 5th International Workshop on OWL: Experiences and Directions, October 26-27, 2008, Karlsruhe, Germany.
- Shi, H. and R. Rao, 2013. Scalable distributed RDFS reasoning using MapReduce and bigtable. Proceedings of the 4th International Conference on Graphic and Image Processing, October 5-7, 2012, Singapore, pp: 87680Z-87686Z.
- Soma, R. and V.K. Prasanna, 2008. Parallel inferencing for Owl knowledge bases. Proceedings of the 37th International Conference on Parallel Processing, September 9-12, 2008, Portland, OR., pp: 75-82.
- Ter Horst, H.J., 2005. Completeness, decidability and complexity of entailment for RDF schema and a semantic extension involving the owl vocabulary. *Web Semantics: Sci. Serv. Agents World Wide Web*, 3: 79-115.
- University of Oxford, 2014. Research facilities and equipment at Oxford. Research Services, University of Oxford, England. <https://www.research-facilities.ox.ac.uk/about/>.
- Urbani, J., S. Kotoulas, J. Maassen, F. van Harmelen and H. Bal, 2010. OWL reasoning with webpie: Calculating the closure of 100 billion triples. Proceedings of the 7th Extended Semantic Web Conference on The Semantic Web: Research and Applications, May 30-June 3, 2010, Heraklion, Crete, Greece, pp: 213-227.
- Urbani, J., S. Kotoulas, J. Maassen, F. van Harmelen and H. Bal, 2012. WebPIE: A web-scale parallel inference engine using MapReduce. *Web Semantics: Sci. Serv. Agents World Wide Web*, 10: 59-75.