EFFICIENT RETRIEVAL AND COMPOSITION OF WEB SERVICES USING MULTILEVEL INDEXING

P.Subha Rani[1], Pg scholar
M.Kalidass[2], Assistant Professor
[1]subha18.90@gmail.com,[2]kalidassme@gmail.com
RVS Trust Group of Institution., Dindigul

Abstract- The number of web services has grown drastically. Efficient management of web services in a service repository is an important issue to address. Given a special field, there often exists an efficient data structure for a class of objects, for example the Google Big table is very suitable for web pages storage and management. Based on the theory of the equivalence relations and quotient sets, this work proposes a multilevel index model for large scale service repositories, which can be used to reduce the execution time of service discovery and composition. Its novel use of keys as inspired by the key in relational database can effectively remove the redundancy of the commonly used inverted index. Its four function based operations are for the first time proposed to manage and maintain services in a repository. The experiments validate that the proposed model is more efficient than the existing structures like sequential and inverted index structures.

I.INTRODUCTION

Web Service Technology has been developing rapidly as it provides a flexible application to application interaction mechanism [5]. Several ongoing research efforts focus on various aspects of web service technology, including the modeling, specification, discovery, composition and verification of web services. The approaches advocated are often conflicting based, as they are on differing expectations on the current status of web services as well as differing models of their future evolution. One way of deciding the relative relevance of the various research directions is to look at their applicability to the currently available web services. To this end, the proposed system took a snapshot of the currently publicly available web services to get an idea of the number, type, complexity and composability of these web services and see if this provides useful information about the research directions.

An automatic Web service composition method based on logical inference of Horn clauses and Petri nets [15]. The Web service composition problem is transformed into the logical inference problem of Horn clauses by exploring the dependency relations among services. An algorithm is presented, to determine whether the composite service exists in time that is linear in the size of the knowledge base. It also returns the rule set necessary for the composition. Petri nets are chosen to model the rule set and its structural analysis techniques are used to obtain the composite service.

Under Service Oriented Architecture (SOA), service composition is used to integrate service components together to meet new business needs [14]. In this system, a novel data driven method is proposed to provide service composition guidance to implement given requirements. Based on the relations between business domain data and service domain data, additional data mediations are developed according to three composition rules. With these data relations and composition rules, a Petri net based approach is proposed to the composition of services. In this approach, all the input output messages of the service operations are modeled as colored places, and service operations themselves are modeled as transitions with input output places. First a Service Net (SN) that contains all operations is generated in a given service portfolio, and then Petri net decomposition techniques are used to derive a subnet of SN, and this subnet meets the need of the business requirement. This work can be seen as an effort to bridge the gap between business and service domains.

With the development of enterprise wide and cross enterprise application integration and interoperation towards Web service, Web service providers tried not only to fulfill the functional requirements of Web service users but also satisfy their nonfunctional conditions in order to survive in the competitive market [10]. A hot research topic is how to configure Web services to meet their demand when the diversity of user requirements, distinction of service components performance, and limitation of resources are considered. This approach builds a Web service configuration net based on Petri nets in order to exhibit Web service configurations in a formal way. An optimal algorithm is presented to help choose the best configuration with the highest quality of service to meet users' nonfunctional requirements.

II.RELATED WORK

Several researchers have studied problems related to indexing and composition of web services.

2.1 Bigtable: A distributed storage system for structured data

Bigtable is a distributed storage system for managing structured data that is designed to scale to a very large size say peta bytes of data across thousands of commodity servers [4]. Many projects are at Google store data in Bigtable, including web indexing, Google Earth, and Google Finance. These applications place very different demands on Bigtable, both in terms of data size and latency requirements (from backend bulk processing to real time data serving). Despite these varied demands, Bigtable has successfully provided a flexible, high performance solution for all of these Google products. In the proposed system, the simple data model provided by Bigtable has been described, which gives clients dynamic control over data layout and format, the design and implementation of Bigtable are described.

2.2 An indexing network: Model and applications

Internet data are heterogeneous, redundant, disordered, and exponentially growing [2]. Finding the right information from them becomes an ever challenging issue. Existing technologies such as inverted index and keyword matching can list user webpage matching with given search keywords. They cannot recognize potential relations among webpages to meet some rising user needs, like exploratory search and personalized search. This approach proposed an indexing network model that organizes information in webpages at three levels: words, webpages, and categories, thereby leading to a semantic association graph. Words are used as the description of webpages and categories. Webpage classification is used to gather similar webpages together. Hyperlinks imply the wisdom of the webpage creator, which can help us generate semantic relations among categories. With a clear organizational structure, an indexing network can provide support for many important applications including intelligent information retrieval, recommendation and decision support. In order to provide access to interfaces for the proposed indexing network, an indexing network algebra is defined. Finally, to validate the proposed model, an indexing network is generated based on 30 million webpages and its structure is analyzed. Methods to achieve browsing navigation and personalized search are given based on the generated network. Results reveal that the use of an indexing network can greatly facilitate exploratory information retrieval and personalized search.

2.3 Interactive web service composition based on Petri net

Business Process Execution Language for Web Services (BPEL) is becoming the industrial standard for modeling web service based business processes [8]. Behavioural compatibility for web service composition is one of the most important topics. The commonly used reachability exploration method focuses on verifying deadlock freeness. When this property is violated, the states and traces in the reachability graph only give clues to redesign the composition. The process must then repeat itself until no deadlock is found. In the proposed system, multiple web services interaction is modeled with a Petri net called Composition net (C net for short). The problem of behavioral compatibility among web services is hence transformed into the deadlock structure problem of a C net. If services are incompatible, a policy based on appending additional information channels is proposed. It is proved that it can offer a good solution that can be mapped back into the BPEL models automatically. Computation can be expensive in complex structure.

III.PROBLEM STATEMENT

Web services composition has been growing and it gives us a possibility to fulfill the user request when no single web service can satisfy the functionality required by the user [3]. The proposed system provides a new system called PSR for the scalable and efficient web services composition search using a relational database. In contrast to previous work, the PSR system pre computes web services composition using joins and indices and also supports semantic matching of web services composition. Our pre computing web services composition approach in RDBMS yields lower execution time for processing user queries despite of and shows good scalability when handling a large number of web services and user queries are demonstrated.

The use of information technology and management systems for the betterment of health care is more and more important and popular [11]. Existing efforts mainly focus on informatization of hospitals or medical institutions within the organizations, and few are directly oriented to the patients, their families, and other ordinary people. The strong demand for various medical and public health care services from customers calls for the creation of powerful individual oriented personalized health care service systems. Service computing and related technologies can greatly help one in fulfilling this task. In this approach, a Public oriented Health care Information Service Platform is presented, which is based on above technologies. It can support numerous health care tasks, provide individuals with many intelligent and personalized services, and support basic remote health care and guardianship. In order to realize the personalized customization and active recommendation of intelligent services for individuals, several key techniques for service composition are integrated, which can support branch and parallel control structures in the process models of composite services and are highlighted.

IV. METHODOLOGY

Efficient management of web services in a service repository is an important issue to address. Based on the theory of the equivalence relations and quotient sets, a multilevel index model for large scale service repositories was proposed, which can be used to reduce the execution time of service discovery and composition. Its novel use of keys as inspired by the key in relational database can effectively remove the redundancy of the commonly used inverted index. Its four function based operations are for the first time proposed to manage and maintain services in a repository. The experiments validate that the proposed model is more efficient than the existing structures like sequential and inverted index structures [1].

All similar services with the same input and output parameters are clustered which constructs the first level index. Equivalence relation is used to describe the whole index. Service retrieval is defined as an operation that accepts a set of parameters and returns a set of services . In order to test the efficiency of the retrieval, a retrieval request set is generated. An index between input class and similar input class is implemented.

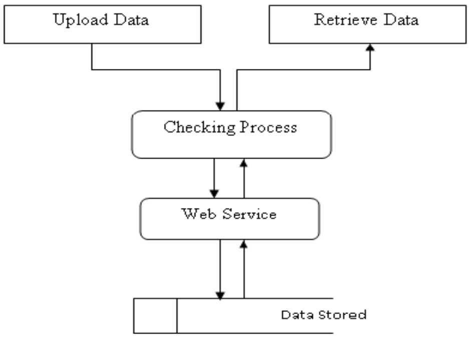

Fig. 1. Service based indexing

When computing service retrieval parameter set, all input-similar classes need be retrieved and it is then checked whether parameter set provided by the user is in the input-similar classes. Most input-similar classes are irrelevant to a user’s request. So, an index is desired to narrow the search space of input similar classes.

There is an index between a similar class and an input-similar class. Since the input similar class count is smaller than the similar class count generally, the retrieval time is further reduced. That Second level index has the integrity and contains no redundancy.

V PROPOSED MODEL

This work implements the efficient retrieval and composition of web services using multilevel indexing. Multilevel indexing model for large scale service repositories, which can be used to reduce the execution time of service discovery and composition. The multilevel index is proposed to avoid unnecessary retrievals of irrelevant ones.

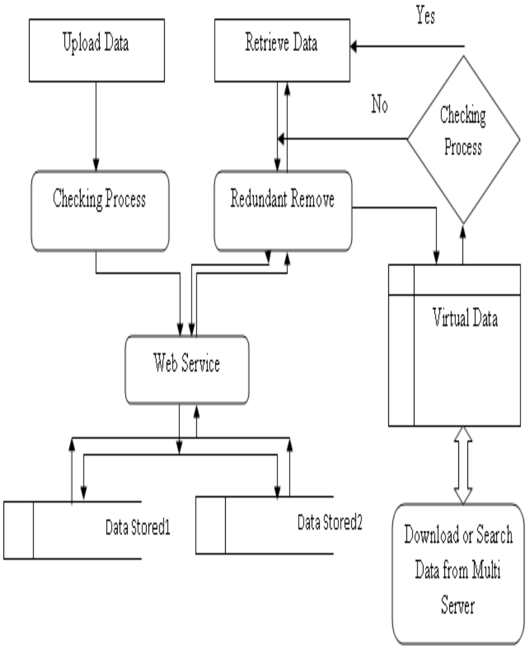

Fig. 2. Architecture of the proposed model

5.1 Equivalent based indexing

There are many similar services are all stored in a repository, If services with the same input and output parameters are clustered into a class, the search space can be reduced. After that a new service can be composed in first level and real service need to be bound. All similar services with the same input and output parameters are clustered which constructs the first level index. Equivalence relation is used to describe the whole index

5.2 Service discovery indexing

Service retrieval is defined as an operation that accepts a set of parameters and returns a set of services that can be invoked by this parameter set. Then the retrieval is needed again to find the next service till all required. parameters are found. This test set is generally used to evaluate the efficiency of service composition and cannot be directly used to evaluate the retrieval. in order to test the efficiency of the retrieval under different index structures, a retrieval request set including some retrieval requests data is generated.

5.3 Redundancy removal indexing

When computing service retrieval parameter set provided by the user in the service set, all input similar classes need be retrieved and it is then checked whether parameter set provided by the user is in the input similar classes. In a large scale service repository, it is clear that most input similar classes are irrelevant to a user’s request Therefore, another index is desired to narrow the search space of input similar classes. the inverted index cannot get rid of all the redundancy In input similar class is linked by a and b, which means that input would be retrieved twice when retrieval of parameter set and service set is computed. Clearly, it is unnecessary and such needs be eliminated.

5.4 Selection of keywords

Service retrieval is defined as an operation that accepts a set of parameters and returns a set of services that can be invoked by this parameter set. The service retrieve is related to only service input parameters. There is an index between a similar class and an input similar class all input similar classes need to be retrieved rather than all similar classes. Since the input similar class count is smaller than the similar class count generally, the retrieval time is further reduced. That Second level index has the integrity and contains no redundancy.

5.5 Service manage in virtual repository

In a large scale repository, most of the services are irrelevent and need not to be retrieved. The multilevel index is proposed to avoid unnecessary retrievals of irrelevant ones. In this the parameter set provided by the user is retrieved and stored into the virtual repository for further service. In order to maintain the new repository the searching time is consumed. Here, virtual request is consumed for every user request.

VI. CONCLUSION

The system proposed a multilevel indexing for managing webservice in large scale repository. In a large scale service repository, most of services are irrelevant and need not be retrieved. The multilevel index is proposed to avoid unnecessary retrievals of irrelevant ones. In this the parameter set provided by the user is retrieved and stored into the virtual repository for further service. In order to maintain the new repository the searching time is consumed. Here, virtual index is managed for every user request. The experimental results demonstrate the efficiency of proposed methods. The multilevel indexing model to manage web services for large scale repositories can be implemented with authentication between the repositories to prevent unauthorized access among them is the future work.

REFERENCES

[1] Yan Wu, Chungang Yan, Zhijun Ding, Guanjun Liu, Pengwei Wang, Changjun Jiang, Member, IEEE, and MengChu Zhou, Fellow, IEEE, A Multilevel Index Model to Expedite Web Service Discovery and Composition in Large-Scale Service Repositories, IEEE transactions on services computing, vol. 9, no. 3, May/June 2016.

[2] C. Jiang, H. Sun, Z. Ding, P. Wang, and M. C. Zhou, An indexing network: Model and applications, IEEE Trans. Syst., Man, Cybernetics: Syst., vol. 44, no. 12, pp. 1584–1597, Dec. 2014.

[3] D. Lee, J. Kwon, S. Lee, S. Park, and B. Hong, Scalable and efficient web services composition based on a relational database, J. Syst. Softw., vol. 84, pp. 2139–2155, Dec. 2011.

[4] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, Bigtable: A distributed storage system for structured data, ACM Trans. Comput. Syst., vol. 26, pp. 1–26, 2008.

[5] J. Fan and S. Kambhampati, A snapshot of public web services, SIGMOD Rec., vol. 34, pp. 24–32, 2005.

[6] J. Puttonen, A. Lobov, and J. L. Martinez Lastra, Semantics-based composition of factory automation processes encapsulated by web services, IEEE Trans. Ind. Informat., vol. 9, no. 4, pp. 2349–2359, Nov. 2013.

[7] M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, Service-oriented computing: State of the art and research challenges, Computer, vol. 40, no. 11, pp. 38–45, Nov. 2007.

[8] P. Sun, C. J. Jiang, and M. C. Zhou, Interactive web service composition based on Petri net, Trans. Inst. Meas. Control, vol. 33, pp. 116–132, 2011.

[9] P. Xiong, Y. Fan, and M. Zhou, A Petri net approach to analysis and composition of web services, IEEE Trans. Syst., Man Cybern. A, Syst., Humans, vol. 40, no. 2, pp. 376–387, Mar. 2010.

[10] P. Xiong, Y. Fan, and M. Zhou, QoS-aware web service configuration, IEEE Trans. Syst, Man Cybern. A, Syst. Humans, vol. 38, no. 4, pp. 888–895, Jul. 2008.

[11] P. Wang, Z. Ding, C. Jiang, and M. Zhou, Design and implementation of a web-service-based public-oriented personalized health care platform, IEEE Trans. Syst., Man, Cybern., Syst., vol. 43, no. 4, pp. 941–957, Jul. 2013.

[12] S. C. Oh, D. Lee, and S. R. Kumara, Effective web service composition in diverse and large-scale service networks, IEEE Trans. Services Comput., vol. 1, no. 1, pp. 15–32, Jan.–Mar. 2008.

[13] W. Tan, Y. Fan, and M. Zhou, A Petri net-based method for compatibility analysis and composition of web services in business process execution language, IEEE Trans. Autom. Sci. Eng., vol. 6, no. 1, pp. 94–106, Jan. 2009.

[14] W. Tan, Y. Fan, M. Zhou, and Z. Tian, Data-driven service composition in enterprise SOA solutions: A Petri net approach, IEEE Trans. Autom. Sci. Eng., vol. 7, no. 3, pp. 686–694, Jul. 2010.

[15] X. Tang, C. Jiang, and M. Zhou, Automatic web service composition based on Horn clauses and Petri nets, Expert Syst. Appl., vol. 38, pp. 13024–13031, 2011.