Dynamic Clustering SOAP Messages Based on Compression and Aggregation Model to Improve Web Services Performance

Nikhil R
PG Student, software engineering, Indian Institute of Information Technology, Srirangam, Tiruchirappalli, Tamilnadu, India.

Sankarram C
Faculty, Department of computer science, Anna University (BIT Campus), Tiruchirappalli, Tamilnadu, India.

Abstract – In Software projects, Web services can be viewed as Internet-based communications that share resources, software and information which is supposed to be available on demand for a wide range of clients over the Internet. It can also be described as the communication middleware between the servers and their clients over the Internet by establishing a connection by Hyper Text Transfer Protocol or other well-known protocols. Along with the standardization of Web services composition language and the widespread acceptance of composition technologies, Web services composition is becoming an efficient and cost-effective way to develop modern business applications. As Web services are inherently unreliable, how to deliver reliable Web services composition over unreliable Web services is a significant and challenging problem. Websites cater to millions of customers. The Web Service Oriented Architecture (SOA), an ontology rooted in the Web Services Description Language (WSDL), and modeled as a hierarchical task network for the dynamic formation of business processes when clients call these services. The extension and augmentation of traditional web services describes the business processes as a dynamic clustering of these decomposable business tasks encompassing all possible decomposition permutations. They offer services to clients dynamically. Some common services are accessed by the clients from the web services using SOAP (Simple Object Access Protocol). But when the number of requests increases the web services load increases and creates bottlenecks. The SOAP messages cause imbalance and results in poor delivery. To overcome this the model proposes aggregation of similar services and compression. The requested similar web services are first aggregated, then compressed and processed. This model reduces bottlenecks and aggregates the web services.

Index Terms – Web Services; Simple Object Access Protocol; Web service Description Language; Model driven engineering.

1. INTRODUCTION

Web Services (WSs) are the most famous implementation of Service-Oriented Architectures (SOA) allowing the construction and the sharing of independent and autonomous software. To ensure efficient implementations of compositions, we have to adequately manage heterogeneity, i.e., diverse functionality (e.g., ticket purchase, payment) and diverse transactional behavior (e.g., compensable or not) granted by multiple distributed services delivering the same functionality. The interoperation of distributed software-systems is always affected by failures, dynamic changes, availability of resources, and others. These effects are non-functional aspects and are caused by the nature of distributed software-systems. In this context, a service that does not provide a transactional property might be as useless as a service not providing the desired functional results. If the composition is based on simple WSs considering only functional requirements, then it is possible that during the execution, the whole system becomes inconsistent in presence of failures. Selection of transactional WSs allows the system to guaranty reliable composition execution. Indeed, the execution of transactional WSs will leave the system in a consistent state even in presence of failures. The Transactional WS composition problem has been extensively treated by using a predefined control structure such as workflows called Advanced Transactional Models. Different software might be built using different programming languages, and hence there is a need for a method of data exchange that doesn't depend upon a particular programming language. Most types of software can, however, interpret XML tags. Thus, Web services can use XML files for data exchange. Rules for communication between different systems need to be defined, such as to how one system can request data from another system, which specific parameters are needed in the data request and what would be the structure of the data produced. Normally, data is exchanged in XML files, and the structure of the XML file is validated against an .xsd file. What error messages to display when a certain rule for communication is not observed, to make troubleshooting easier All of these rules for communication are defined in a file called WSDL (Web Services Description Language), which has the extension .wsdl.

The service provider sends a WSDL file to UDDI. The service requester contacts UDDI to find out who is the provider for the data it needs, and then it contacts the service provider using the

SOAP protocol. The service provider validates the service request and sends structured data in an XML file, using the SOAP protocol. This XML file would be validated again by the service requester using an XSD file.

A directory called UDDI (Universal Description, Discovery and Integration) defines which software system should be contacted for which type of data. So when one software system needs one particular report/data, it would go to the UDDI and find out which other system it can contact for receiving that data. Once the software system finds out which other system it should contact, it would then contact that system using a special protocol called SOAP (Simple Object Access Protocol). The service provider system would first of all validate the data request by referring to the WSDL file, and then process the request and send the data under the SOAP protocol.

A service-oriented architecture (SOA) is a design pattern in which application components provide services to other components via a communications protocol, typically over a network. The principles of service-orientation are independent of any vendor, product or technology.

A service (or web service) can have more than one operation each containing application logic which is exposed via API (Application Programming Interface). An example would be retrieving an online bank statement. By that definition, a service is a discretely invocable operation. However, in the Web Services Description Language (WSDL), a "service" is an interface definition that may list several discrete services or operations. Elsewhere, the term is used for a component that is encapsulated behind an interface connected over a network to cooperate. Every computer can run any number of services, and each service is built in a way that ensures that the service can exchange information with any other service in the network without human interaction and without the need to make changes to the underlying program itself.

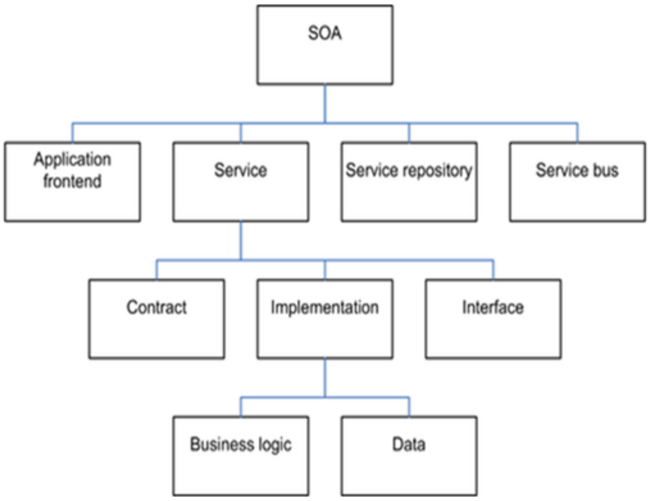


Figure 1 Service Oriented Architecture

2. RELATED WORK

2.1 CLUSTERING WEB SERVICE XML

Laura Irina Rusu et al. [6] incorporated Clustering XML Documents for improved Data mining have in their article on XML data mining, explains several concepts related to clustering XML documents and presents the task of clustering XML documents when their content or structure changes over time. In real-world applications, the number of changes from one version of an XML document to another cannot be predicted. It's always possible that an initial clustering solution becomes obsolete after the changes take place. To combat this built-in obsolescence, this article describes a non-redundant methodology for recalculating the new clusters of XML documents after the changes. You work through a step-by-step case example to help you understand the technique and apply it in practice.

Al-Shammaryet al. [2] in their work “Dynamic fractal clustering technique for soap web messages” The Simple Object Access Protocol (SOAP) is an XML based protocol that is widely used over the Internet as it supports interoperability by establishing access among Web servers and clients from the same or different platforms. However, SOAP Web services suffer the bottlenecks and congestions as a result of Web messages being bigger than the real payload in addition to the potentially increasing demand of the requested Web services. Aggregation of SOAP messages is an effective solution that has been developed to significantly reduce network traffic by aggregating SOAP messages at the server side and then multicast them to the Web clients. The major problem of the aggregation techniques is that they require efficient similarity criteria that can compute the similarity of SOAP messages as group-wise and not just pair-wise. In this paper, a new unsupervised auto class Fractal clustering technique is proposed for clustering SOAP messages into a dynamic number of clusters according to their Fractal similarities. The experimental results showed that the proposed Fractal clustering technique can improve the performance of Web services significantly better than other clustering standards such as the K-means and PCA combined with K-means by enabling the aggregation model to aggregate the most similar messages in one group resulting in better messages size reduction. Furthermore, the proposed Fractal clustering technique potentially reduces the required processing time in comparison with other standards.

2.2 COMPRESSION & AGGREGATION

Rosu et al. [11] analyzed Several compression techniques and KhoiAnhPhanet al, [9] textual aggregation models have been developed to reduce the size of the messages. For example, XMillHartmutLiefke, [8] distributes the XML tags into different containers and compresses them using semantic compressors. Differential encoding reduces the computational overhead by computing the differences between the current active message and the previous one in order to compress them only and avoid the overhead. A similarity-based aggregation technique, KhoiAnh Phan et al. [9] aims to reduce network traffic by combining similar messages and deliver the compact message using multicast protocol.

Dhial Al- Shammary et al [2] their paper Redundancy-aware SOAP messages compression and aggregation for enhanced performance say that many organizations around the world have started to adopt Web services as well as server farms and clouds hosted by large enterprise and data center for various applications. Web Services offer several advantages over other communication technologies. However, they have high latency and often suffer from congestion and bottlenecks due to the massive load generated by web service requests from large numbers of end users. SOAP (Simple Object Access Protocol) is the basic XML-based communication protocol of Web services. XML is a verbose encoding language in comparison with other technologies such CORBA and RMI. In this paper, two new redundancy-aware SOAP Web message aggregation models – Two-bit and One-bit XML status tree – are proposed to enable the Web servers to aggregate SOAP responses and send them back as one compact aggregated message in order to reduce the required bandwidth, latency, and improve the overall performance of Web services. XML message compressibility, the Jaccard based clustering technique, and the vector space model are three similarity measurements that are proposed to cluster SOAP messages as groups based on their similarity degree. The clustering based similarity measurements enable the aggregation techniques to potentially reduce the required network traffic by minimizing the overall size of the messages. The experiments show significant performance for both aggregation techniques achieving compression ratios as high as 25 for aggregated SOAP messages.

Liefkeet al.[8] analyzed in their article an efficient compressor for XML data describe a tool for compressing XML data, with applications in data exchange and archiving, which usually achieves about twice the compression ratio of gzip at roughly the same speed. The compressor, called XMill, incorporates and combines existing compressors in order to apply them to heterogeneous XML data: it uses zlib, the library function for gzip, a collection of data type specific compressors for simple data types, and, possibly, user defined compressors for application specific data types.

An XML binary tree structure based aggregation model byAl-Shammary et al. [2] was developed with the aim of providing a high compression ratio for aggregated messages. Two bit and One-bit compression techniques Al-Shammary et al. [2] are a general tree structure based models that can significantly compress individual XML messages. In this paper, new Two-bit and One-bit aggregation models are proposed that exploit the redundancies found in SOAP messages to reduce the aggregated message size

Compressibility measurement, Jaccard coefficient Wang et al. [12] and vector space technique have been developed in order to cluster SOAP messages based on their similarity. Compressibility measurement investigates the possibility of size reduction that can be achieved with SOAP message pairs. Jaccard coefficient and vector space techniques are proposed to group SOAP messages into larger predefined size clusters (not only pairs).

2.3 PERFORMANCE

Davis. D, Parashar et al. [3] implemented their work Latency Performance of SOAP Implementations present an experimental evaluation of the latency performance of several implementations of Simple Object Access Protocol (SOAP) operating over HTTP, and compares these results with the performance Of Java RMI, CORBA, HTTP, and with the TCP setup time. SOAP is an XML based protocol that supports RPC and message semantics. While SOAP has been designed as an interoperable business-to-business protocol usable over the Internet, we believe that applications will also use SOAP for interactive web applications running within an intranet. The objective of this paper is to identify the sources of inefficiency in the current implementations of SOAP and discuss changes that can improve their performance. SOAP implementations studied include Microsoft SOAP Toolkit, the SOAP: Lite Perl module, and Apache SOAP.

Flesca et al [4] in their work Fast detection of XML structural similarity have stated that because of the widespread diffusion of semi structured data in XML format, much research effort is currently devoted to support the storage and retrieval of large collections of such documents. XML documents can be compared as to their structural similarity, in order to group them into clusters so that different storage, retrieval, and processing techniques can be effectively exploited. In this scenario, an efficient and effective similarity function is the key of a successful data management process. We present an approach for detecting structural similarity between XML documents which significantly differs from standard methods based on graph-matching algorithms, and allows a significant reduction of the required computation costs. Our proposal roughly consists of linearizing the structure of each XML document, by representing it as a numerical sequence and, then, comparing such sequences through the analysis of their frequencies. First, some basic strategies for encoding a document are proposed, which can focus on diverse structural facets. Moreover, the theory of discrete Fourier transform is exploited to effectively and efficiently compare the encoded documents (i.e., signals) in the domain of frequencies. Experimental results reveal the effectiveness of the approach, also in comparison with standard methods.

Phan et al. [9] implemented their work "similarity-based SOAP multicast protocol to reduce bandwidth and latency in web services" that web services technology provided several advantages over other technologies, however it still has serious limitations, including high latency and high protocol overhead. To improve performance, SOAP network traffic needs to be substantially reduced. This paper presents a novel approach, called similarity-based SOAP multicast protocol (SMP), to address the issue of latency. SMP reduces network traffic by aggregating syntactically similar SOAP messages to form a "compact SMP message";. The addresses of clients are encoded as strings in the SMP message header. The similarity of SOAP messages is measured in pairs and is based both on the message template and on the values of each XML tag in the messages. Each XML node in a SOAP message is indexed with an identifier and its position in the SOAP message. Only the indexed form of a SOAP message is sent to clients. Intermediary routers along the paths from server to clients parse the content of each SMP message passing through them and perform necessary operations to forward it to neighboring routers. Experiments show that SMP can achieve up to 70% reduction in network traffic compared to traditional SOAP unicast.

Gray .NA et al.[5]implemented their work Comparison of Web services, Java-RMI and CORBA service Implementations have stated that an emerging trend in the Signal and Image Processing (SIP) community is the appearance of middleware and middleware standards that can be readily exploited for distributed computing applications by the SIP community. High performance computing and High Performance Embedded Computing (HPEC) applications will benefit significantly from highly efficient & portable computational middleware for signal & image processing. Open middleware standards such as VSIPL, MPI, CORBA, Java RMI, and Web Services (based on SOAP/XML), offer a unique opportunity for the rapid development of easily maintained HPEC codes that combine portability and flexibility across a number of applications This middleware infrastructure will support the rapid development and deployment of portable, efficient, SIP critical applications that will be of immediate benefit to many.

Fahringer et al. [10] analyzed a tool which helpful in managing parallel distributed system to optimize the web service performance. The different levels involved different strategies to verify the system management. The middleware components consist of different messages transferred over a period of time.

3. PROPOSED SYSTEM

The proposed work aims to develop a novel dynamic clustering model to construct clusters with a very high degree of similarity by grouping SOAP messages. The model uses Huffman Compression. All the processes are dynamic aggregates first and then compresses. The project aims to construct clusters with a very high degree of similarity by grouping SOAP messages dynamically where the propose model does not require determining number of clusters. The following are the technical flaws identified from the literature survey:

1. Finding similarity is a major problem in the previous proposed systems

2. Previous works have been Increases the network traffic and congestion in the web messages.

3. Computational overhead result is huge in the previous works

The precise prediction of the proposed model supports Huffman compression based aggregation tool to potentially reduce the required network traffic by reducing the overall size of the SOAP messages. The results show a substantial reduction for XML Web messages by the proposed model as high as 30% in comparison with Vector Space Model and dynamic fractal model.

The proposed dynamic clustering based compression and aggregation model provides a low complication clustering since it requires potentially less clustering time in comparison with other clustering models. Efficient dataset for precise clustering: the proposed XML Web messages dataset is a group of numeric vectors viewing the frequencies of XML items.

3.1 SPECIFIC OBJECTIVES OF THE PROPOSED SYSTEM

1. To reduce the overall size of the soap message by compression which will reduce the network traffic?

2. To enhance the web performance by reducing the bandwidth and reducing the required time.

3. Better result provided for medium and large dataset.

The main objective is to cluster the web services which are disparate like image processing, instant charts, mathematical function services aggregate them and cluster into the correct format finally compressing thus saving resources and computational overheads.

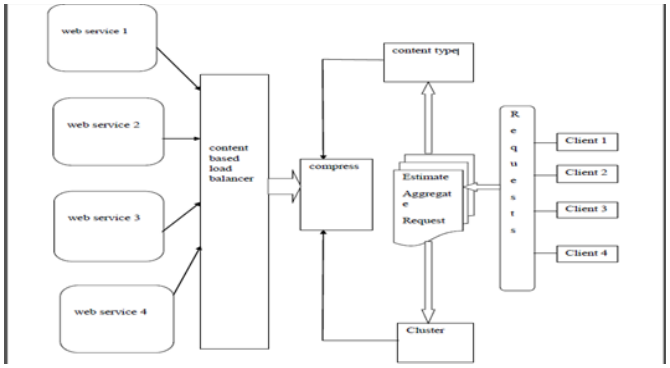
The bandwidth can be reduced by aggregation which will result in the message transfer time to improve web service performance, the required time will be reduced and achieve faster transaction. The different messages over the network will be compared and ordered according to the size.

I. THE DATASET COMPRISES OF VARIOUS SIZES WHERE DIFFERENT DATA CARRIED TO PERFORM DIFFERENT FUNCTIONS. THE LARGE DATASET AND MEDIUM DATASET WILL BE COMPRESSED TO PROVIDE BETTER RESULT FOR THE GIVEN DATA.

3.2 ARCHITECTURE DIAGRAM

The architecture comprising web services that is actually an interface consisting various operations accessible through network through the standard xml messaging. Web services function is to fulfill a specific or a set of tasks. A web services can be used alone or other web services to carryout business transaction. WSDL defines the interface to carryout interaction.

The various clients in the system request the server by various messages communicated over the network. The given request will be taken to the server by aggregating the request , The aggregated messages will be then moved to form as cluster and various clusters formed over a time and they will be compressed to reduce the unwanted data ,determining the high degree of similarity among SOAP message and also introduce a Huffman compression based aggregation tool for grouping the dynamic cluster based on the lower distance and also compressed xml message into a one compact message. To yield a better output and reducing required processing time for clustering xml message in web technology. By increasing the performance of web services. Evaluation strategy shown the better output.



The architecture diagram of the proposed work is given in Figure 3.2.

4. OVERVIEW

Table I. Analysis Patterns

Pattern	Description
Informal	The Domain is analyzed in an informal way.
Formal	A domain analysis methodology is used.
Extract from code	Mining of domain knowledge from legacy GPL code by inspection

The analysis patterns we have identified are shown in Table I.

5. CONCLUSION AND FUTURE WORK

5.1 SUMMARY

The aggregator in the web server executes bilateral negotiation using independent intelligent agents. The agent is completely ignorant of future strategies and preferences; it only tries to maximize its own gain based on its preferences and offers received from the client. The SOAP object in the module implements multiple strategy models and achieves the maximum results in less time.

The SOAP object selects the most appropriate aggregation and compression required for a client. The SOAP then initializes separate results for each client. This is used by the layer agents for decision making during clustering. The result is formalized as a string object and that result is separated and communicated back to the parties by the agent as a service in response to the negotiation request.

5.2 FUTURE WORK

The project can further be enhanced by adding the application to mobile based servers where the environment and resources are even more constrained and applications perform transactions diversely without connectivity. Further the web service based transactions can be made into an automated process for the entire period of the website without compromising on the privacy of the user. This enhances the users security and rewards the users with better transaction.

ACKNOWLEDGMENTS

We would like to express our gratitude to Indian Institute of Information Technology (IIIT – Srirangam Campus) Tiruchirappalli, India for supporting this research.The authors would like to thank all staffs for invaluable guidance throughout the development of this project.

REFERENCES

[1] Al- Shammar D,KhalilI. Dynamic fractal clustering technique for soap web messages. In:Proceedings of the IEEE international conference on Services Computing (SCC);2011.p.96–103

[2] Davis D, Parashar M. Latency performance of soap implementations. In: Proceed- ings of the 2nd IEEE/ACM international symposium on cluster computing and the grid; May 2002. p. 407.

[3] GrayNA., Comparison of Webservices, Java RMI, and CORBA service implementations. In:The fifth Australasian workshop on software and system architectures; 2004.p.52

[4] Irina L.Clustering XML documents for improved datamining.IBM;1–17.

[5] Kim J-S, Andrade H, SussmanA. Principles for designing data/compute intensive distributed applications and middleware systems for heterogeneous environments. J Parallel Distributed Comput 2007;67(7):755–71.

[6] Prodan R, Fahringer T. Zenturio: a grid middleware-based tool for experiment management of parallel and distributed applications. J Parallel Distributed Comput 2004;

[7] Bendsen, P., 2004. Model-driven business UI based on maps. In: Proceedings of the 2004 ACM SIGMOD International Conference on Management of data. ACM, New York, NY, USA, pp. 887–891. doi:10.1145/1007568.1007678.

[8] Berger, S., Grossmann, G., Stumptner, M., Schrefl, M., 2010. Metamodel-based infor-mation integration at industrial scale. In: Proceedings of the 13th International Conference on Model Driven Engineering Languages and Systems: Part II. Springer-Verlag, Berlin/Heidelberg, pp. 153–167.

[9] Bettini, L., 2013. Implementing Domain-Specific Languages with Xtext and Xtend, first ed. Packt Publishing, Birmingham, UK.

[10] Ceri, S., Brambilla, M., Fraternali, P., 2009. The history of WebML lessons learned from 10 years of model-driven development of web applications. Conceptual Modeling: Foundations and Application. SpringerVerlag, Berlin/Heidelberg, pp. 273–292.doi:10.1007/978-3-642-02463-4_15.

[11] Popp, R., Raneburger, D., 2011. A high-level agent interaction protocol based on a communication ontology. In: Huemer, C., Setzer, T. (Eds.), E-Commerce and Web Technologies, Lecture Notes in Business Information Processing, vol. 85. Springer, pp. 233–245. Doi: 10.1007/978-3-642-23014-1_20.

[12] Falb, J., Kaindl, H., Horacek, H., Bogdan, C., Popp, R., Arnautovic, E., 2006. A discourse model for interaction design based on theories of human communication. In: CHI'06 Extended Abstracts on Human Factors in Computing Systems. ACM, New York, NY, USA, pp. 754–759. doi:10.1145/1125451.1125602

[13] Shtelma, M., Cartsburg, M., Milanovic, N., 2009. Executable domain specific language for message-based system integration. In: Proceedings of the 12th International Conference on Model Driven Engineering Languages and Systems. Springer-Verlag,Berlin/Heidelberg, pp. 622–626. Doi: 10.1007/978-3-642-04425-0_48.