Impact Factor (SJIF): 3.632

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

Volume 3, Issue 2, February-2016

Survey on the Methodologies to Improve the Scalability of SWS

Mr. Ajay Kumar[1], Dr. Naresh Chauhan[2], Dr. Jyoti[3]

[1]*Computer Science & Engineering, JSSATE, Noida*
[2]*Computer Engineering, YMCAUST, Faridabad*
[3]*Computer Engineering, YMCAUST, Faridabad*

Abstract: *A system is said to scale if it is suitably efficient and practical when applied to large situations (e.g. a large input data set, a large number of outputs or users, or a large number of participating nodes in the case of a distributed system). If the design or system fails when a quantity increases, it does not scale. Similarly the nodes get overloaded by service registration queries or service discovery queries, the performance of the system is degraded. So in order to scale the service discovery system there is a need to develop some techniques which can provide the scalability to the semantic web services (SWS). This paper describes some methodologies to improve the scalability of these services in order to resolve the performance issues.*

Keywords: *Scalable Service discovery, Network scalability, Scalable semantic web services*

I. INTRODUCTION

A. SEMANTIC WEB

Developers could not process the documents on a global scale with the current web so that one possible solution is to modify the Web documents, and one such modification is to add some extra data to these documents, the purpose of this extra data is to enable the computers to understand the meaning of these documents. So Semantic Web is an extension of the current Web in which information has well defined meaning which enables the computers and humans to work in cooperation. It is a web of data that can be processed directly and indirectly by machines. It is to allow machines to "understand" the web better so that they can help people with making proper sense out of the large amounts of content available out there. It provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries.

B. SEMANTIC WEB SERVICES (SWS)

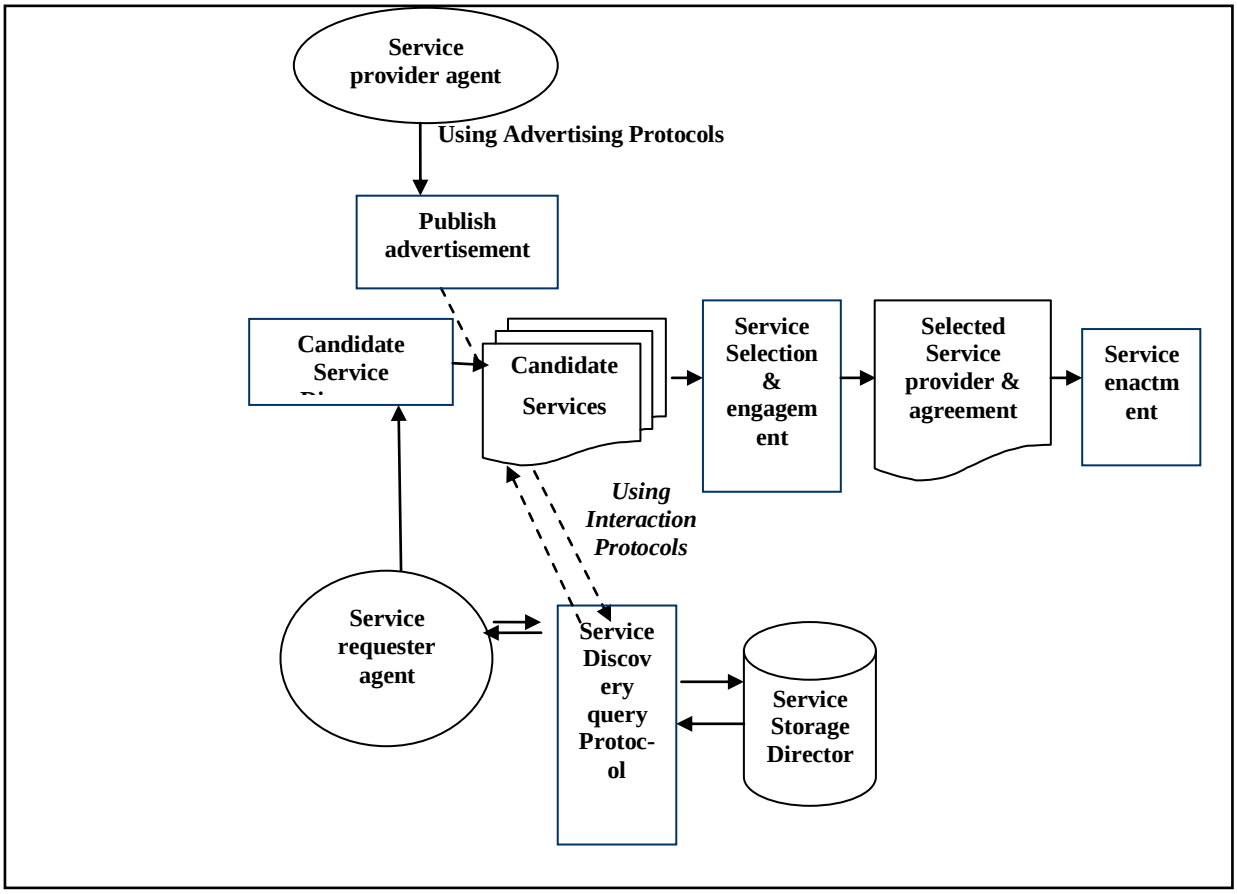
Semantic Web Services are the web services for Semantic web. Web Service is a software system designed to support interoperable machine-to-machine interaction over a network and Semantic Web Service can be viewed as a way to extend the capabilities in the direction of dynamic interoperability and addresses the need for interoperability to represent the content communicated between distributed components mentioned in the conceptual architecture in the following section.

1.1. Architecture Of Semantic Web Services

The SWS architecture [2] covers the various group of functions performed by Semantic Web agents (service providers, requesters, and middle agents called matchmakers) In this sub-section the paper describes the paper describes the functions performed by these agents

1.1.1. Service Advertisement

It is the responsibility of server to tell the environment which semantic web services are available and what they provide. A server has to advertise these services to the environment. Traditional web services use WSDL [2][25] (Web Service Description Language) for this purpose. WSDL is the XML document which provides the name and location of web services. Semantic web services on the other hand will provide semantically enhanced information means it provide meaning to the information i.e. metadata. OWL-S [2][32] (Ontology Web Language for Semantic Web) for example uses the Service-Profile for this, while WSMO (Web Service Modeling Ontology) advertises service by means of its Goals defined by client or requester and Capabilities. Also the server needs to provide a process model which describes how the client can use the services to achieve a specific effect and result. The service provider must be able to advertise the semantic web services it provides in terms of a specific ontology. The server must be able to provide a process model (model to implement these services) of the services it offers.



"Figure 1.Semantic web Service Architecture"

1.1.2. Service Discovery

It is the responsibility of client to find a set of web services and to choose one that meets the client's requirements. The client needs to find a web service that is capable of achieving the desired results. The system should provide a repository / storage directory that collects all known semantic web services. The repository/ storage directory should be able to select services from the set of known services that gives the desired results.

1.1.3. Service engagement- Negotiation & Contraction

Service engagement means agreement between both the agents i.e. service provider and service requester. The agreement is about the attributes of product like quality, price to be paid for a service and negotiation with each prospective service to reach agreement on the terms of service to be provided.

1.1.4. Service Enactment

It is the interactive process incorporated by passing messages between clients and services that accomplishes their mutual objectives. If the desired objectives are not accomplished then there would be the protocols interactions to address compensation issues.

A Critical look at the above architecture concludes that the servers are being overloaded by its functionality of advertisements & fulfilling the client's requests Due to which some performance issues occurs and results in poor scalability which is defined in the following section.

C. SCALABILITY

It is the ability of a system, network, or process to handle a growing amount of work in a capable manner or its ability to be enlarged to handle that growth. For example, it can refer to the capability of a system to increase total throughput under an increased load when resources are added. A system is said to scale if it is suitably efficient and practical when

applied to large situations (e.g. a large input data set, a large number of outputs or users, or a large number of participating nodes in the case of a distributed system). If the design or system fails when a quantity increases, it does not scale. Similarly when the nodes get overloaded by service registration queries in service discovery mechanism [8][9][10] mentioned in the above architecture, the performance of the system can be degraded. Scalability can be achieved by various methods used in service discovery mechanisms, by using extra hardware in the system, by using load balancing algorithms etc. In order to improve the performance, there is a need to design a scalable architecture for Semantic web Services which can improve the scalability of the system. The next section focuses on the detailed literature survey performed in this area of Semantic Web Services

II. RELATED STUDY

A lot of work has been done by researchers till now with respect to improve the scalability of semantic web services so that performance issues can be resolved. The work done by the researchers as of now is reviewed in this section.

Hogan et al. [3] compute the closure of an RDF graph doing two passes over the data on a single machine. They have implemented only a fragment of the OWL Horst semantics, in order to prevent ontology hijacking. Several distributed system was proposed to calculate the closure and querying.

Mika and Tummarello [4] use MapReduce to answer SPARQL queries over large RDF graphs, but details and results are not reported.

Soma and Prasanna [5] present a technique for parallel OWL inference through data partitioning. The experiments were conducted only on small datasets (1M triples) with a good speedup but the runtime is not reported.

Marvinp [6] presents a technique which partitions the data in a peer-to-peer network but results with very large datasets have not been presented.

In Weaver and Hendler [7] incomplete RDFS reasoning is implemented on a cluster replicating the schema on all the nodes. This approach is embarrassingly parallel and it cannot be extended to more complicated logic like OWL

Schlicht and Stuckenschmidt [8] presented a promising technique to parallelize DL reasoning with a good speedup but the performance was evaluated on a small input.

Stephen Gilmore and Mirco Tribastone [14] presented an model that is based on process algebra which allows service providers to investigate how models of Web service execution scale with increasing client population sizes

Christopher Olston, Amit Manjhi, Charles Garrod [15] developed a technology to enable a third party to offer scalability as a subscription service with “per-click” pricing to application providers.

Mark Nottingham [16] outlined one approach to scaling Web Services, and proposes further work which leverages Extensible Markup Language (XML) Protocol's features to help scale them and improve performance

Deshmukh, Prof. Kumarswamy Pamu [17] described the load balancing strategies, algorithms, methods by investigating the comparative behavior of load balancing with different parameters

Rhodes Hall [18] explored two load balancing algorithms with distributed software load balancers

ZhangLin, Li Xiao-ping and Su Yuan [19] presented a content-based load balancing algorithm. The mechanism of this algorithm is that a corresponding request is allocated to the server with the lowest load according to the degree of effects on the server and a combination of load state of server.

III. BRIEF DESCRIPTION OF VARIOUS METHODOLOGIES

After a detailed study of the work done by all researchers to improve the scalability of system, the advantages and disadvantages of all the methodologies used are mentioned in the Table 1 given below.

"Table 1.Various methodologies to improve scalability"

Sr. No.	Methodology	Analysis	Outcomes
1.	Use parallel OWL inference is used through data partitioning.	Abstract Materialized knowledge bases perform inference when data is loaded into them, so that answering queries is reduced to simple lookup and thus are faster.	Advantages: • Speed up the performance & scaled well Disadvantage: • Inference process is slow and memory intensive.
2.	Partitioning of data in a peer-to-peer network	Scalable Resource Document Format (SRDF) reasoning is used to deal with massive volumes of Semantic Web data	Advantages: • Improves scalability of the system Disadvantages: • Divide and conquer strategy has applied over small dataset not for large dataset till now
3.	Use Resource Description Format Schema (RDFS -extension of RDF) reasoning is implemented on a cluster which replicates the schema on all the nodes and swarn intelligence algorithms	It distributes both data and requests onto multiple computers and it describes a novel approach for reasoning within a fully distributed and self-organized storage system that does not require any schema replication.	Advantages: • Scale the system and Speed up the performance Disadvantages: • It is very difficult to run on ontology environment • This model has high computational cost
4.	Use a model that is based on process algebra which allows service providers to investigate how models of Web service execution scale with increasing client population sizes	In Performance Evaluation Process Algebra a system is viewed as a set of components which carry out activities either individually or in cooperation with other components.	Advantages: • This model has low computational cost Disadvantages: • It is very difficult to implement these kind of models in ontology environment.
5.	Use a third party to offer scalability as a subscription service with "per-click" pricing to application providers.	A fully distributed update propagation scheme which has independence relationships between query and update templates are determined offline and then used at runtime to limit the number of proxy servers receiving notification of each update.	Advantages: • Unlimited scalability to applications so that users are never denied access due to overloaded situations Disadvantages: • Lack of cache management techniques that always avoid overloading home servers by continuously monitoring and reacting to changing conditions

6.	Use the Load balancing strategies, algorithms & methods	Various load balancing algorithms like content based, Join idle queue algorithm, are used to distribute load on different computers by calculating the load.	Advantages: • Scale up the system and improve the performance. Disadvantages: • Load balancers are cost effective and lack of efficient load balancing algorithms
7.	Use more than one server to store the services	Some intermediate devices are used to handle the problem of scalability and some optimization techniques are used to allow scalability of web services.	Advantages: • Increased Scalability • Increased Performance Disadvantages • Increased Cost
8	Use the Load balancing strategies, algorithms with distributed software load balancers:	Load balancers are used as a hardware to balance the load in the system .Load balancers distributes the load of a node which is being overloaded to other nodes in the system	Advantages • Load balancers are divisible • Easy to assemble • They provide scalability Disadvantages: • They are cost effective

IV. COMPARISON

After the brief discussion on the methodologies to improve scalability, it is observed that there are some advantages and disadvantages of each method. So the comparison of these methods on the basis of some parameters is mentioned in Table 2 given below.

"Table 2.Comparison of various methodologies"

Parameters / Methods	Memory Intensive	Cost Effective	Easy To Implement in ontology environment	Applicable on large data set	Speed up the performance
OWL inference	√	√	×	×	√
Data partitioning	×	×	×	×	√
RDFS reasoning to replicate the schema	√	×	√	√	√
Model based on process algebra	×	×	√	√	√
Distributed update propagation scheme	×	×	×	√	√
load balancing algorithms	×	√	×	√	√

intermediate devices	×	√	×	√	√

V.CONCLUSION

This paper describes the research work done by the researchers to improve the scalability of the system in order to make the semantic web services scalable. It concludes the advantages and disadvantages all the methodologies to improve the performance of the system. In order to eliminate all the limitations of these methods, there is a need to design new storage directory structures, service discovery mechanisms, load balancing techniques etc

VI. FUTURE WORK

A Scalable Architecture can be implemented in the structured distributed environment in order to solve the performance and functional issues through various new techniques of storage mechanism and storage information processing models.

VII. REFERENCES

1. World Wide Web Consortium. "Web Services" [Online].Available: http://www.w3.org/2002/ws/ (visited:01/07/2009)
2. Erik Christensen, F. Curbera, G. Meredith, and S. Weerawarana (03/15/2001). "Web Services Description Language" [Online]. Available: http://www.w3.org/TR/wsdl(visited:01/07/200
3. Kagal, L., T. Finin, and A. Joshi, "A Policy Based Approach to Security for the Semantic Web", in *ISWC 2003, LNCS 2870, (2003)*, 402-418.
4. Yu, T. and M. Winslett, Supporting "Structured Credentials and Sensitive Policies through Interoperable Strategies for Automated Trust Negotiation", *in ACM Transactions on Information and System Security*, 6(1), Feb., 2003.
5. R. Soma and V. Prasanna. "Parallel inferencing for OWL knowledge bases", *In International Conference on Parallel Processing*", pages 75-82, 2008.
6. J. Weaver and J. Hendler. "Parallel materialization of the finite rdfs closure for hundreds of millions of triples". *In Proceedings of the ISWC* '09, (2009).
7. E. Oren et al. Marvin: "A platform for large-scale analysis of Semantic Web data". *In Proceedings of the International Web Science conference*, (2009).
8. A. Schlicht and H. Stuckenschmidt. "Distributed Resolution for Expressive Ontology Networks", *In Proceedings of the Web Reasoning and Rule Systems* 2009, page 87. Springer, (2009).
9. E. Guttman and C. Perkins. "Service location protocol", version, June (1999).
10. Sun Microsystems. Jini architecture specification version 2.0, June (2003)
11. Michael Nidd. Service discovery in DEAPspace ieee-pcm, 8(4):39–45, August (2001).
12. J. Wu and M. Zitterbart. Service awareness in mobile ad hoc networks.
13. Boulder, Colorado, USA,March 2001. Paper Digest of the 11th IEEE Workshop on (LANM)
14. S. Gilmore and J. Hillston. "The PEPA Workbench: A Tool to Support a Process Algebra-based Approach to Performance Modelling", *In Proceedings of the Seventh International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, number 794 in Lecture Notes in Computer Science, pages353–368, Vienna, May (1994). Springer-Verlag.

15. Olston, Christopher; Manjhi, Amit; Garrod, Charles; Ailamaki, Anastassia; Maggs, Bruce M.; and Mowry, Todd C., "A Scalability Service for Dynamic Web Applications" (2005). *CS Department.* Paper 1115. http://repository.cmu.edu/compsci/1115

16. DFSScale - M. Satyanarayanan. "The Influence of Scale on Distributed File System Design", *In IEEE Transactions on Software Engineering*, January (1992).

17. http://www.radharc.com/whitepapers/Server Load balancing.pdf

18. http://foie.ece.cornell.edu/~isn/2011-11 14%20ISN%20Seminar%20-%20Yi%20Lu.pdf.

19. C.-S. Yang, M.-Y. Luo. "content placement and management system for distributed Web-server systems". In *Proc. of IEEE 20th Int. Conf. on Distributed Computing Systems* (ICDCS'20).

20. Ankolekar, A., et. al., "DAML-S: Web Service Description for the Semantic Web", *The Semantic Web - ISWC* (2002), 348-363.

21. Gibbins, N., S. Harris, and N. Shadbolt, "Agent-Based Semantic Web Services", in *WWW2003, Budapest, Hungary*, May 20-24, (2003), 710-717

22. http://www.w3.org/TR/ws-arch. Accessed February (2012).

23. XMethods, http://www.xmethods.net , Accessed February (2012).

24. http://www.remotemethods.com , Accessed February (2012), Remote Methods: Home of Web Services.

25. WebServiceList, http://www.webservicelist.com , Accessed February (2012).

26. http://www.tutorialspoint.com/webservices/web_services_architecture.html .

27. http://en.wikipedia.org/wiki/Web_services_protocol_stack

28. http://msdn.microsoft.com/enUS/library/windows/desktop/aa966274%28v=bts.10%29.aspx

29. http://msdn.microsoft.com/enus/library/ms996486.aspx

30. http://www.w3schools.com/wsdl/wsdl_documents.asp

31. http://en.wikipedia.org/wiki/Universal_Description_Discovery_and_Integration

32. Zaidi Fayçal and Touahria Mohamed, "A Semantic Web Services for Medical Analysis using the OWL-S Language", in *International Journal of Computer Applications (0975 – 8887)*, Volume 30– No.5, September (2011).

33. Nikolaos Loutas et. al. "The Semantic Search Engine (S3E)", *published on Springer Science + Business Media*, LLC (2011).

34. Vitezslav Nezval and Francois Bartolo, "A Model for Easy Public Searching of Web Services", *JJ Yonazi et al.* (Eds.): IceND 2011, CCIS 171, pp. 209-222, 2011, and Springer-Verlag Berlin Heidelberg (2011).

35. Fangfang Liu, Yuliang Shi, Jie Yu , Tianhong Wang and Jingzhe Wu, "Measuring Similarity of Web Services Based on WSDL", in *IEEE International Conference on Web Service,* (2010).