# Accelerated Service Oriented Architecture Implementation of Deduplication Principle

**K. Venkatesh, D. Narasimhan, R. Bala Krishnan**

***Abstract— The phrase "Web Services (WSs)" are emerging as a creative scheme for furnishing the services to various immanent devices over the World Wide Web. The hasty intensification of the WSs applications and the availability of the vast count of the Service Providers create the certainty of selecting the "efficient" Service Provider by the consumers. The scenario Deduplication and Quality-of-Service (QoS) swears out as an objective to distinguish various Service Providers (SPs). The process of selecting proficient WSs / SPs, positioning and optimization of WSs Compositions are exigent dimensions of research with momentous entailments for the fruition of the "Web of Services" revelation. The term "Semantic WSs" follows appropriate semantic descriptions of WS functionality and a medium to facilitate programmed cogitating over WS Compositions (WSCs). The persisting model of the Semantic Web Services (SWSs) deals with the intriguing emerges like wretched forecast of best WSs and gemination of services with effective SPs, which heads to Quality level degradation on the Semantic Web. To deal the above identified issues, the anticipated research is planned to construct a model to manipulative the content similarities (semantic), consumption of a mixture of WSs and its corresponding SPs. After assessing these params, all the WSs are stratified on the basis of its consumption. Ultimately, the nominated scheme, selects the best and non duplicated copy of the WSs on the basis of its rating and placed it in the WSC. The process of detecting the duplicate copy would be performed by the Cryptographic Hash value of the Services. From the experimental annotations, it is recognized that our anticipated design amends the functionality of the SWSs in terms of Processor Utilization, Accessing Time, and its Space optimizations.***

***Keywords— Deduplication, Sematic web, Service Oriented Architecture, Hash Function, Web Services Composition***

## I. INTRODUCTION

The terminology "Web Services" (WSs) are software components intended to afford bear to physical interactions over a network, with the availability of huge amount of WSs, the term "Quality-of-Service" (QoS) is habitually engaged for relating non-functional features of WSs. In the midst of the various QoS features of WSs, several features are autonomous to users and have alike standards for dissimilar users (e.g., recognition, ease of use, etc.). The principles of the user autonomous QoS features are frequently accessible by various Service Providers (SPs) or by the popular third-party registries (e.g., UDDI). Web services (WSs) is a neutral architectural principle for various systems. It follows the "fill-gap" strategy for loosely coupled services and assists to create a new service from existing services. The major merit for web services is that it reuses the existing parts of code and append new building parts which enhance the services. In other terms, it redefines the applications from reusable parts. The components of WSs provide better service to users. The major challenge faced over the Service Oriented Architecture (SOA) is on improving the Quality-of Service (QoS) with minimal response time. Another mile stone to achieve in web service is to optimize the service with efficient non duplication of services. In order to achieve the above stated features, the proposed model is structured with the principle deduplication over the SOA.

Deduplication much needed technique for data centre environment. It applies hashing technique to identify the redundant files or streams. Chunking is a process of breaking data in standardized unit. File level deduplication normally less versatile, it potentially identifies components within certain files. Block-level deduplication sense that will compare blocks regardless of file type, applications and operating system origin. Block based deduplication is requires high processing power and it will create huge amount of metadata information to track individual blocks [9]. Chunks are identified based crypto graphical hashing techniques provide unique fingerprint value. It identifies the redundant content in source level or at Target Level. The principle of the deduplication scheme is stated in the Figure 1.

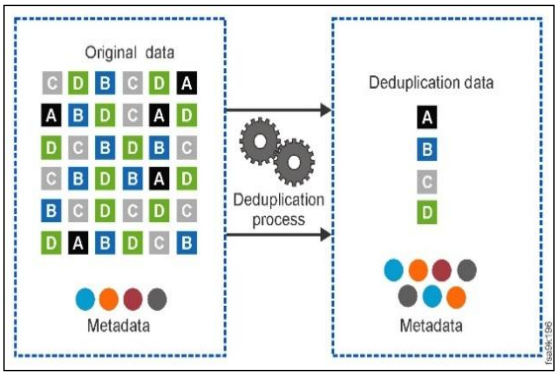


**Figure 1. Principle of deduplication.**

The other incisions of the article are depicted as succeeds: Section 2 presents about the Literature critique of the above stated terminologies. Section 3 elaborates the limitations of the underlying model and section 4 elaborates the proposed

**Revised Manuscript Received on September 2, 2019.**

**K. Venkatesh,** Assistant Professor, Department of CSE, Srinivasa Ramanujan Centre, SASTRA DEEMED UNIVERSITY, Tamil Nadu, India.

**D. Narasimhan,** Associate Professor, Department of Mathematics, Srinivasa Ramanujan Centre, SASTRA DEEMED UNIVERSITY, Tamil Nadu, India.

(Email: narasimhan@src.sastra.edu)

**R. Bala Krishnan,** Assistant Professor, Department of CSE, Srinivasa Ramanujan Centre, SASTRA DEEMED UNIVERSITY, Tamil Nadu, India.

work in which a new scheme for identifying duplicate static contents over the web services repository have been presented. Section 5 exhibits the experimental observations of the anticipated model. Finally Section 6 concludes our work.

## II. LITERATURE CRITIQUE

In this section, various existing WSC approaches and basic Deduplication principles are presented. The popular underlying WS approaches are non-functional and semantic based. Starting from the primary set of accessible forces, the Services Composition can be specified and the practice is offered below.

Web Services Composition (WSC): It aims at selecting and WSs offered by various vendors. Automation of WSC focuses on overcomes the issue that by a single SP can satisfy the goal of an user[1]. A huge count of principles has been proposed, which includes Logic, Matchmaking, Graph-Theory and AI-based.

WSC Optimization: It focuses on choosing conquer elements of service to optimize the quality of the WSC[13].

Measurement of Scalability: This feature is to maintain WSCs with a large number of services and the services are prioritized by the scalability dimension. In this dimension, the GA-based approaches [10], and grade superior than IP-based services even if suboptimal results are accomplished. Complete GA-based functions advance the scalability by analysing with different params [11] such as the evolution policy, inhabitants and so on. This crisis can also be modelled as a Knapsack problem, wherein Peng J et al.[10] presented a searching scheme based on the stochastic nature of dynamic programming to resolve the quandary.

Non-Functional measurement: The non-functional dimension assorts the features on the basis of their facility to deem non-functional contents of WSCs. By using the descriptions of semantic services, optimization could be carried out in their composition [13, 14], and not taking into account of the granted services and its quality measure. Such approaches are not fit to rank WSCs services according to business rations. Mateos et. al. [11] presents a model which deals with the quality of the WSCs and the quality is accessed by means of the services response time, reliability, availability as well as domain reliant features.

Semantic level measurement: It proceeds by its ability to optimize the SCs by the semantic quality of services. Through the increase in the quality of the compositions and their connections, the mediator counts are generated manually, which are compulsory to find the semantic heterogeneity between the exchanged/shared data in the WSC [11]. Most of the non functional quality-based schemes are prioritized in dimension as very low. They believe only the semantic matches along with output-input connections of various WSs. Others are not focuses on evaluation on Semantic WSCs[13,14]. The approaches with low priorities on the non-functional dimension are scored very well here [9].

The approach of semantic web deals with the job of optimizing the SCs by their semantic similarity. By improving the quality of the compositions and their interconnections, the services offering process could be improved and it reflects in terms of service offering time, which upgrades the overall quality of the system. This architecture of Web Service or Distributed Web Server framework is presented in Figure 2.

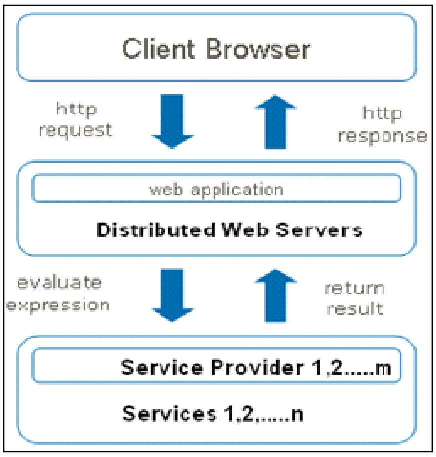


**Figure 2. Architectural flow of Distributed Web Servers**

The terminology Data deduplication is followed to find and remove the duplicate pieces of content (Chunks) from a large storage repository [4,5]. The most appropriate candidates for deduplication implementation were platform virtualization and backup server, because both will handle a huge quantity of similar contents [6]. Deduplication takes position on the file level and at block level.

In the File-Level Deduplication (FLDD), duplicate or redundant copies[7,8] of the content are eliminated and this kind of deduplication is called as single instance storage (SIS).

In Block-Level Deduplication (BLDD) [2,3], redundant or duplicated blocks of data on unique files are eliminated. It occupies more space than SIS and is otherwise called variable length deduplication.

## III. IDENTIFIED LIMITATIONS AND NEED FOR THE PROPOSED MODEL

From our literature critique, it is to be observed that various researchers have been suggested a numerous approaches on Web Services which includes Matchmaking-Based [10, 11], Logic-Based [12], and Genetic Algorithm-QoS-based mechanisms to improve the performance of Distributed Web Servers and there is need to offer efficient services to the clients without duplication issues. Common drawbacks of Web Services Composition are listed below.

1. Inevitability of selecting the “right” service provider
2. Poor ranking and optimization of WSCs
3. Poor Redundant Services Prediction Accuracy and Authentication

To address the above mentioned limitations, the proposed model is constructed with a proficient QoS based Web Services Composition Mechanism with Deduplication detection facility, which helps to improve the overall QoS over the Service Oriented Architecture (SOA).

## IV. PROPOSED METHODOLOGY

From the prior segment, it is to be identified that a variety of schemes, that are presented here to improve the process of finding duplication of services over the Semantic Web environment. To overcome these identified issues, this proposed model is constructed with a web services composition mechanism with the implementation of cryptographic hash function principle for identification of duplicate services during the services inclusion and offering phase of operation.

. The model of the proposed scheme is stated in Figure 3. Initial Client requests are received various sources by the web server application. After analyzing the request the available services are offered to the corresponding clients by the distributed web servers. Web Clients / Requestors are having exclusive identification details and having some access restriction policies at each user level. The end user who is requested the service need to satisfy the rules to obtain the service from the service providers or from the WSC. The Composition mechanism maintains the list of frequently accessed services from various service providers and the access pattern of users. From the list the frequently accessed services copy would be retained in the composition to grant services to the clients. The Deduplication Detection Mechanism (DDM) in the WSC frequently checks the services during the Services Loading Phase by using its Cryptographic Hash value of the Services.

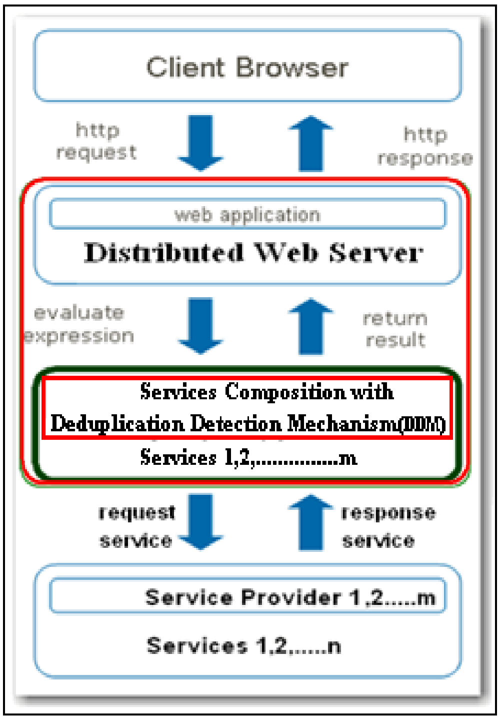


**Figure 3. Architecture of Proposed WSC Model with Deduplication Detection Mechanism (DDM)**

If same services with two different names are frequently requested in sense, it would be loaded twice in the existing model, whereas in the proposed work the hash value detects the redundant services and those services will be loaded once with two reference names and for those different requests the same file is offered to the clients. The execution procedure of Web Services Composition with Deduplication Detection Mechanism (WSC-DDM) is stated in Figure 4.

***Procedure: WebServices Composition with Deduplication Detection Mechanism (WSC-DDM)***

***Inputs: ServiceName***
***Output: HashValue, ReferenceName***

```
begin
HashValue = { }, ReferenceName={ }
Serv_hash=Hash(ServiceName)//Cryptographic Hash

//LookUp on Service availability in  DataBase table
if(AvailabilityinLookUPTable(Serv_hash))//true
//No need to load the service on WSC
ReferenceName=(ExtractfromLookUpTable(Serv_hash
));
else
//Load the Service in WSC
WSC_Services = WSC_Services + { Copy
(ServiceName)}

//Update the Hash Value in  DB
LookUpTable(insert(Serv_hash));
ReferenceName=(ExtractfromLookupTable(Serv_hash)
);
endif
return {Serv_hash,ReferenceName};
end
```

**Figure 4. Web Services Composition with Deduplication Detection Mechanism (WSC-DDM) Specification**

The WSC-DDM algorithm executes during the Services Loading Phase of Web Services Composition (WSC). The frequently requested services list is extracted by the WSC model and before the process of loading the services the proposed WSC-DDM checks the availability of the service in that composition and it could be done by comparing the cryptographic hash value of the services. The hash value of the requested service is computed and the hash value is compared with the previously available hash values in the Database LookUptable.

If the hash value exists then the corresponding service name would be extracted from the LookUptable and it is returned by the WSC-DDM procedure to offer the service to the client. Those identified services copy are not included in the WSC because it is already available in the composition with a different name.

If the hash value is not exists then it would be identified that it is a new service to the WSC environment and it is loaded on to the environment and the details of hash value and the reference name are included in the LookUptable.

## V.EXPERIMENTAL RESULTS

The projected WSC-DDM has been implemented using Java Programming Language with MySQL as backend. The performance of the system is accessed on a Pentium Core 2


Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Duo system of 2.6 GHz with 2 GB RAM on Windows 8 platform. The experiments have been conducted by considering a total of 417 Portable document files as services.

The following table shows a list of sample pdf services offered in the model with its corresponding hash values and size in bytes.

**Table 1. List of Sample Services, size and its Hash value**

| S.No | File | Size in bytes | Hash Value (SHA-1) |
|---|---|---|---|
| 1 | File1.pdf | 91785 | 1ff255f66cdd7003667999664665ffb777ccdbb7554bb088fff3ee6aae225dd744d228552773cc5bb9cccff1aaa441bb |
| 2 | File2.pdf | 469609 | b774ee1773bb0ee344fbb977599d443112ff400dddc66122233bcc055d88e885888330fff88eccc667ffb665bba555dd |
| 3 | File3.pdf | 118299 | f22c55233cff3bb0aabddc448662660dda111ccb77daaddd7cc777fee6228bb6990bbc226226ff500e00c33888244466 |
| 4 | File4.pdf | 139090 | f99866f99577e554ccf88222c99033d779ee9ee7005bb5ffbeec774bbedd711399055099a99affc991000dd455988900 |
| 5 | File5.pdf | 176012 | b88766d33200caa522add8229447117ee1aa3cc2556dddcc9ff8228ffc11722c882337bb0991557bb733266211ccc066 |
| 6 | File6.pdf | 289958 | d00233e888677866844533 6774663bbbff088edd688e77d77033cff4cc755acc6aa1ff6bb2bb6bbbdd944311d99100044 |
| 7 | File7.pdf | 223408 | f88333933499f99f222550336006cc333944b55d449ff2aa1aab00a88caa6000bba11733a22f88cff0bba776bbeffa77 |
| 8 | File8.pdf | 502251 | cee311e55aee9ddd88311f448eeeffa55c555993dd6ee2ee733f222cca66bee755011b66cffeee1115cc7443773ee277 |
| 9 | File9.pdf | 262960 | 0cc2889665440eeb22f77bdd522d88b44988a777221336335ff599799388477399d229228332003118111ee7445cc533 |
| 10 | File10.pdf | 622199 | ebb4111eea224336006ff155455e007662559bb2eeaddb770992eeaff866bccc77e11e88f88666a11caa966955766733 |
| 11 | File11.pdf | 334603 | 7aa022833f441bb5330bb5335cc399a882cc755666988cff666f33011877277eccdcc2ee955dcc1ccdcc722acc055322 |
| 12 | File12.pdf | 583492 | dcc000999644beefee622d00a77266766077daa4dd0dda223115bb366accbdd4dd4115884aad77cbb388b88377edd222 |
| 13 | File13.pdf | 146234 | 155eaad33dbbb77044577722e118bb399022d333bbd222ff600788cdd300900144777bff5ff4dd2dde66c99888955c99 |
| 14 | File14.pdf | 484473 | 933cffbcca88add1ffd99efffbb4cc377f777cc8ff8bb233400b553441ff066333344fbb366122577baab22999c99bdd |
| 15 | File15.pdf | 421580 | 4664cc088d773bbeff2ff100177a997998777dd9ff944766feebee1002aa0bbedd9ffe229dd9bb6ddb001ee8dd900844 |
| 16 | File16.pdf | 325906 | 0dd2aa933bbbf22877033500a664aad11733c77799055633dcc9dd8aa5224558aa3885bb8553cc1661aabccc00a77cdd |
| 17 | File17.pdf | 259604 | 9ddaaab55f11b555cc8ff0997ff666144c77e339330ee955c55611a99f33d77322d99855077188e444aa6cca66933abb |
| 18 | File18.pdf | 686372 | 944b00ccc7000ff6220bb477899066eaa1dd155188900166edd933d88777411e11e885ddd779dd6ee4eeabbdaa4887aa |
| 19 | File19.pdf | 469609 | b774ee1773bb0ee344fbb977599d443112ff400dddc66122233bcc055d88e885888330fff88eccc667ffb665bba555dd |
| 20 | File20.pdf | 176012 | b88766d33200caa522add8229447117ee1aa3cc2556dddcc9ff8228ffc11722c882337bb0991557bb733266211ccc066 |
| 21 | File21.pdf | 469609 | b774ee1773bb0ee344fbb977599d443112ff400dddc66122233bcc055d88e885888330fff88eccc667ffb665bba555dd |
| 22 | File22.pdf | 176012 | b88766d33200caa522add8229447117ee1aa3cc2556dddcc9ff8228ffc11722c882337bb0991557bb733266211ccc066 |
| 23 | File23.pdf | 176012 | b88766d33200caa522add8229447117ee1aa3cc2556dddcc9ff8228ffc11722c882337bb0991557bb733266211ccc066 |

Sometimes a same service is offered by multiple service providers with different names. The similarity between the services would be identified by its Hash value. Before loading the services (Frequently accessed by clients) in the WSC, the hash value of that service is compared with the previously available services in the WSC. The proposed model maintains the WSC services and its Hash value in a table named LookUp. If the hash value is already available, then the service is already loaded in the WSC with a same or different name and new service reference name would be included or updated in the WSC LookUp table.

Table 2 lists frequently accessed services having same content and different name. From the proposed WSC-DDM the lists of redundant services are identified and it would be loaded once in the WSC.

**Table 2. List of Redundant Services, size and its Hash value**

| S. No | Services Name | Size in bytes | Hash Value (SHA-1) |
|---|---|---|---|
| 1 | File19.pdf<br>File 21.pdf<br>File2.pdf | 469609 | b774ee1773bb0ee344<br>fbb977599d443112ff<br>400dddc66122233bcc<br>055d88e885888330ff<br>f88eccc667ffb665bb<br>a555dd |
| 2 | File20.pdf<br>File21.pdf<br>File22.pdf<br>File23.pdf<br>File5.pdf | 176012 | b88766d33200caa522add<br>8229447117ee1aa3cc255<br>6dddcc9ff8228ffc11722c<br>882337bb0991557bb7332<br>66211ccc066 |

Figure 5 presents the functionality of the proposed model which detects the best non redundant Services on WSC for client requests from various environments.

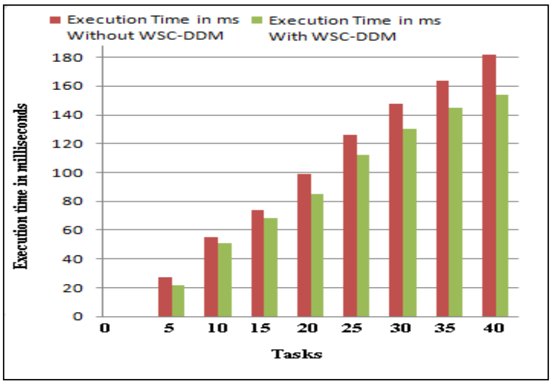


**Figure 5. Evolution of WSC in terms of Execution time with and without Deduplication Detection Mechanism (DDM)**

The Web Services Composition (WSC) with Deduplication Detection Mechanism (DDM) reduces the request handling time which improves the overall System performance. From the experimental outcomes of the proposed model, it is to be revealed that the proposed model suits well for detecting Deduplication of services on WSC model and it is capable for handling huge volume of client requests

The model with WSC-DDM feature grants an efficient performance over the SOA and hence the proposed model of the Semantic Web is enhanced significantly with essential QoS and the outcomes with comparative values are presented in the Figure 6.


**Figure 6. Evolution of WSC in terms of frequently accessed services with and without Deduplication Detection Mechanism (DDM)**

## VI. CONCLUSION

The proposed WSC model with DDM is compared with the existing models without the Deduplication detection mechanism which reduces the efficiency of Semantic Web in terms of holding duplicate copies of same services with different names. To address the above stated issue, the proposed model is implemented and from the experimental observations, it is to be identified that the proposed model offers efficient services in terms of

- Handling more user requests
- Detecting the Duplicate Services
- Execution time and Storage Space Optimization.

## REFERENCES


1. R. Bala Krishnan et al., Development of an Efficient QoS based Web Services Compositions Mechanism for Semantic Web. Research Journal of Applied Sciences, Engineering and Technology 4(8): 987-994, 2012.
2. Tian Y, Khan SM, Jiménez DA, Loh GH (2014) Last-level cache deduplication. In: Proceedings of the 28th ACM International Conference on Supercomputing, pp 53–62. https://doi.org/10.1145/2597652.2597655.
3. Hovhannisyan H, Qi W, Lu K, Yang R, Wang J (2016) Whispers in the cloud storage: a novel cross-user deduplication-based covert channel design. Peer-to-Peer Networking and Applications, pp 1–10. https://doi.org/10.1007/s12083-016-0483-y.
4. Tang Y, Yang J. Secure deduplication of general computations. In *Proc. the USENIX Annual Technical Conference* (*ATC*), July 2015, pp.319-331.
5. R. Bala Krishnan et al., Web Services Exploration over the Semantic Web Composition using Diagonal Matrix Intrigue International Journal of Applied Engineering Research, Volume 9, Number 4(2014) pp: 401-408.
6. Zhang W, Yang T, Narayanasamy G, Tang H. Low-cost data deduplication for virtual machine backup in cloud storage. In *Proc. the 5th USENIX Workshop on Hot Topics in Storage and File Systems* (*HotStorage*), June 2013, Article No. 12.
7. Xia W, Jiang H, Feng D, Hua Y. Similarity and locality based indexing for high performance data deduplication. IEEE Transactions on Computers, 2015, 64(4): 1162–1176.
8. Xia W, Jiang H, Feng D, Douglis F, Shilane P, Hua Y, Fu M, Zhang Y, Zhou Y. A comprehensive study of the past, present, and future of data deduplication. Proceedings of the IEEE, 2016, 104(9):1681–1710.
9. Meister D, Brinkmann A. Dedupv1: improving deduplication throughput using solid state drives (SSD). In: Proceedings of the 26th Symposium on Massive Storage Systems and Technologies. 2010, 1–6.
10. Peng J, Li H, Liu Y, Juan L, Jiang Q, Wang Y, et al. InteGO2: a web tool for measuring and visualizing gene semantic similarities using Gene Ontology. BMC Genomics. 2016;17 Suppl 5:530. doi: 10.1186/s12864-016-2828-6.
11. Mateos, C., Crasso, M., Zunino, A., & Ordiales Coscia, J. L. (2015a). A stitch in time saves nine: Early improving code-first web services discoverability. *International Journal of Cooperative Information Systems*, *24*(02).
12. Own, H., & Yahyaoui, H. (2015). Rough set based classification of real world web services. *Information Systems Frontiers*, *17*(6), 1301–1311.
13. Garriga, M., Mateos, C., Flores, A., Cechich, A., & Zunino, A (2016). {RESTFul} service composition at a glance: a survey. *Journal of Network and Computer Applications*, *60*, 32–53.
14. Freddy Lecue and Nikolay Mehandjiev, “Seeking Quality of Web Service Composition in a Semantic Dimension”, IEEE Transactions on Knowledge and Data Engineering, Vol. 23, No. 6, June2011.