

# Improved Shortest Path Method to Select the Best Path from Multi-Path Web Service Composition Graph

Dr. S. Justin Samuel<sup>1</sup>, M.Gnana Sagaya Sharmila<sup>2</sup>

<sup>1</sup>Professor, <sup>2</sup>PG Scholar

Faculty of Computing,  
Dept. of Information Technology,  
Sathyabama University  
Chennai, Tamil Nadu, India

**Abstract**— The Web is moving from the collection of pages toward a collection of services that interoperate through the internet. Constitution of web services is a promising approach for the integration of global enterprise applications functionalities. When a request is given, there may be multiple web services required and invoked to assemble into a solution. Hence, the composition graph generated for a given user request may have multiple candidates with various qualities at each task level. Also, several composition paths may exist to provide the same solution with different accumulated quality. The existing system solves the problem using optimal path selection using sequential web service composition. In Sequential composition, the best path selected among the multiple paths available but, it is a time consuming process. Ultimately, performance is an issue in this system. The proposed System overcomes the complexity of the existing system. Parallel web service composition approach provides the choice to the users to select the optimal path quickly. This approach is efficient in the web service composition. This is implemented by using Dijkstra with heap algorithm through BPEL process.

**Index Terms**— Web Service Composition, Parallel web service composition, Best composition path

## I. INTRODUCTION

Web services are constantly changing, and the manual composition cannot meet the actual needs of the application. There are several web services which can meet the same requirements with different quality parameters. There are many composition techniques for composing the different kinds of web services for a given user request. They are Sequential type, Parallel type, Conditional type, Cycle type. Sequential type is the basic model. Now we are implementing parallel type model using Dijkstra's and heap algorithm through BPEL process. BPEL process service component must gather information from multiple asynchronous sources, since each process can take undefined amount of time, we divide the process into parallel flow, this parallel flow can invoke many services at a same time and thus it is time efficient.

There are several techniques for finding the best web services for a given user request. Nowadays, much research

is focused on Web service composition. But composition of web services with same requirements and different quality parameters is a research problem. There are many algorithms for finding the best path among the composition path. Multiple compositions may provide same response to the user requirements. The main objective is to find the best composition of services based on the quality parameters like cost, response time, reliability and availability etc.

This paper addresses the problem of extracting the right composition path from the complete composition graph. Services which are matching each task level with different qualities are considered in this paper. In this work, we have used the parallel composition pattern. For finding the best path in the service composition graph, we use the Dijkstra's and heap algorithm through the BPEL process. The rest of the document is organized as follows. Section 2 describes previous work on web service composition. In section 3, we present a web service composition. Section 4 describes Architecture overview of proposed system. In Section 5 discuss about the Web Service Composition Implementation. Section 6 describes the Sequential Web Service Composition. Section 7 presents the modified Dijkstra algorithm. Section 8 describes the Parallel Web Service Composition. Section 9 presents the Dijkstra with heap algorithm. Section 10 presents the performance result. Section 11 presents the conclusion.

## II. RELATED WORKS

There is a considerable amount of research done in the web service composition. In this section we review some of this work. This paper [2] presents an automatic web service composition. Huge amount of computation is performed during processing. Parallel execution of processes supported in Multiprocessor platforms. In [3], it addresses the infrastructure for dynamic web service composition. The main element of the infrastructure is the composition mechanism which based on quality of service. [4] focused on the framework for parallelizing web service composition algorithms investigating how to partition the composition problem into multiple parallel threads. In paper [5], the authors have briefly discussed about the planning framework for the automated composition of web

services implemented based on the Business Process Execution Language. This Paper [6] addresses the Web Service Composition based on Reinforcement learning. Three mainly works are done in this paper. Paper [1] describes the best path from the composition graph based on the Dijkstra's Algorithm. Paper [7] describes Automatic Composition of Semantic Web Services. This Web Services developed a Composition engine. This engine employs a multi-step narrowing algorithm and implemented using a logic programming technology. In paper [8] author describes many Web Service composition algorithms based on AI planning graph. This graph can find a possible solution in polynomial time but possible with redundant Web services. In paper [9] author describes the global optimization with local selection methods allows to reduce the efforts compared to the existing system. In [10] author describes to find the best path based on genetic algorithm.

### III. WEB SERVICE COMPOSITION

The best motivating example of the web service composition is tour package planning. Assume that there are three tasks in the planning namely, ticket booking, hotel booking, rental car booking. For each task, we may have several services from several providers. The planner has to find out the best plan according to the user request. The whole problem can be formulated as a complete composition graph. W11, W12, W21, W22, W23, W31, W32 are the web services. Fig.1 shows the different combinations of services to plan for the tour package. W11, W12 are the services for ticket booking. W21, W22, W23 are services for hotel booking. W31, W32 are the services for car rental booking. W44 is the final destination node.

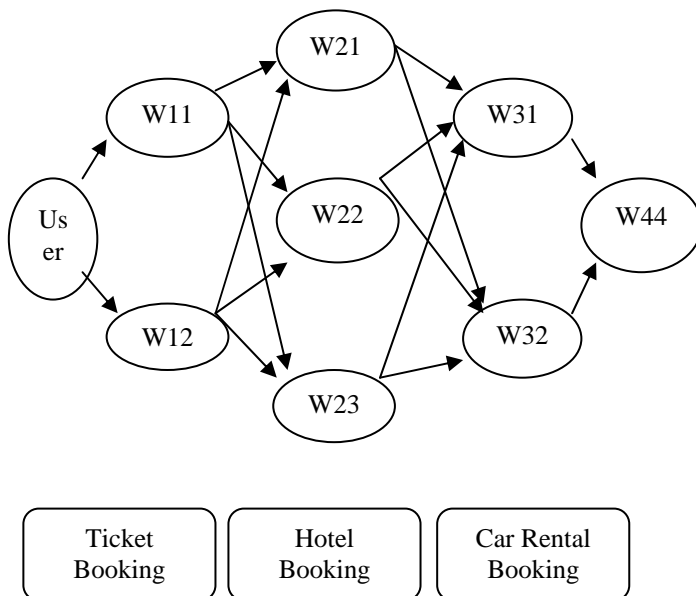


Fig 1 Web Service Composition Problem

### IV. ARCHITECTURAL DIAGRAM

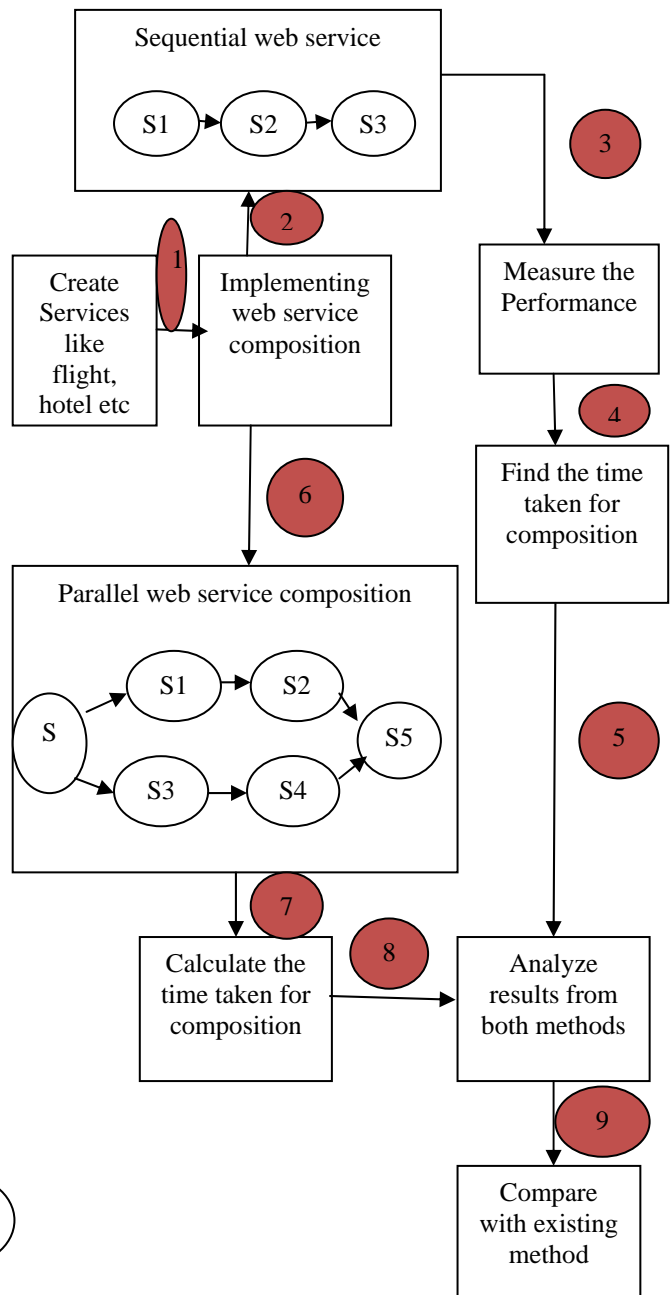


Fig 2. Architectural Diagram

In this architecture diagram, first services like hotel, flight etc are created. Each web service is used in the sequential web service composition and the performance is measured. After wards, calculate the QOS values of each service composition. The QOS factors are Cost, Response Time, Availability and Reliability. QOS factors are also used in the parallel web service composition and used to measure the performance.

The parallel web service composition is implemented using Dijkstra’s with heap algorithm through BPEL (Business Process Execution Language). Finally, the results of Sequential web service composition and Parallel Web service composition are analysed. The time taken for finding the best composition is measured and based on that the performance measurement is done.

**V. WEB SERVICE COMPOSITION IMPLEMENTATION**

Travel service composed of three atomic services called Airplane, Hotel and Rent car services. We have also determined that the first two services will be executed in parallel using a “split + join”, but the last one should be executed on a sequence form. For example, we need to execute the travel service, sending it the parameters: date-arrival, date-departure, destination-city and country. As a result we will obtain the flights, hotels, and car rental information. In order to execute the travel service, we have used OWL-S API. For a client side, we defined an endpoint called Travel as the name of our service. Continuing the execution, we invoke the travel service and the OWL-S works on executing the others services that belongs to this composition.

**VI. SEQUENTIAL WEB SERVICE COMPOSITION**

In Sequential Web Service Composition, when a request is given, there are several web services required and implored to converge into a solution. For a single task several web services from several providers with several qualities are developing day to day. There are several service selection algorithms available for discovering services for each task level in the service composition graph. Hence, the composition graph generated for a given user request may have several candidates with various qualities at each task level. Also, several composition paths may live to provide the same solution with different aggregated quality. In normal sequential flow produce the result for travelling service.

TABLE I

SEQUENTIAL COMPOSITION

Activity	Flight	Hotel	Car
A1	84	44	44
B1	33	28	33
A2	27	21	24
B2	16	22	27
A3	46	47	51
B3	78	28	34

**VII. DIJKSTRA ALGORITHM WITH HEAP FOR FINDING THE PATH**

Dijkstra’s algorithm with heap is used based on the QoS factors. Some of the QoS factors are user Preference cost, response time, availability and reliability. First the algorithm is used to find solution using sequential composition. Later the same is used to find the solution with parallel composition. Cost and response time are the QoS parameters considered in this approach.

**A) Cost**

The cost quality  $c_{ij}$  is the amount that a User requires to pay to achieve this service I using task j.

$$\sum_{i=1}^n \sum_{j=1}^m c_{ij}$$

**B) Response time**

The time quality  $t_{ij}$  measures the achieving time between the request is sent and the results are received.

$$\sum_{i=1}^n \sum_{j=1}^m t_{ij}$$

Here cost and response time has been displayed.

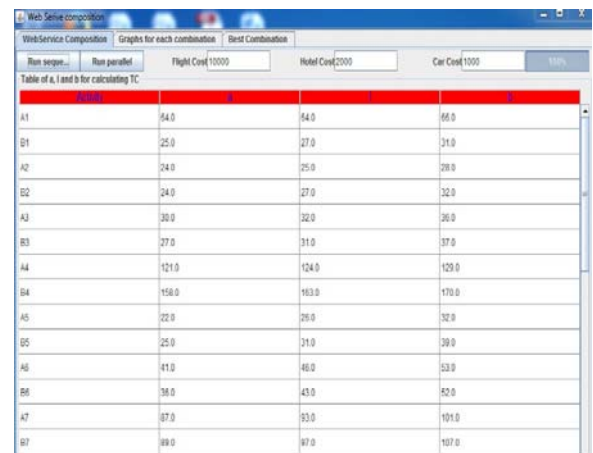


Fig 3. Time taken for Sequential Web Service Composition

After displaying the response time, the algorithm generates all possible combination of paths. These Paths are compared with the sequential composition paths. In Fig.3, all possible combinations has been displayed.

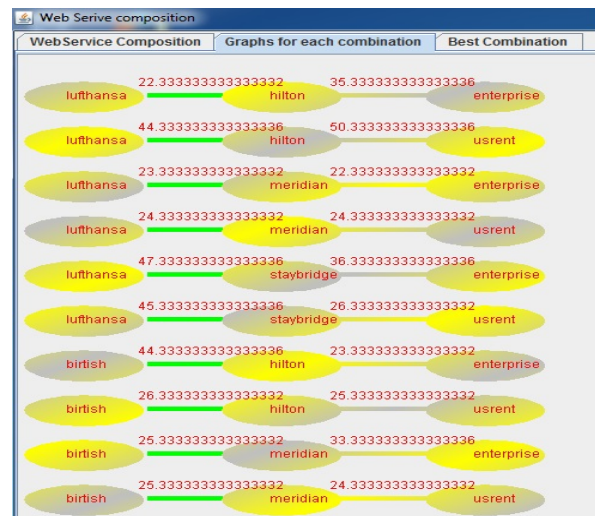


Fig 3. Possible Sequential Paths

The algorithm discover a best combination of path among all possible combination paths and the best one is shown in Fig.4.

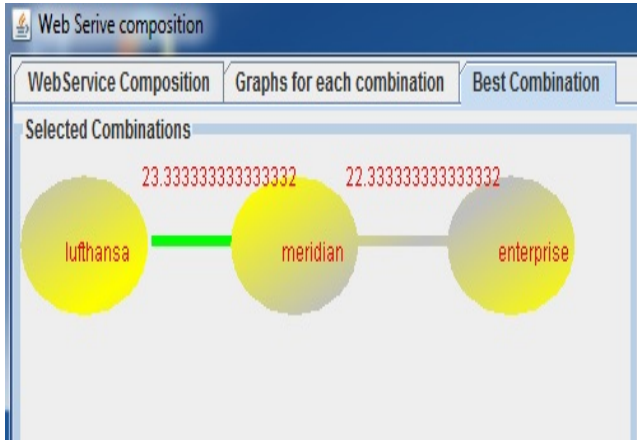


Fig 4. Best among Possible Sequential Paths

**VIII. PARALLEL WEB SERVICE COMPOSITION**

Parallel web service composition performs the operation in parallel manner. Here we are using dijkstra’s with heap algorithm through BPEL (Business Process Execution Language) Process. BPEL process service component must gather information from several asynchronous sources, since each process can take undefined amount of time, we divide the process into parallel flow, this parallel flow can invoke many services at a same time and thus it is time efficient. General flow of Parallel Web Service Composition is given in Fig 5. A1, A2, A3, A4, A5 refer the parallel tasks.

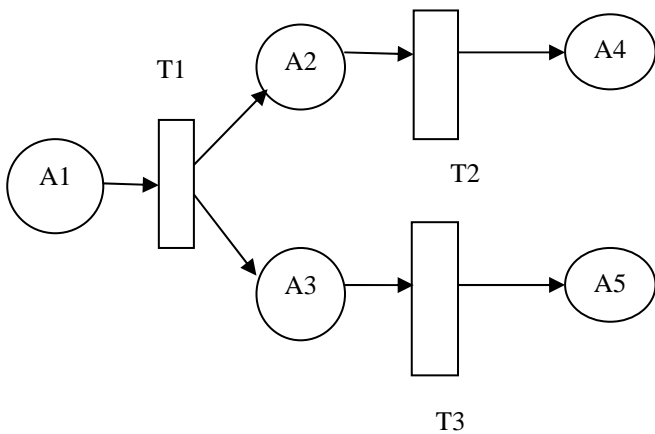


Fig 5 Example of Parallel web service composition

**IX. DIJKSTRA WITH HEAP ALGORITHM FOR FINDING THE BEST PATH**

Algorithm 1 shows the pseudo code for Dijkstra with heap. The time taken for parallel composition for the request is calculated and they are listed in Fig.6. The notations used in the algorithm are given in Table II.

**TABLE II  
NOTATIONS USED IN THE ALGORITHM.**

Symbol	Description
G	Graph
L	Location
A	Input node
N	New node
H	Heap function

**A. Algorithm 1**

1. dijkstra’s-heap (G, a){
2. First-entry= (0, a)
3. H= [first-node]
4. L= {first-node: 0}
5. Dist-so-far= {a: first-node}
6. Final-dist= { }
7. While len (dist-so-far)>0{
8. Distnode= heappopmin (H, L)
9. Final-dist [node] = dist
10. Del dist-so-far [node]}
11. for x in G[node]{
12. If x in final-dist{
13. Continue
14. New-dist= G[node] [x] + final-dist [node]
15. N= (new-dist, x)}
16. If x not in dist-so-far
17. Insert-heap (H, N, L)
18. Dist-so-far[x] = N}
19. Else if {new-entry<dist-so-far:
20. Decrease-Val (H, L, dist-so-far[x], N)}
21. Dist-so-far[x] = N
22. Return final-dist}

The Response time table is given below.

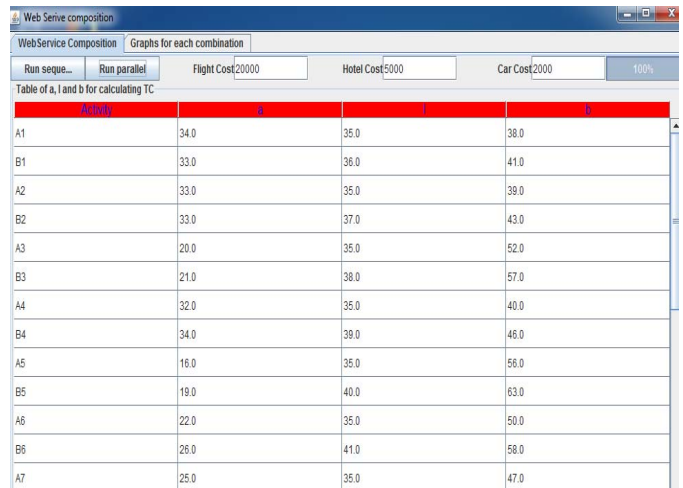


Fig 6 Time taken for Parallel Web Service Composition

Based on the Response time, all the possible combination of paths are displayed. The best paths are shown and listed in the ascending order, in case there are multiple paths with same values. Fig.7 shows the list of such paths after parallel composition. Depending on the input values the result shows that there are multiple solutions with same preferred quality. The best combination can be selected from the list(any one) which is in ascending order.

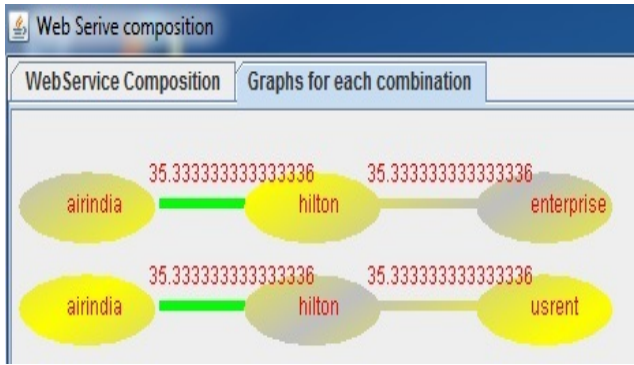


Fig 7. Best Paths Listed after Parallel

**X. PERFORMANCE ANALYSIS**

The comparison is made between the exhaustive search method and the proposed method Dijkstra’s with heap algorithm. The results obtained in the experiments are plotted in a graph with number of nodes in X-axis and the time taken for a set of nodes in Y-axis. Hence, the graph obtained as follows.

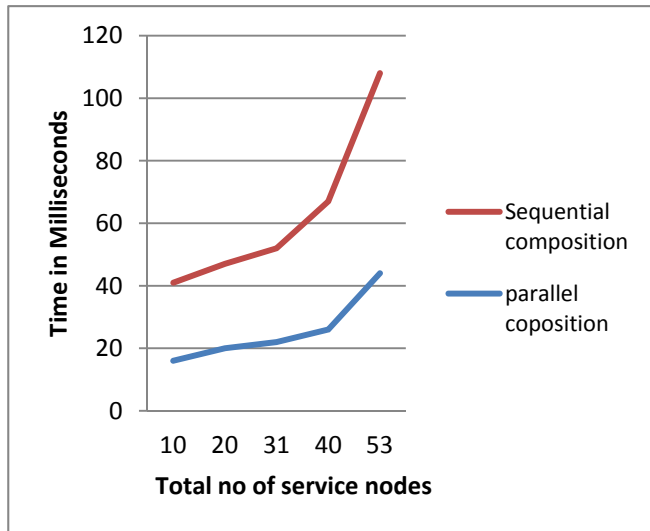


Fig 8. Performance Graph

**XI. CONCLUSION**

In this paper, we compare sequential web service composition and parallel web service composition. The result shows that the proposed method reduces time complexity. The best composition path is found by using this dijkstra’s with heap algorithm through BPEL process. BPEL process service component must gather information from multiple asynchronous sources, since each process/path can take undefined amount of time, we divide the process into parallel flow, this parallel flow can invoke many services at a same time and thus it is time efficient.

**REFERENCES**

- [1] S. Justin samuel and Dr.T.Sasipraba, "User Preferred Optimal Execution Path Selection from Quality Enabled Multi-Path Web Service Composition Graph", *ciIT International of networking and communication Engineering*, Vol 4, No 3, March 2012.
- [2] Peter Bartalos and Maria Bielikova, "Semantic Web Service Composition Framework Based on Parallel Processing" *IEEE conference on Commerce and Enterprise Computing*, 20-23 July 2009.
- [3] Frederico G, Jose M, "An Infrastructure for Evolving Dynamic Web Services Composition" *International Journal of Intelligent Computing Research (IJICR)*, Volume 1, Issue ½, March/June 2010.
- [4] Patrick Hennig, Wolf-Tilo Balke, "Highly Scalable Web Service Composition using Binary Tree –based Parallelization" *proc. of 8th International conference on web services IEEE ICWS'10, Miami, Florida, USA, July 5-10, 2010*
- [5] M. Pistore, P.Traverso and P.Bertoli, "Automated Composition of Web Services by Planning in Asynchronous Domains" *In 15th International conference on automated planning and scheduling, 2005*
- [6] Liang-Bing Feng, Masanao, Obayashi, Takashi Kuremoto, Kunikazu Kobayashi and Shun Watanabe, "QoS Optimization for Web Services Composition based on Reinforcement Learning" *International Journal of Innovative Computing, Information and Control* Volume 9, Number 6, June 2013.
- [7] Kona, Ajay, Gupta, "Automatic Composition of Semantic Web Services" *IEEE International Conference on Web Services (ICWS 2007)*.
- [8] Zheng, Yan, "An Efficient syntactic web service composition algorithm based on the planning graph model", *Proceedings of the IEEE International Conference on Web Services, 2008*, pp. 691-699.
- [9] Mohammad, Alrifai, "Combining global optimization with local selection for efficient QoS-aware service composition", *Proceeding WWW '09 Proceedings of the 18th international conference on World wide web ACM, New York, NY, USA-2009*
- [10] Honghong, Keting, Cristoforo, "Multi-path QoS-Aware Web Service Composition using Variable Length Chromosome Genetic Algorithm", *Information Technology Journal*, 2011, Volume: 10, Issue: 1, Page No.: 113-119.