International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 24 (2017) pp. 15902-15913

Analysis of User driven Availability and Resource level Availability for a Composite Web Service

S. Uma
Associate Professor, Department of Computer Applications
BMS College of Engineering, Bull Temple Road, Bangalore, India.
Orcid Id: 0000-0001-5982-984X

D. Evangelin Geetha
Associate Professor, Department of Computer Applications
MS Ramaiah Institute of Technology, MSR Nagar, Bangalore, India.
Orcid Id: 0000-0002-7253-4314

T.V. Suresh Kumar
Professor, Department of Computer Applications
MS Ramaiah Institute of Technology, MSR Nagar, Bangalore, India.
Orcid Id: 0000-0002-0945-5939

Abstract

Composition of Web services plays an important role in defining a new service in the business environment. When defining a new web service, it is very important to preserve the quality attributes as promised in Service Level Agreement (SLA). High Availability is considered as one of the most significant quality attribute in the SLA. Since the Composite web service is a combination of multiple autonomous services, it is very challenging to assure that the quality attributes will be met as per the Service Level Agreement (SLA). For a composite web service, the workflow for each service request is dynamic in nature. To guarantee the availability of Composite web service as per SLA, appropriate modeling technique is essential to compute the availability. In this paper, model based approaches to compute the availability of composite web service from the perspective of users and from the perspective of resources is proposed. Conditional Probability Tree and Markov model are employed to compute user driven availability and resource level availability respectively. The experimentation and the results of validation are very promising and provide a way to use our technique for Composite web service for proper implementation of SLA.

Keywords: Web Service Availability, Service Composition, Conditional Probability Tree, Markov Model

INTRODUCTION

Service Oriented Architecture (SOA) is a model through which services can be published, discovered and used by applications or other services. The goal of SOA is to achieve loosely coupled, standard-based and platform-independent distributed computing [1]. Web service is the technology used to implement Service Oriented Architecture. Web services have enabled the enterprises to deliver their internal core business processes as services which are accessible through the Internet [2]. Web services composition is a technique of integrating multiple atomic services into a single service to achieve a complex function. A Composite web service is an assembly of various atomic services collaborated to implement a new set of operations [3]. The composition of web services can be categorized as static and dynamic [4]. In a static composition, the binding of atomic services take place at design time. In contrast to static web service, dynamic composition involves automatic discovery, selection and binding of atomic services at run time. In order to fulfill the Service Level Agreement (SLA) of a composite web service, the availability of service is considered as one of the important quality aspect. From the qualitative point of view, Availability is defined as the ability of the system to be in the functional state when it is required for usage. From the quantitative point of view availability A(t) is a probability of the system in a functional state at an arbitrary point of time [5]. Most of the current research on service availability addresses the issues associated with the availability of the atomic services and not Composite web services. For a composite web service involving multiple atomic services, maintaining high availability is a serious source of concern. A service that is often unavailable will lead to negative impact on the service provider.

Hence, the objective of this work is to compute the availability of Composite web services with appropriate modeling techniques. The availability of the Composite web service is aggregated from the availability of the atomic services. Irrespective of the cause, unavailability or interruption of service is not bearable from the perspective of user. Web services are implemented on distributed platform with several layers of components comprising of both software and hardware. In order to comply with the assured level of availability as per SLA, equal prominence is to be given for both software and hardware. Hence, in this study the availability of composite web service is computed from two different perceptions. The two different types of availability considered in this work are,

1. *User driven Availability:* User driven availability is computed based on the user operational profile [6]. The User driven Availability is dependent on the order in which atomic services are invoked in a workflow and the probability of activation of a specific workflow. User driven availability computes the availability of the composite web service from the software point of view. The User driven availability has to take into consideration the availability of atomic services and the interactions among them to build a composite web service. Due to the non-uniformity in the workflow of each request, Conditional Probability tree is chosen for modeling the User driven availability.
2. *Resource Level Availability:* Resource level availability is based on the availability of resources used by the Composite web service [7]. It depends on the components on which the atomic services are implemented. Resource Level availability is computed based on the infrastructure used for implementing the web service. The infrastructure for a composite web service includes service composition engine, web server, database server and network for communication. Over a period of time, the failure of resources will have a significant impact on the availability of Composite web services. Due to the change in state of the functional resources that deteriorates over time, Markov model is chosen to represent the state of the system.

The computation of service availability from these two perspectives can be used to initiate preventive action if there is a mismatch in the agreed level of availability.

RELATED WORK

A good amount of research work has been carried out on Web Service Composition. For web services, maintaining the guaranteed level of QoS is a key challenge. Web service Availability is one of the significant quality aspects for a Web Service. When a single service is not capable of satisfying the given request, group of services are organized as a business process to fulfil the request. This kind of web service organization is named as web service composition [8]. The composition plan for sequencing or parallelizing the execution of various atomic web services to build a complex web service is provided in [9]. The authors focus on finding multiple composition plans and then selecting the most suitable one for each user. To address the performance bottleneck and single point of failure in centralized web service systems, a distributed dynamic service composition method based on Business Abstract Plan (BAP) is proposed in [10]. Markov model is used in Web Service Composition by various researchers with different objectives. A Markov-Chain process is a stochastic process having transition probabilities. This model helps in getting information about how to relate one state of a process to the next state. The long term trend of any process is dependent on the initial state. By changing the initial state of the system, the final result can be changed [11]. A combination of Hierarchical Task Network (HTN) planning and a Markov decision process model have been proposed for Web Service Composition in [12]. To model the problem of service composition, Partially Observable Markov Decision Processes (POMDP) integrated with reinforcement learning is proposed in [13]. Zeng et al. [14] have proposed a linear integer programming approach for identifying the QoS based best composition. A model-based Availability for evaluation of composed web services is proposed in [15]. The Architecture for highly available web services is proposed for mission critical applications in [16]. The essential idea in this work is the improvement in availability of web services by the introduction of a central hub. An availability estimation approach based on status-based user perceived service availability metric is proposed in [17]. An availability metric calculation technique based on service status is proposed in [18]. A heartbeat based mechanism is proposed for detecting the web service failure in [19]. This work aims at improving the service availability by decreasing failover time and failure detection time through service monitoring.

From the review of literature, the following research gaps have been identified:

- Most of the previous studies have addressed availability from the perspective of service provider and service requestor targeting at atomic web services and not composite web services.
- Inadequate consideration of the dynamic behaviour of composite web services and its influence on total system availability
- The failure pattern of resources over a period of time and its influence on Composite web service Availability

This study makes an attempt to bridge the gap by addressing the availability for composite web services from the user point of view and from the service provider point of view.

USER DRIVEN AVAILABILITY

There are numerous atomic services present on the web. These atomic services may not be capable to fulfill all the service requests of the users on their own. However, a collection of such atomic services can be assembled together to satisfy the request of the user. The process of assembling multiple atomic services into a single service is known as Composite web service. The group of atomic services to be used and the order of their invocation vary from one request to another request. User driven availability refers to the availability of such Composite web services. The user operational profile is used for modeling the User driven Availability. The model is built based on the different possible workflows of a Composite web service. Each workflow is defined by the set of atomic services and the probability of activation of specific service in the workflow.

Modeling of User driven Availability

A composition schema based approach is devised to model the User driven availability of composite web service. The Composition schema is represented as a Conditional probability tree. The Conditional probability tree is used due to the dynamic workflow nature of the Composite web service. The conditional probability tree has different types of nodes. The root node denotes the beginning of the tree, where a decision maker has multiple decision choice or uncertain outcomes. Decision nodes signify the choices that can be chosen by the user. The leaf nodes represent the final event for the unique choice made by the decision maker. For a Composite web service, the combination of atomic services used for each workflow is unique. Hence, for every possible workflow, it is essential to compute the availability. The possibility of dynamic workflow demands for a conditional probability tree to model the composite web service. Using this model, the availability can be computed for each possible workflow and for the entire Composite web service.

The probability tree structure is useful in computing the overall availability of the Composite web service with the following variations in the atomic service.

1. Computing the availability with single service instance for each atomic service.
2. Computing the availability with multiple instances of the atomic services.
3. Computing the availability with multiple similar atomic services.

A sample structure of the conditional probability tree is given in Figure 1. The root node represents the composite web service. The descendent nodes are the different atomic services used to realize the Composite web service. Each path from the root node to the leaf node is a unique workflow which satisfies the individual service request. *AService 1.1* is the composite web service. P_{ij} is the probability of activation of a specific atomic service in the workflow. *AService1.1.1, AService 1.1.1.1, AService 1.1.1.2* is one possible workflow for the Composite web service *AService 1.1*. The availability of the Composite web service for each workflow is computed by traversing the unique path from root to every leaf node.

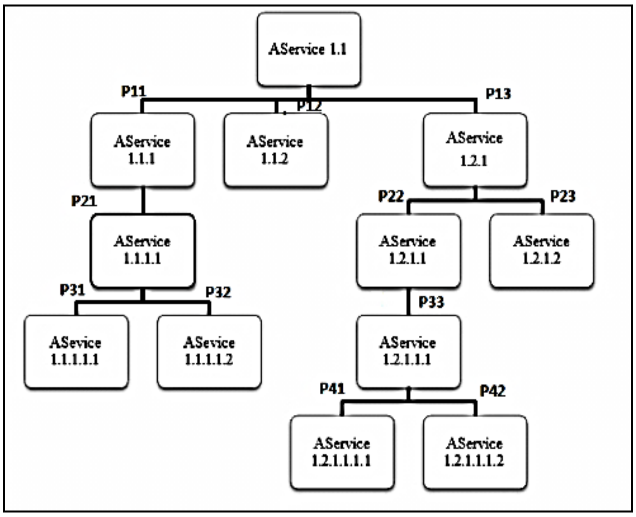

Figure 1: General Conditional Probability Tree Structure

ALGORITHM

Following is the algorithm to compute the User driven availability for each workflow using conditional probability tree. In this algorithm, *k* denotes the number of possible workflows and *n* denotes the number of atomic web services for each workflow. *m* denotes either the number of service instances of an atomic service or the number of similar atomic services. *SA(k)* is the aggregate service availability with *k* instances. *TA* is the user driven total service availability.

1. *Identify the atomic services of the Composite web service*
2. *Model the service composition schema as a conditional probability tree*
3. *For each workflow j in the conditional probability tree*

{

/ n is the number of levels in the workflow*/*

for (i =1; i <= n; i++)

{

if (number of service instance ==1)

/SA_i is the service availability in i^{th} level of the tree*/*

$SA_i = SA(1)$

else

```
{
/*m is the number of service instances/number of similar
services*/
        for (k=2; k <=m; i++)
                SA(k) = SA(k-1)+(1-SA(k-1)*SA(k);
        SAi = SA(k);
      }
   }
   Service Availability j = SA1 * SA2* SA3* ... *SAn
}
```

The User driven Composite web service Total Availability is computed using the formula

$$TA= \sum_{k=1}^{w} pj \ A(j) \quad (1)$$

where, w is the number of possible workflow

pj is the probability of activation of workflow j

A(j) is the service availability for workflow j.

The algorithm works as follows. The first step is to select the suitable atomic services to build a Composite web service. The second step deals with the formation of Conditional Probability tree structure with all possible workflows based on the choice of the user. Step 3 includes the computation of User driven Availability for every possible workflow which takes into consideration the number of available service instances or the number of available service providers for each of the atomic service in the workflow.

EXPERIMENTATION

To illustrate our model, we have considered *Process Order web service* as a Composite web service. The Composite web service is invoked by the user with a request to order goods. In response to this call, the composite web service generates a call to *ChecklocalStock* service and determines whether the requested goods are available in the stock. If the stock is available, the supplier approves the order to the customer, by calling *ConfimOrderService*. If the stock is not available, the supplier invokes warehouse web service *SearchExternalSupplier* to check whether the goods are present at that location. If ship is available, then the supplier sends confirmation to the customer by invoking *SendConfirmOrder* Service. Otherwise, the *CancelOrderService* is invoked and order cancellation message is sent as a reply to the user. The Conditional Probability tree for the given case study is depicted in Figure 2.

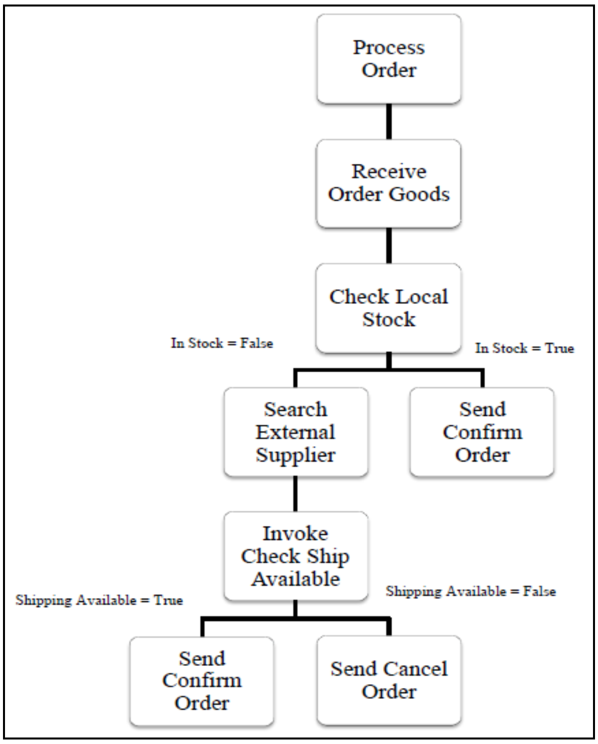

Figure 2: Conditional Probability tree for Process Order Web Service

RESULT AND DISCUSSION

Three possible workflows have been identified for the Process Order Composite web service. Workflow 1 comprises of *Receive Order Goods, Check Local Stock, Search External Supplier, Check Ship Available* and *Send Confirm Order* atomic services for the *Process Order* Composite service. Workflow 2 includes *Receive Order Goods, Check Local Stock* and *Send Confirm Order atomic services.* Workflow 3 involves *Receive Order Goods, Check Local Stock, Search External Supplier, Check Ship Available* and *Send Cancel Order* atomic services. For all the three workflows, the Service availability is computed with single instance and multiple instances of atomic services using the algorithm given in Section 3.2. Further, Service availability is computed with single service and multiple similar services for each workflow. Based on the availability of each workflow, the User driven total service availability is computed for the Composite web service. The availability of five-nines (99.999%) is considered as high availability for services [20]. Table 1 through Table 11 shows the different availability values of atomic services and the User driven total availability computed for Process Order Composite web service.

Table 1: Availability for Work Flow 1

AVAILABILITY FOR WORKFLOW 1(%)				
	1 instance	**2 Instances**	**3 Instances**	**4 Instances**
Receive Order Goods	98.20873	99.9679135	99.9994252	99.9999897
Check Local Stock	98.21478	99.96813	99.9994311	99.9999898
Search External Supplier	98.64843	99.9817327	99.9997531	99.9999967
Check Ship Available	99.33705	99.995605	99.9999709	99.9999998
Send Confirm Order	98.24127	99.9690688	99.999456	99.9999904
PROCESS ORDER AVAILABILITY	**92.858658**	**99.8825023**	**99.9980363**	**99.9999665**

Table 2: Availability with different service providers for workflow 1

AVAILABILITY WITH DIFFERENT SERVICE PROVIDERS(%) - (Workflow 1)				
	Service Provider 1	**Service Provider 2**	**Service Provider 3**	**Service Provider 4**
Receive Order Goods	98.20873	98.70675	99.59343	98.28197
Check Local Stock	98.21478	97.66385	99.03958	98.98417
Search External Supplier	98.64843	98.42405	97.70279	99.74428
Check Ship Available	99.33705	97.80866	98.80944	98.93907
Send Confirm Order	98.24127	99.34178	98.79190	99.71866

Table 3: Total Availability for Workflow 1

COMPUTED TOTAL AVAILABILITY WITH DIFFERENT SERVICE PROVIDERS (%) - (Workflow 1)				
	Service Provider 1	**Service Provider 2**	**Service Provider 3**	**Service Provider 4**
Receive Order Goods	98.20873	99.9768344	99.9999058	99.9999984
Check Local Stock	98.21478	99.9582947	99.9995995	99.9999959
Search External Supplier	98.64843	99.9787	99.9995107	99.9999987
Check Ship Available	99.33705	99.9854725	99.999827	99.9999982
Send Confirm Order	98.24127	99.9884237	99.9998601	99.9999996
PROCESS ORDER AVAILABILITY	**92.85866**	**99.8877729**	**99.9987032**	**99.9999908**

The inference from the computation of user driven availability for the process order Composite web service with workflow 1 is as follows:

i) If there is a single web service provider, then the user driven total availability with single instance of each atomic service is 92.858658% which is inadequate to fulfill the required User driven total Availability. Hence, it is necessary to use either 3 instances or 4 instances of atomic services, so that the user driven total availability reaches above 99%

ii) If there are multiple web service providers, then the user driven total availability with single service provider of each atomic service is 92.858658% which is very low. With the aggregate of 4 Service Providers the user driven total availability reaches above 99.9999908% which is considered as the desirable value.

Table 4: Availability for Work Flow 2

AVAILABILITY FOR WORKFLOW 2 (%)				
	1 instance	**2 Instances**	**3 Instances**	**4 Instances**
Receive Order Goods	98.20873	99.9679135	99.9994252	99.9999897
Check Local Stock	98.21478	99.96813	99.9994311	99.9999898
Send Confirm order	98.24127	99.9690688	99.999456	99.9999904
PROCESS ORDER AVAILABILITY	**94.759103**	**99.9051423**	**99.9983123**	**99.99997**

Table 5: Availability with different service providers for workflow 2

AVAILABILITY WITH DIFFERENT SERVICE PROVIDERS(%) - (Workflow 2)				
Availability	**Service Provider 1**	**Service Provider 2**	**Service Provider 3**	**Service Provider 4**
Receive Order Goods	98.20873	98.70675	99.59343	98.28197
Check Local Stock	98.21478	97.66385	99.03958	98.98417
Send Confirm Order	98.24127	99.34178	98.79190	99.71866

Table 6: Total Availability for Workflow 2

COMPUTED TOTAL AVAILABILITY WITH DIFFERENT SERVICE PROVIDERS(%) - (Workflow 2)				
	Service Provider 1	**Service Provider 2**	**Service Provider 3**	**Service Provider 4**
Receive Order Goods	98.20873	99.9768344	99.9999058	99.9999984
Check Local Stock	98.21478	99.9582947	99.9995995	99.9999959
Send Confirm Order	98.24127	99.9884237	99.9998601	99.9999996
PROCESS ORDER AVAILABILITY	**94.759103**	**99.92357**	**99.9993654**	**99.9999939**

The observation from the computation of user driven availability for the process order Composite web service with workflow 2 is as follows:

i) If there is a single web service provider, then the user driven total availability with single instance of each atomic service is 94.759103% which is not satisfactory. The user driven total availability reaches above 99% even with 2 service instances of atomic services. It is suggested to use 3 or 4 instances of atomic services, so that availability with five 9s is achieved.

ii) If there are multiple web service providers, then the user driven total availability with single service provider of each atomic service is 94.759103% which is unacceptable from the point of view of users. With the aggregate of 3 Service Providers the user driven total availability improves above 99.999% which is an ideal value.

Table 7: Availability for Work Flow 3

AVAILABILITY FOR WORKFLOW 3(%)				
	1 instance	**2 Instances**	**3 Instances**	**4 Instances**
Receive Order Goods	98.20873	99.9679135	99.9994252	99.9999897
Check Local Stock	98.21478	99.96813	99.9994311	99.9999898
Search External Supplier	98.64843	99.9817327	99.9997531	99.9999967
Check Ship Available	99.33705	99.995605	99.9999709	99.9999998
Send Cancel Order	99.59371	99.9983493	99.9999933	100
PROCESS ORDER AVAILABILITY	**94.136994**	**99.9117574**	**99.9985736**	**99.999976**

Table 8: Availability with different service providers for workflow 3

AVAILABILITY WITH DIFFERENT SERVICE PROVIDERS(%) - (Workflow 3)				
	Service Provider 1	**Service Provider 2**	**Service Provider 3**	**Service Provider 4**
Receive Order Goods	98.20873	98.70675	99.59343	98.28197
Check Local Stock	98.21478	97.66385	99.03958	98.98417
Search External Supplier	98.64843	98.42405	97.70279	99.74428
Check Ship Available	99.33705	97.80866	98.80944	98.93907
Send Cancel Order	99.59371	98.42233	98.26772	99.22667

Table 9: Total Availability for Workflow 3

COMPUTED TOTAL AVAILABILITY WITH DIFFERENT SERVICE PROVIDERS(%) - (Workflow 3)				
	Service Provider 1	**Service Provider 2**	**Service Provider 3**	**Service Provider 4**
Receive Order Goods	98.20873	99.9768344	99.9999058	99.9999984
Check Local Stock	98.21478	99.9582947	99.9995995	99.9999959
Search External Supplier	98.64843	99.9787	99.9995107	99.9999987
Check Ship Available	99.33705	99.9854725	99.999827	99.9999982
Send Cancel Order	99.59371	99.9935901	99.999889	99.9999991
PROCESS ORDER AVAILABILITY	**94.13699**	**99.8929342**	**99.998732**	**99.9999904**

The implication from the computation of user driven availability for the process order Composite web service with workflow 3 is as follows:

i) When there is a single web service provider, the user driven total availability with single instance of each atomic service is 94.13699% which is inadequate. It is worth to have either 3 instances or 4 instances of atomic services, so that the user driven total availability rises above 99.99%

ii) When there are multiple web service providers, the user driven total availability with single service provider of each atomic service is 93.91825% which is very poor. With the aggregate of 4 Service Providers the ideal value of 5 nines is attained for user driven total availability.

Another important inference from all the three Workflows of Composite web service is that the high availability of atomic services need not result in high availability of user driven total availability for a Composite web service. As a proof, when considering workflow 1 with single service instance, the availability of all the atomic services are above 98%, but the User driven total availability for composite service is less than 93%. Hence, composition schema represented through Conditional probability tree is a very promising model to compute the User driven Availability of a Composite web service. The computation aids in maintaining the Availability as per the SLA by deciding the appropriate number of service instances or service providers for the Composite web service.

The computed User driven Composite web service total availability with multiple instances and multiple services similar atomic services respectively are tabulated in Table 10 and Table 11. This computation aids in determining the number of atomic service instances to be used or the number of similar services to be used for maintaining the guaranteed level of Composite Service Availability as per the SLA. From the results, it is found that with three workflows, minimum 4 instances of atomic services are required to maintain the ideal value for User driven total service availability. To maintain the ideal five 9s of User driven total service availability with different service providers minimum three service providers are needed.

Table 10: User driven Total Service Availability with multiple instances

USER DRIVEN TOTAL SERVICE AVAILABILITY (%)				
	SINGLE INSTANCE	**2 INSTANCES**	**3 INSTANCES**	**4 INSTANCES**
WORKFLOW 1	92.858658	99.8825023	99.9980363	99.9999665
WORKFLOW 2	94.759103	99.9051423	99.9983123	99.99997
WORKFLOW 3	94.136994	99.9117574	99.9985736	99.999976
TOTAL AVAILABILITY FOR COMPOSITE SERVICE	**93.9182517**	**99.8998007**	**99.9983074**	**99.9999708**

Table 11: User driven Total Service Availability with multiple similar services

USER DRIVEN TOTAL SERVICE AVAILABILITY (%)				
	SERVICE PROVIDER 1	**SERVICE PROVIDER 2**	**SERVICE PROVIDER 3**	**SERVICE PROVIDER 4**
WORKFLOW 1	92.85866	99.92357	99.9993654	99.9999939
WORKFLOW 2	94.75910	99.92357	99.9993654	99.9999939
WORKFLOW 3	94.13699	99.8929342	99.998732	99.9999904
TOTAL AVAILABILITY FOR COMPOSITE SERVICE	**93.91825**	**99.9133581**	**99.9991543**	**99.9999927**

From the results, it is proven that the composition schema represented through Conditional probability tree is suitable for computing the User driven availability for every possible workflow. The model based approach also has helped in computing the enhancement in User driven availability by adding

- New service instances for each atomic service.
- Service instances for each atomic service and its effect on the overall composite web service availability.
- New service with the same interface for each atomic service and its dominance on the overall composite web service availability.

Resource Level Availability

Resource level availability is the availability which is computed based on the resources used by the composite web service. Resource Level Availability depends on the resources of the system through which the service is provided. Web service composition engine, web server and database server are three main resources considered for computing the availability of the web service. The composite service is implemented at the middleware level by the Service composition engine. The execution of the composite service is driven by the composition engine [21]. The service composition engine provides the run-time environment to execute the web service business logic by invoking other services. Composite services could be executed in multiple engines to ensure high availability. Web servers receive the service requests, process the request and provide the required response to the users. The web servers are responsible for implementing the functions requested by the users. Web servers are vital in terms of transferring data between the presentation layer and database servers. The web service layer handles the collection and transformation of data in both the directions. The database servers are used for persistent storage of data. The data related operations of a web service are handled by the data base server.

The atomic services of the composite web service use a number of resources to fulfill the service request. Resource Level Availability is modeled based on the resources using which the Composite web service is provided. The resources required for implementing the web service are service composition engine, web servers, database servers and network for communication. In this work, Resource level availability is modeled based on the assumption that the network is always available for communication and no failure occurs in the network. To offer a highly available web service, a redundant architecture with a number of service composition engines, web servers and database servers are essential. Each atomic web service can be hosted on a dedicated computer. The service will be available even if only one service composition engine, one web server and one database server is available.

Modeling of Resource Level Availability

To model the Resource Level Availability of a composite web service, Continuous-Time Markov Chain (CTMC) model is proposed. Markov Model [22, 23] is a stochastic model which is a function of system's state and time. The behavior of the model is described using a state diagram. The Markov model is chosen to compute composite web service availability because of its ability to capture the interdependencies and dynamic relationships between the system resources. Markov model is a proven technique to model the complex attributes of the system behavior [24]. The description of the CTMC model is as follows: $\{X(t) : t \geq 0\}$ is a Continuous-Time Markov chain with state space S and Generator matrix $Q = [q_{ij}]$. q_{ij} is the transition rate from state i to state j if $i \neq j$ and $q_{ii} = -q_i = -\sum_{i \neq j} q_{ij}$. $\pi(t)$ is the transient state probability vector and $\pi(0)$ is the initial probability vector. The transient behavior is defined by the Kolmogorov differential equation (KDE) as

$$\frac{d}{dt}\pi(t) = Q\ \pi(t), \text{ given } \pi(0) \qquad (2)$$

The instantaneous availability of the system $A(t) = \sum \pi_{(i,j,k)}$, where $\pi_{(i,j,k)}$ is the probability of the system to be in state S_{ijk}. The main entities of Markov model are system state and state transition. System state characterizes the state of the system. The state transition represents the change in state as a result of occurrence of an event. In our model, the system state represents whether the resource is functional or non-functional. The state can keep track of the number of functional resources. The state transition represents the failure rate of the system resource which could be for a service composition engine, web server, or database server. When CTMC reaches a failure state, no further transitions are possible.

To illustrate the suitability of CTMC for web service availability modeling, a system having p service composition engines, m web servers and n data base servers is considered to handle the service requests. Each service request is processed by one of the free service composition engine, one of the web server and by one of the database server. The system is considered to be available, if at least one composition service engine, one web server and one database server is in functional state. The resource failure rates are distributed exponentially. λ_C is the failure rate of service composition engine, λ_S is the failure rate for web server and λ_D is the failure rate for the database server measured for one year. Each state S_{ijk} denotes the system with i functional service composition engines, j functional web servers and k functional data base servers. For example, S_{232} signifies that 2 service composition engines, 3 web servers and 2 database servers are in functional state.

RESULTS AND DISCUSSION

Figure 3 represents the Markov model with 2 service composition engines, 2 web servers and 2 database servers. The system is considered to be available in the following states, S = (222, 221, 212, 122, 211, 121, 112, 111). These eight states are considered as functional states of the system. The states, S' = (220, 202, 022, 210, 201, 120, 021, 012,102, 011, 101, 110) denotes the unavailability of any one of the service composition engine, the web server or the database server. In these twelve states, the service request cannot be processed by the system.

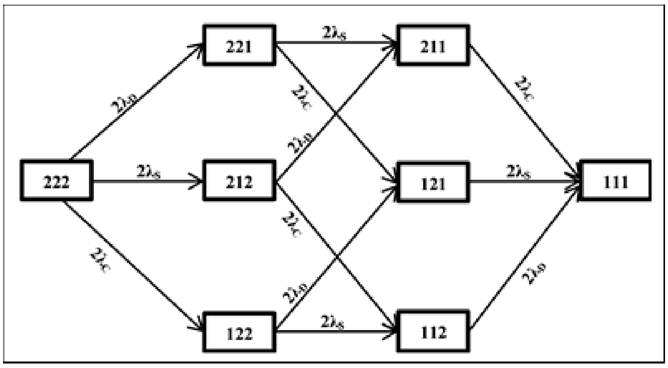

Figure 3: Markov availability model with 8 states

The generator matrix Q for the above Markov model with 8 functional states is as follows:

$$\begin{pmatrix} -(2\lambda C+2\lambda S+2\lambda D) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2\lambda D & -(2\lambda C+2\lambda S+\lambda D) & 0 & 0 & 0 & 0 & 0 & 0 \\ 2\lambda S & 0 & -(2\lambda C+\lambda S+2\lambda D) & 0 & 0 & 0 & 0 & 0 \\ 2\lambda C & 0 & 0 & -(\lambda C+2\lambda S+2\lambda D) & 0 & 0 & 0 & 0 \\ 0 & 2\lambda S & 2\lambda D & 0 & -(2\lambda C+\lambda S+\lambda D) & 0 & 0 & 0 \\ 0 & 2\lambda C & 0 & 2\lambda D & 0 & -(\lambda C+2\lambda S+\lambda D) & 0 & 0 \\ 0 & 0 & 2\lambda C & 2\lambda S & 0 & 0 & -(\lambda C+\lambda S+2\lambda D) & 0 \\ 0 & 0 & 0 & 0 & 2\lambda D & 2\lambda C & 2\lambda S & -(\lambda C+\lambda S+\lambda D) \end{pmatrix} \quad (3)$$

From the generator matrix Q, the system of differential Equations is formed based on Kolmogorov differential equation for all the functional states of the system.

$$\pi_{222}'(t) = -(2\lambda_C+2\lambda_S+2\lambda_D)\pi_{222} \quad (4)$$

$$\pi_{221}'(t) = 2\lambda_D\pi_{222} - (2\lambda_C+2\lambda_S+\lambda_D)\pi_{221} \quad (5)$$

$$\pi_{212}'(t) = 2\lambda_S\pi_{222} - (2\lambda_C+\lambda_S+2\lambda_D)\pi_{212} \quad (6)$$

$$\pi_{122}'(t) = 2\lambda_C\pi_{222} - (\lambda_C+2\lambda_S+2\lambda_D)\pi_{122} \quad (7)$$

$$\pi_{211}'(t) = 2\lambda_S\pi_{221} + 2\lambda_D\pi_{212} - (2\lambda_C+\lambda_S+\lambda_D)\pi_{211} \quad (8)$$

$$\pi_{121}'(t) = 2\lambda_C\pi_{221} + 2\lambda_D\pi_{122} - (\lambda_C+2\lambda_S+\lambda_D)\pi_{121} \quad (9)$$

$$\pi_{112}'(t) = 2\lambda_C\pi_{212} + 2\lambda_S\pi_{122} - (\lambda_C+\lambda_S+2\lambda_D)\pi_{112} \quad (10)$$

$$\pi_{111}'(t) = 2\lambda_C\pi_{211} + 2\lambda_S\pi_{121} + 2\lambda_D\pi_{112} - (\lambda_C+\lambda_S+\lambda_D)\pi_{111} \quad (11)$$

By solving the above differential equations, with $\lambda_C = 0.001$, $\lambda_S = 0.0025$, $\lambda_D = 0.001$ and with initial condition $\pi_{222}(0) = 1$, the computed total system availability A(t) in the time period t = 0 to 12 with the step size 1 is shown in Figure 4. It is observed that the total Resource level availability deteriorates over a period of time. In the time interval 0 to 2 the total system availability is maintained with 100% and subsequently the availability is declining gradually. This analysis is useful in deciding the number of resources to be employed for maintaining the promised level of Composite web service availability.

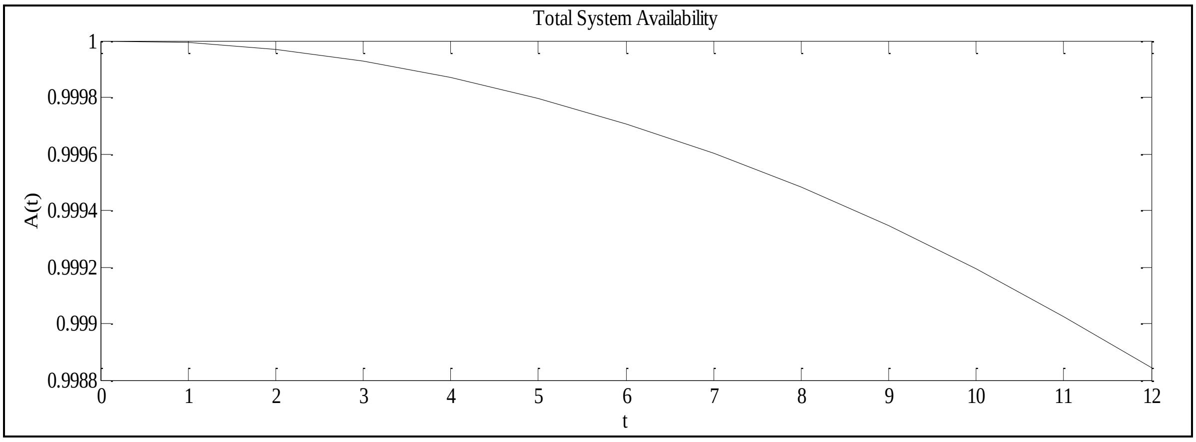

Figure 4: Resource level Total System Availability (8 states)

Sensitivity Analysis

To study the impact on total resource level availability with an additional web server to the existing system, the system is modeled with 2 service composition engines, 3 web servers and 2 data base servers. Figure 5 represents the Markov model for this system. With this model, the system is considered to be available in the following 12 functional states, S = (232, 231, 222,132,131, 221, 122, 212,121, 211, 112, 111). The states, S' = (230, 032, 130, 220, 022, 202, 210, 021, 120, 201,012, 102, 011, 101, 110) denotes the unavailability of any one of the service composition engine, the web server or the database server. In these fifteen states, the service request cannot be handled by the system.

From the generator matrix Q, the system of differential Equations is formed using Kolmogorov differential equation for all the functional states of the system.

The generator matrix Q for the above Markov model with 12 functional states is as follows:

$$\begin{pmatrix} -(2\lambda C+3\lambda S+2\lambda D) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2\lambda D & -(2\lambda C+3\lambda S+\lambda D) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3\lambda S & 0 & -(2\lambda C+2\lambda S+2\lambda D) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2\lambda C & 0 & 0 & -(\lambda C+3\lambda S+2\lambda D) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2\lambda C & 0 & 2\lambda D & -(\lambda C+3\lambda S+\lambda D) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3\lambda S & 2\lambda D & 0 & 0 & -(2\lambda C+2\lambda S+\lambda D) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2\lambda C & 3\lambda S & 0 & 0 & -(\lambda C+2\lambda S+2\lambda D) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2\lambda S & 0 & 0 & 0 & 0 & -(2\lambda C+\lambda S+2\lambda D) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3\lambda S & 2\lambda C & 2\lambda D & 0 & -(\lambda C+2\lambda S+\lambda D) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2\lambda S & 0 & 2\lambda D & 0 & -(2\lambda C+\lambda S+\lambda D) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2\lambda S & 2\lambda C & 0 & 0 & -(\lambda C+\lambda S+2\lambda D) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2\lambda S & 2\lambda C & 2\lambda D & -(\lambda C+\lambda S+\lambda D) \end{pmatrix} \quad (12)$$

$$\pi_{232}'(t) = -(2\lambda_C+3\lambda_S+2\lambda_D)\,\pi_{232}(t) \quad (13)$$

$$\pi_{231}'(t) = 2\lambda_D\pi_{232}(t) - (2\lambda_C+3\lambda_S+\lambda_D)\,\pi_{231}(t) \quad (14)$$

$$\pi_{222}'(t) = 3\lambda_S\pi_{232}(t) - (2\lambda_C+2\lambda_S+2\lambda_D)\,\pi_{222}(t) \quad (15)$$

$$\pi_{132}'(t) = 2\lambda_C\pi_{232}(t) - (\lambda_C+3\lambda_S+2\lambda_D)\,\pi_{132}(t) \quad (16)$$

$$\pi_{131}'(t) = 2\lambda_C\pi_{231}(t) + 2\lambda_D\pi_{132}(t) - (\lambda_C+3\lambda_S+\lambda_D)\,\pi_{131}(t) \quad (17)$$

$$\pi_{221}'(t) = 3\lambda_S\pi_{231}(t) + 2\lambda_D\pi_{222}(t) - (2\lambda_C+2\lambda_S+\lambda_D)\,\pi_{221}(t) \quad (18)$$

$$\pi_{122}'(t) = 2\lambda_C\pi_{222}(t) + 3\lambda_S\pi_{132}(t) - (\lambda_C+2\lambda_S+2\lambda_D)\,\pi_{122}(t) \quad (19)$$

$$\pi_{212}'(t) = 2\lambda_S\pi_{222}(t) - (2\lambda_C+\lambda_S+2\lambda_D)\pi_{212}(t) \quad (20)$$

$$\pi_{121}'(t)=3\lambda_S\pi_{131}(t)+2\lambda_C\pi_{221}(t)+2\lambda_D\pi_{122}(t)-(\lambda_C+2\lambda_S+\lambda_D)\pi_{121}(t) \quad (21)$$

$$\pi_{211}'(t) = 2\lambda_S\pi_{211}(t) + 2\lambda_D\pi_{212}(t) - (2\lambda_C+\lambda_S+\lambda_D)\pi_{211}(t) \quad (22)$$

$$\pi_{112}'(t) = 2\lambda_S\pi_{122}(t) + 2\lambda_C\pi_{212}(t) - (\lambda_C+\lambda_S+2\lambda_D)\pi_{112}(t) \quad (23)$$

$$\pi_{111}'(t) = 2\lambda_S\pi_{121}(t) + 2\lambda_C\pi_{211}(t) + 2\lambda_D\pi_{112}(t) - (\lambda_C+\lambda_S+\lambda_D)\pi_{111}(t) \quad (24)$$

By solving the above differential equations, with $\lambda_C = 0.001$, $\lambda_S = 0.0025$, $\lambda_D = 0.001$ and with initial condition $\pi_{232}(0) = 1$, the total system availability A(t) in the time period t = 0 to 12 is shown in Figure 6. It is witnessed that the total Resource level availability in the time interval 0 to 12 is progressively reducing, but still the availability is maintained above 99.9%. The addition of a single web server has greatly improved the total system availability. This study helps in changing the number of resources based on the assured level of availability as per the SLA.

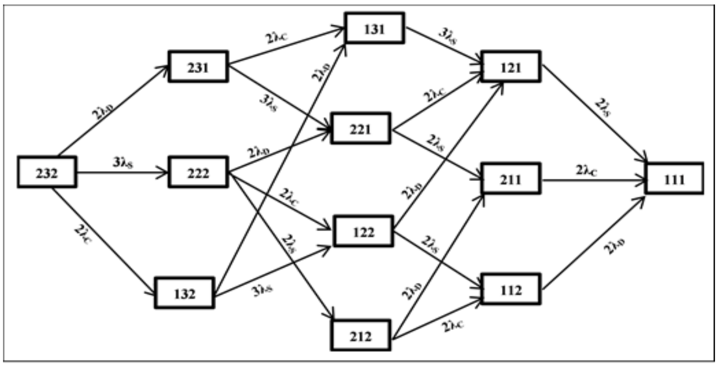

Figure 5: Markov availability model

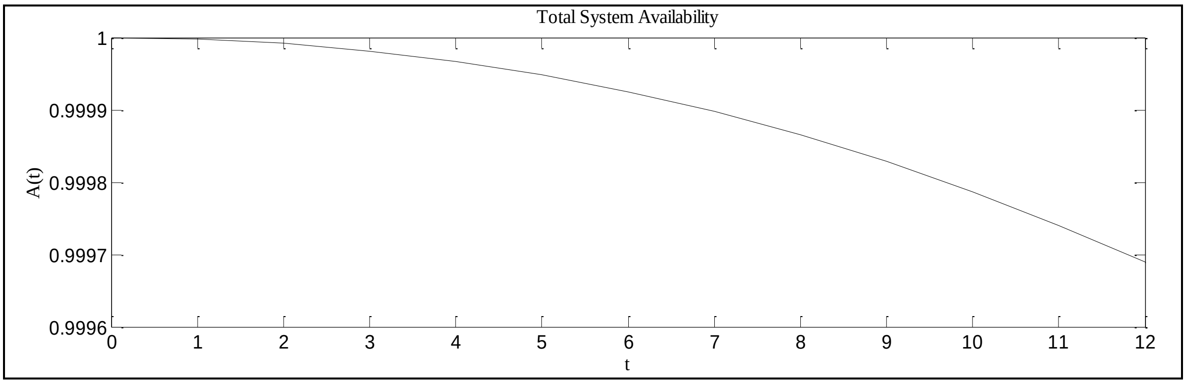

Figure 6: Resource level Total System Availability

Table 12: Comparison of Resources Availability with difference in the number of resources

A(t)	Resource Level Availability with 2 service composition engines, 2 web servers and 2 database servers (%)	Resource Level Availability with 2 service composition engines, 3 web servers and 2 database servers (%)
t = 0	100	100
t = 1	100	100
t = 2	100	100
t = 3	99.99	100
t = 4	99.99	100
t = 5	99.98	99.99
t = 6	99.97	99.99
t = 7	99.96	99.99
t = 8	99.95	99.99
t = 9	99.93	99.98
t = 10	99.92	99.98
t = 11	99.90	99.98
t = 12	99.88	99.97

Adequate sizing of infrastructure is one of the key factors for maintaining the promising level of service availability. Computation of Resource Level Availability through Markov modelling addresses this key issue by finding the trade-offs in terms of number of resources to be used for achieving the promised level of availability for the Composite web service. Sensitivity analysis is done to find the impact on the service availability based on the number of resources. Table 12 shows the comparison of total service availability in the time interval 0 to 12, with the initial condition of 2 functional service composition engines, 2 functional webservers, 2 functional database servers and with 2 functional service composition engines, 3 functional webservers and 2 functional database servers. The failure rate considered for service composition engine $\lambda_C = 0.001$ web server $\lambda_S = 0.0025$, and the failure rate for the database server $\lambda_D = 0.001$. For the first configuration, the total system availability computed per year is 99.88 which mean the system is unavailable for 10 hours 31 minutes and 8.3 seconds [25]. For the second configuration the total system availability per year is 99.97 which mean the system is unavailable only for 2 hours 37 minutes and 47.1 seconds. From this comparison, we observe that addition of a single web server to the existing infrastructure results in a significant improvement in the total system availability. It is evident from the results that Markov model is an appropriate model for determining the number of resources to be employed in order to maintain the total system availability as per the SLA.

CONCLUSION

Availability is one of the most vital quality aspects of the Web service. In this paper, we have presented two types of availability modeling for the composite web service. User driven Availability and Resource level availability are modeled for the composite web service. The composition schema modeled through Conditional probability tree facilitates in deciding the number of atomic service instances to achieve the promised User driven availability. This approach further enables in deciding the addition of similar atomic services and its impact on the total availability of composite web service. Resource level availability modeled through Continuous Time Markov chain is beneficial in assessing the availability of the web service based on the number of functional resources. Using this model, sensitivity analyses has been conducted in order to decide the number resources of each type and its impact on the total system availability of the web service. The results achieved from the two models will benefit in finding the correct tradeoff to achieve the total service availability of a composite web service as expected by the user.

REFERENCES

[1] Mike P. Papazoglou, Willem-Jan Heuvel, Service oriented architectures: approaches, technologies and research issues, VLDB J. 16 (3) (2007) 389–415.

[2] Sheng, Quan Z., et al. "Web services composition: A decade’s overview." Information Sciences, 280 (2014): 218-238.

[3] Boualem Benatallah, Marlon Dumas, Quan Z. Sheng, Facilitating the rapid development and scalable orchestration of composite web services, Distrib.Parallel Dat. 17 (1) (2005) 5–37.

[4] Alonso, Gustavo, et al. Web services: concepts, architectures and applications. Springer Science & Business Media, 2013.

[5] Eusgeld, Irene, et al. "Hardware reliability." Dependability metrics. Springer Berlin Heidelberg, 2008. 59-103.

[6] Martinello, Magnos. "Availability Modeling and Evaluation of Web-based Services-A pragmatic approach." PhD diss., Institut National Polytechnique de Toulouse, 2005.

[7] Peltz, Chris. "Web services orchestration and choreography." Computer 36.10 (2003): 46-52.

[8] Barakat, Lina, et al. "Efficient multi-granularity service composition." Web Services (ICWS), 2011 IEEE International Conference on. IEEE, 2011.

[9] Chiplunkar, Niranjan N. "Necessity of dynamic composition plan for web services." Applied and Theoretical Computing and Communication

Technology (iCATccT), 2015 International Conference on. IEEE, 2015.

[10] Zhao, Minglei, Wendong Zhao, and Laixian Peng. "BAPAWSC: A new dynamic web service composition algorithm in distributed environments." Information Science and Technology (ICIST), 2013 International Conference on. IEEE, 2013.

[11] Kemeny, John-G., J. Snell, and Knapp Laurie. "Denumerable Markov chains." (1967).

[12] Chen, Kun, Jiuyun Xu, and Stephan Reiff-Marganiec. "Markov-htn planning approach to enhance flexibility of automatic web service composition." Web Services, 2009. ICWS 2009. IEEE International Conference on. IEEE, 2009.

[13] Lei, Yu, et al. "Dynamic Web Service Composition Based on State Space Searching." Parallel and Distributed Systems (ICPADS), 2015 IEEE 21st International Conference on. IEEE, 2015.

[14] Zeng, Liangzhao, et al. "Qos-aware middleware for web services composition." IEEE Transactions on software engineering 30.5 (2004): 311-327

[15] Iacono, Mauro, and Stefano Marrone. "Model-based availability evaluation of composed web services." Journal of Telecommunications and Information Technology 4 (2014): 5.

[16] Abraham, Subil, Mathews Thomas, and Johnson Thomas. "Enhancing web services availability." e-Business Engineering, 2005. ICEBE 2005. IEEE International Conference on. IEEE, 2005.

[17] Shao, Lingshuang, et al. "User-perceived service availability: A metric and an estimation approach." Web Services, 2009. ICWS 2009. IEEE International Conference on. IEEE, 2009.

[18] Sandesh Tripathi, S Q Abbas, Rizwan Beg, International Journal on Web Service Computing (IJWSC), Vol.2, No.3, September 2011.

[19] Tayyaba Anees, Heimo Zeilinger. "SOA Model for High Availability of Services" The Sixth International Conference on Communication Theory, Reliability, and Quality of Service (2013), 70-75

[20] Drysdale, David. High-Quality Software Engineering. Lulu. com, 2007.

[21] Li, Wubin, et al. "Execution optimization for composite services through multiple engines." International Conference on Service-Oriented Computing. Springer, Berlin, Heidelberg, 2007.

[22] Shooman, M.L., Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design. 2003: Wiley.

[23] Trivedi, Kishor S. Probability & statistics with reliability, queuing and computer science applications. John Wiley & Sons, 2008.

[24] Reibman, Andrew, Roger Smith, and Kishor Trivedi. "Markov and Markov reward model transient analysis: An overview of numerical approaches." European Journal of Operational Research 40.2 (1989): 257-267.

[25] https://uptime.is/