Particle Filtering Based Availability Prediction for Web Services

Lina Yao and Quan Z. Sheng

School of Computer Science
The University of Adelaide, Australia
{lina,qsheng}@cs.adelaide.edu.au

Abstract. Guaranteeing the availability of Web services is a significant challenge due to unpredictable number of invocation requests the Web services have to handle at a time, as well as the dynamic nature of the Web. The issue becomes even more challenging for composite Web services in the sense that their availability is inevitably affected by corresponding component Web services. Current Quality of Service (QoS)-based selection solutions assume that the QoS of Web services (such as availability) is readily accessible and services with better availability are selected in the composition. Unfortunately, how to real-time maintain the availability information of Web services is largely overlooked. In addition, the performance of these approaches will become questionable when the pool of Web services is large. In this paper, we tackle these problems by exploiting particle filtering-based techniques. In particular, we have developed algorithms to precisely predict the availability of Web services and dynamically maintain a subset of Web services with higher availability. Web services can be always selected from this smaller space, thereby ensuring good performance in service compositions. Our implementation and experimental study demonstrate the feasibility and benefits of the proposed approach.

1 Introduction

Web services and service-oriented computing (SOC) represent a new paradigm for building distributed computing applications over the Internet. Unfortunately, after the development of nearly one decade, Web services are still in their infancy [10,17,13]. According to a recent study in Europe [2], the Web currently contains 30 billion Web pages, with 10 million new pages added each day. In contrast, only 12,000 real Web services exist on the Web. Even worse, many Web services have been deployed with dependability problems (e.g., unexpected behaviors, reliability, availability etc).

Guaranteeing the availability of a Web service is a significant challenge due to the unpredictable number of invocation requests the Web service has to handle at a time, as well as the dynamic nature of the Web. Over the last few years, many works have emerged in solving Web service availability problem. Almost all of these approaches are based on the concept of *service community* where Web services with similar functionalities (but different non-functional properties such as quality of service (QoS)) [1,18] are grouped in a particular community. The basic idea on improving the availability of Web service in a composition is to substitute Web services with poor quality using other

G. Kappel, Z. Maamar, H.R. Motahari-Nezhad (Eds.): ICSOC 2011, LNCS 7084, pp. 566–573, 2011.

services with better quality from the same service community. This typically involves QoS based service selection.

Most QoS service selection approaches assume that the QoS information (e.g., availability of Web service) is pre-existing and readily accessible. This unfortunately is not true. In reality, the availability status, as well as other QoS properties, of a Web service is highly uncertain, which changes over the time. How to accurately estimate and predict the availability status of a Web service becomes an important research problem. In addition, given the wide adoption of Web service in industry, more and more Web services will be available and the size of service communities will be inevitable large. Selecting from such a big space will lead to performance problem. Ideally, low quality Web services should be automatically filtered during service composition.

In this paper, we focus on solving above problems. In particular, we propose a particle filter based approach to precisely predicate and adjust Web service availability in real time. By continuously monitoring the service status, our approach offers more efficient and effective solution in service composition while ensure the high availability of composite Web services. Our work can be summarized as the following three original contributions:

- A model for availability of Web services using particle filter technique, which can perform precise prediction of the availability of Web services. Service availability is considered by combining both historical information and the predicted availability.
- An algorithm to optimize Web services selection by dynamically reducing the candidate Web services search space during Web services composition, and
- An implementation and experimental studies to validate the proposed approach.

The rest of the paper is organized as follows. Section 2 briefly introduces service availability model and the particle filter techniques. Section 3 describes the details of our approach and the algorithms. Section 4 reports the implementation and some preliminary experimental results. Finally, Section 5 overviews the related work and Section 6 provides some concluding remarks.

2 The Service Availability Model and the Particle Filter

In this section, we briefly introduce the service availability model and the particle filter technique, which serves as the core component of our approach on high availability of Web services composition.

2.1 Modeling Web Services Availability

There are different classifications of availability and many ways to calculate it [3]. Almost all existing approaches (e.g., [19,8,4]) use *operational* availability that measures the average availability over a period of time (i.e., the ratio of the service uptime to total time). Although this is simple to calculate, it is hard to measure the availability of a Web service at a specific time.

In this work, we model Web service availability as *instantaneous* (or point) availability. The instantaneous availability of a Web service s is the probability that s will

be operational (i.e., up and running) at a specific time t. The following discusses how to calculate the instantaneous availability of a Web service.

At given time t, a Web service s will be available if it satisfies one of the following conditions:

- The Web service s is working in the time frame of [0,t] (i.e., it never fails by time t). We represent the probability of this case as $\mathcal{R}(s,t)$.
- The Web service s works properly since the latest repair at time u ($0 < u < t$). The probability of this condition is $\int_0^t \mathcal{R}(s,t-u)m(s,u)du$, where $m(s,u)$ is the renewal density function of service s.

Based on these two conditions, the availability of service s at time t, $\mathcal{A}(s,t)$, can be calculated using the following formula:

$$\mathcal{A}(s,t) = \mathcal{R}(s,t) + \int_0^t R(s,t-u)m(s,u)du \tag{1}$$

2.2 The Particle Filter

We consider the availability of Web services as a dynamic system (i.e., it changes from time to time), which can be modeled as two equations: *state transition* equation and *measurement* equation. The states can not be observed directly and need to be estimated, while the measurements can be observed directly. Specifically, state transition is represented as:

$$x_k = f_k(x_{k-1}, u_{k-1}, v_{k-1}) \tag{2}$$

where f_k is a non-linear function, x_k, x_{k-1} are current and previous states, v_{k-1} is the state noise in non-Gaussian distribution, and u_{k-1} is the known input. Similarly, measurement is represented as

$$z_k = h_k(x_k, u_k, n_k) \tag{3}$$

where h_k is a non-linear function, z_k is a measurement, x_k is a state, and u_k is the known input.

The availability of Web services changes over time, which is full of uncertainty due to problems of network issues, hosting servers, and even service requester environments. We exploit the generic particle filter [7] to solve the dynamic availability of Web services, which will be discussed in the next section.

3 The Approach

Figure 1 shows the basic idea of our approach. Specifically, we propose to add a *filtering layer* between Web service layer and composition layer (right side of Figure 1). The layer of Web services contains several service communities and each of them consisting of Web services with similar functionalities. Each community may have large number of members.

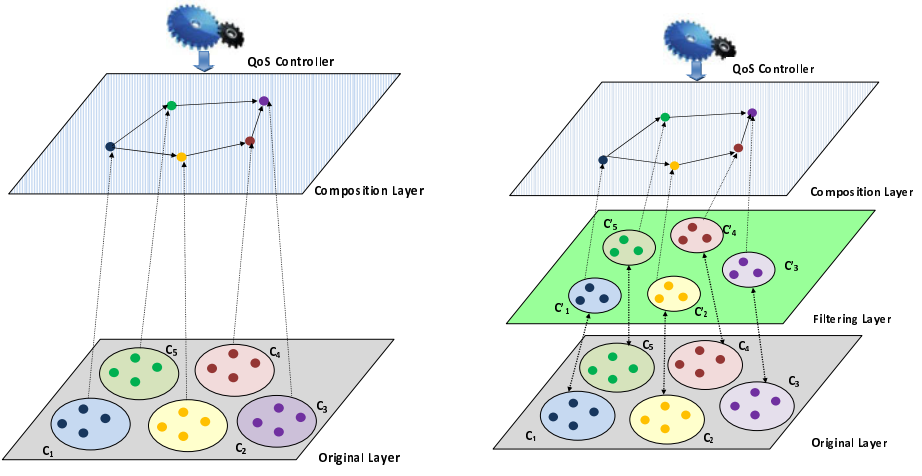


Fig. 1. (a) Existing approaches and (b) Our proposed approach

The filtering layer is essentially a subset of service communities, which consists of Web services with high availability that will directly involve in service compositions. The Web services are selected based on the accurate estimation and ranking algorithm described in this section. It should be noted that the relationship between Web service communities and the filtering layer is *dynamic* and *adaptive*. Our approach dynamically adjusts the members in the filtered service communities where degrading Web services will be replaced automatically with Web service with better availability from service communities. Web services' availability state is highly dynamic and therefore needs an adaptive approach to monitor and track each Web service's state. This is important to conduct optimized selection algorithm for composite Web services.

In our approach, we model the availability of a Web service i at time t as $x_i(t)$, which maintains the probability distribution for service availability estimation at time t, and inducted as the belief $Bel(x_i(t)) = \{x_i(t), w_i(t)\}$, $i = 1, 2, ..., \mathcal{M}$, where $w_i(t)$ are the different weight values, which indicate the contribution of the particle to the overall estimation, also called *important factors* ($\sum w_i(t) = 1$). Algorithm 1 shows the brief process on how it works.

Based on Algorithm 1, we can sort the top k Web services with high availability according to the monitoring and prediction. We call this estimated availability $\mathcal{E}_i$. In addition, for the overall filtering algorithm, we also take the history information on availability $\mathcal{H}_i$ into account, on top of the estimated availability by using the particle filter technique. The historical fluctuation of Web services availability has important impact on the current availability of the services. We call this historical fluctuation $\mathcal{H}$ impact as *availability reputation*. The most common and effective numerical measure of the center tendency is using the $mean$, however, it is sensitive to the extreme values

Algorithm 1. Particle Filter based Algorithm

1. **Initialization:** compute the weight distribution $\mathcal{D}_w(a)$ according to IP address distribution.
2. **Generation:** generate the particle set and assign the particle set weight, which means $\mathcal{N}$ discrete hypothesis

- generate initial particle set $\mathcal{P}_0$ which has $\mathcal{N}$ particles, $\mathcal{P}_0$ = $(p_{0,0}, p_{0,1}, ...p_{0,\mathcal{N}-1})$ and distribute them in a uniform distribution in the initial stage. Particle $p_{0,k} = (a_{0,k}, weight_{0,k})$ where a represents the Web service availability.
- assign weight to the particles according to our weight distribution $\mathcal{D}_w(a)$.

3. **Resampling:**

- Resample $\mathcal{N}$ particles from the particle set from a particle set $\mathcal{P}_t$ using weights of each particles.
- generate new particle set $\mathcal{P}_{t+1}$ and assign weight according to $\mathcal{D}_w(a)$

4. **Estimation:** predict new availability of the particle set $\mathcal{P}_t$ based on availability function $f(t)$.
5. **Update:**

- recalculate the weight of $\mathcal{P}_t$ based on measurement m_a, $w_{t,k}$= $\prod(\mathcal{D}_w(a))(\frac{1}{\sqrt{2\pi\phi}})exp(-\frac{dx_k^2 + dy_k^2}{2\phi^2})$, where $\delta a_k = m_a - a_{t,k}$
- calculate current availability by mean value of $p_t(a_t)$

6. Go to step 3 until convergence

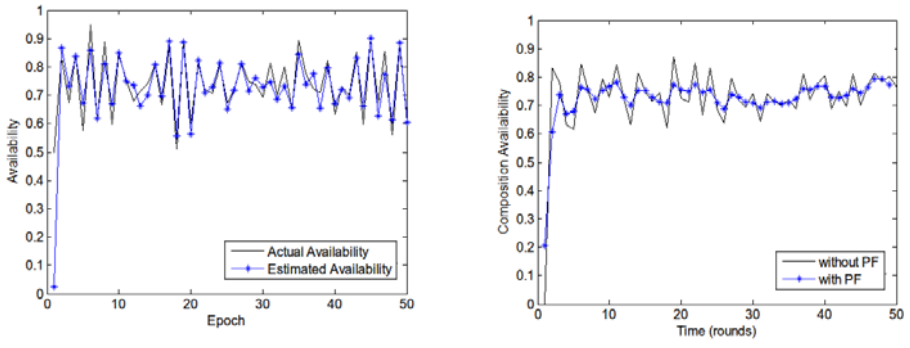
Algorithm 2. Overall Adaptive Filtering Algorithm

Input: initial availability values, α, τ.
Output: predicted availability, referencing availability, candidate list.
1. Read in the initial parameters;
2. Calculate each values for Web service $a_{ij}(s,t)$ in Web service community j at time t;
3. Predict the availability state of next time slot using particle filter (see Algorithm 1);
4. Looking up database and calculate the *mean* values of availability $\mathcal{H}$.
5. Calculating the reference availability $\mathcal{R}$.
6. Update the top k candidate list in each Web services community for every time interval τ;
7. Go to step 2.

(e.g., outliers) [5]. In our work, we define the final availability of a Web service as *reference availability* $\mathcal{R}$, which is calculated using:

$$\mathcal{R}_i(\tau) = \alpha\mathcal{E}_i(\tau) + (1-\alpha)\mathcal{H}_i(\sum_{1}^{\tau-1}(\tau-1)) \tag{4}$$

where $\alpha \in [0,1]$ is the weight and users can assign different weight based on their different preference, τ is a time span which can be defined by users. Finally, we summarize the overall particle filter algorithm in Algorithm 2.



(a) (b)

Fig. 2. (a) Actual availability vs estimated availability and (b) Availability of a composite Web service

4 Experimental Results

The proposed approach has been implemented in a prototype system in Java. In this section, we present two experimental results. For the experiments, we simulated 500 Web services of five different Web service communities (i.e., 100 Web services for each service community). We set the failure probability for the Web services as 3.5 percent, which complies with the findings in [6].

The first experiment studies the estimation accuracy of our approach, we simulated Web services' availability fluctuation and tracked their fluctuation of availability for 50 time steps (each time step counted as an *epoch*). The actual availability of Web services and corresponding estimated availability using our particle filter approach were collected and compared. Figure 2 (a) shows the result of one particular Web service. From the figure, we can see that our approach works well in tracing and predicting the availability of Web services.

The second experiment studies the impact our approach brought to the availability of composite Web services. We randomly generated composite Web services by composing services from five different communities. We simulated a comparatively significant fluctuation on the availability (i.e., changes in availability) of Web services for 50 different rounds and collected the availability information of the composite services under the situations of i) using our approach and ii) without using our approach. The availability of a composite Web service, $\mathcal{A}_c$, is represented as the mean value of its component Web services, i.e., $\mathcal{A}_c(c,t) = \alpha(\sum_{i=1}^{n} \mathcal{A}(s_i,t))/n$. Figure 2 (b) shows the availability of a particular composite Web service. From the figure we can see that the availability of the composite Web service is more stable when using our approach. In contrast, without using our approach, its availability is very sensitive to the fluctuations of service availability.

5 Related Work

There is a large body of research work related to the topic we have discussed in this paper. One important area on achieving high availability of Web services focuses on replication technology [11,12,14]. Serrano et al. [12] discuss an autonomic replication approach focusing on performance and consistency of Web services. Salas et al. [11] propose a replication framework for highly available Web services. Sheng et al. [14] further developed the idea by proposing an on-demand replication decision model that offers the solution to decide how many replicas should be created, when and where they should be deployed in the dynamic Internet environment. While these approaches focus on improving service availability through replication, our work concentrates on monitoring and predicting service availability. Our work is complementary to these works in the sense that the estimations provide a good source of information for replication decisions.

Many works achieve high availability of Web services based on the concept of *service communities* where Web services are selected based on QoS [8,19,16,9]. The basic idea is that services with similar functionalities are gathered as communities. If a Web service is unavailable, another service will be selected. However, most approaches assume that QoS is readily accessible and ignore its dynamic nature.

The works presented in [4,15] are the most similar ones to our work. In [4], Guo et al. model a composition process into the Markov Decision Process and use Kalman Filter to tracking the state of composite Web services. Sirin et al. [15] propose a filtering methodology that exploit matchmaking algorithms to help users filter and select services based on semantic Web services in composition process. However, these works focus on adaptive maintaining the composition of Web services and do not pay attention on the availability of component Web services. Our approach uses particle filter to precisely predict the availability of Web services and dynamically maintains a subset of Web services with higher availability, from which service developers can choose in their compositions.

6 Conclusion

Despite active development and research over the last decade, Web service technology is still not mature yet. In particular, guaranteeing the availability of Web services is a significant challenge due to unpredictable number of invocation requests the Web services have to handle at a time, as well as the dynamic nature of the Web. Many existing approaches ignore the uncertain nature of service availability and simply assume that the availability information of a Web service is readily accessed. In this paper, we have proposed a novel approach to monitor and predict Web service's availability based on particle filter techniques. Furthermore, we have developed algorithms to filter Web services for efficient service selection. The implementation and experimental results validated our approach.

Our ongoing work includes validating our approach on real Web services, conducting more experiments to study the performance of our approach (e.g., scalability). We also consider to extend our approach to other important service dependability properties (e.g., reputation, reliability, security).

References

1. Benatallah, B., Sheng, Q.Z., Dumas, M.: The Self-Serv Environment for Web Services Composition. IEEE Internet Computing 7(1) (January/February 2003)
2. Domingue, J., Fensel, D.: Toward A Service Web: Integrating the Semantic Web and Service Orientation. Service Web 3.0 Project, `http://www.serviceweb30.eu`
3. Elsayed, A.: Reliability Engineering. Addison-Wesley (1996)
4. Guo, H., Huai, J.-p., Li, Y., Deng, T.: KAF: Kalman Filter Based Adaptive Maintenance for Dependability of Composite Services. In: Bellahsène, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 328–342. Springer, Heidelberg (2008)
5. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann (2006)
6. Kim, S., Rosu, M.: A Survey of Public Web Services. In: Proceedings of the 13th International World Wide Web Conference (WWW 2004), New York, NY, USA (May 2004)
7. Kitagawa, G.: Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. Journal of Computational and Graphical Statistics 5(1), 1–25 (1996)
8. Liu, Y., Ngu, A., Zeng, L.: QoS Computation and Policing in Dynamic Web Service Selection. In: Proceedings of the 13th International World Wide Web Conference (WWW 2004), New York, NY, USA (May 2004)
9. Maamar, Z., Sheng, Q.Z., Benslimane, D.: Sustaining Web Services High Availability Using Communities. In: Proceedings of the 3rd International Conference on Availability, Reliability, and Security (ARES 2008), Barcelona, Spain (March 2008)
10. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-Oriented Computing: State of the Art and Research Challenges. IEEE Computer 40(11), 38–45 (2007)
11. Salas, J., Pérez-Sorrosal, F., Patiño-Martínez, M., Jiménez-Peris, R.: WS-Replication: A Framework for Highly Available Web Services. In: Proceedings of the 15th International Conference on World Wide Web (WWW 2006), Edinburgh, Scotland (May 2006)
12. Serrano, D., Patiño-Martìnez, M., Jimenez-Peris, R., Kemme, B.: An Autonomic Approach for Replication of Internet-based Services. In: Proceedings of the 27th IEEE International Symposium on Reliable Distributed Systems (SRDS 2008), Napoli, Italy (October 2008)
13. Sheng, Q.Z., Maamar, Z., Yahyaoui, H., Bentahar, J., Boukadi, K.: Separating Operational and Control Behaviors: A New Approach to Web Services Modeling. IEEE Internet Computing 14(3), 68–76 (2010)
14. Sheng, Q.Z., Maamar, Z., Yu, J., Ngu, A.H.: Robust Web Services Provisioning Through On-Demand Replication. In: Proceedings of the 8th International Conference on Information Systems Technology and Its Applications (ISTA 2009), Sydney, Australia (April 2009)
15. Sirin, E., Parsia, B., Hendler, J.: Filtering and Selecting Semantic Web Services with Interactive Composition Techniques. IEEE Intelligent Systems 19(4), 42–49 (2004)
16. Wang, X., Vitvar, T., Kerrigan, M., Toma, I.: A QoS-Aware Selection Model for Semantic Web Services. In: Dan, A., Lamersdorf, W. (eds.) ICSOC 2006. LNCS, vol. 4294, pp. 390–401. Springer, Heidelberg (2006)
17. Yu, Q., Bouguettaya, A., Medjahed, B.: Deploying and Managing Web Services: Issues, Solutions, and Directions. The VLDB Journal 17(3), 537–572 (2008)
18. Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., Sheng, Q.Z.: Quality Driven Web Services Composition. In: Proceedings of The 12th International World Wide Web Conference (WWW 2003), Budapest, Hungary (2003)
19. Zeng, L., Benatallah, B., Ngu, A., Dumas, M., Kalagnanam, J., Chang, H.: QoS-aware Middleware for Web Services Composition. IEEE Transactions on Software Engineering 30(5), 311–327 (2004)