

# Speeding Up Web Service Composition with Volatile External Information

John Harney  
LSDIS Lab, Dept. of Computer Science  
University of Georgia, GA 30602  
jfh@cs.uga.edu

Prashant Doshi  
LSDIS Lab, Dept. of Computer Science  
University of Georgia, GA 30602  
pdoshi@cs.uga.edu

## ABSTRACT

This paper introduces a novel method for composing Web services in the presence of external volatile information. Our approach, which we call the *informed-presumptive*, is compared to previous state-of-the-art approaches for Web service composition in volatile environments. We show empirically that the informed-presumptive strategy produces compositions in significantly less time than the other strategies with lesser backtracks.

## Categories and Subject Descriptors

H.3.5 [Online Information Systems]: Web-based Services

## General Terms

Algorithms, Theory

## Keywords

Web services, Adaptation, Volatile environments, Expiration times

## 1. INTRODUCTION

Web service compositions (WSCs) are an important area of study in services oriented computing. They offer the potential for automatically formulating processes in an efficient and timely manner. WSCs are predominantly modeled as being *static*, where the participating services are assumed to exhibit fixed quality of service (QoS) parameters during the composition and execution. However, in practice, WSC environments are often *volatile* – QoS of participating services may change during the composition. Therefore, WSCs must compose in the presence of this volatility.

Previous approaches [?, ?] associate expiration times with the service parameters. When the parameters of the component services expire, the WSC issues a query to gather the updated information. If the new parameter values are different from the previous ones, the WSC backtracks to the step in the composition where the service with the revised parameters is used. A significant limitation of these approaches is that some of the backtracks are redundant – changes in the parameter values do not necessitate changes in the composition. As excessive backtracking leads to time-consuming compositions, we may improve on these approaches by reducing the amount of backtracking.

We introduce a new method for managing volatility during composition, which we call the *informed-presumptive* approach. The method predicts whether the newly queried parameter values will cause significant changes in the composition. If the composition is

*expected* to change due to the revised parameters, then the informed-presumptive will backtrack to the point in the composition where the queried service occurs. Otherwise, it ignores the changed parameter value, as its impact on the composition is negligible. Thus, we seek to find a range of revised values that are expected to induce a change in the composition. Using the technique of *gradient descent*, we find the ranges of values with minimal computational overhead. In using the informed-presumptive, we form compositions in less time compared to previous state-of-the-art strategies, by eliminating backtracks that are not required. We empirically demonstrate the speed up using a simulated volatile environment.

## 2. METHODOLOGY

For a composition problem containing states,  $S$ , and component services,  $A$ , we find the service(s) to be invoked at each state  $s \in S$ . Let  $\hat{A}(s) \in A$  be the choice of services available at each state. The composition finds the optimal service(s) that maximizes the value function:

$$\pi^*(s) = \underset{a \in \hat{A}(s)}{\operatorname{argmax}} V^a \quad (1)$$

In Eq. ??,  $\pi^*$  is the optimal policy which maps each state to the optimal service(s) to invoke at that state,  $\pi^* : S \rightarrow A$ ; In defining the value function,  $V^a$ , we borrow from [?], and use a weighted summation of the QoS parameters of the service,  $a$ , and sum over all services in the composition. We use the normalized cost,  $\bar{C}^a$ , and availability  $\bar{A}v^a$ .

Let  $\bar{p}^a$  denote the revised parameter value vector for the service,  $a$ :

$$\bar{p}^a \stackrel{\text{def}}{=} [\bar{C}'^a, \bar{A}v'^a] \quad (2)$$

where  $\bar{C}'^a$  and  $\bar{A}v'^a$  are the revised cost and availability obtained by querying service  $a$ . (Of course,  $\bar{p}^a$  can be extended to account for other QoS parameters as well.)

As a change in the value function indicates a change in the composition, we seek those revised parameter vectors for each service that bring a change in the value function. Let  $\pi$  be the original policy and  $\pi^*$  be the revised policy given the new parameter vector, then we find all  $\bar{p}^a$  for which:

$$V^{\pi^*}(\bar{p}^a) - V^{\pi}(\bar{p}^a) = 0 \quad (3)$$

$V^{\pi}(\bar{p}^a)$  is the expected value of the original composition,  $\pi$ , in the presence of the revised parameter vector,  $\bar{p}^a$ , of service  $a$ , while  $V^{\pi^*}(\bar{p}^a)$  is the value of the composition following the optimal policy  $\pi^*$ . The parameter vectors,  $\bar{p}^a$ , that satisfy Eq. ?? are those for which no backtracking is needed.

We may view the difference in Eq. ?? as the error in using the original composition given the revised parameter vector,  $\bar{p}^a$ , and

## Average Composition Times

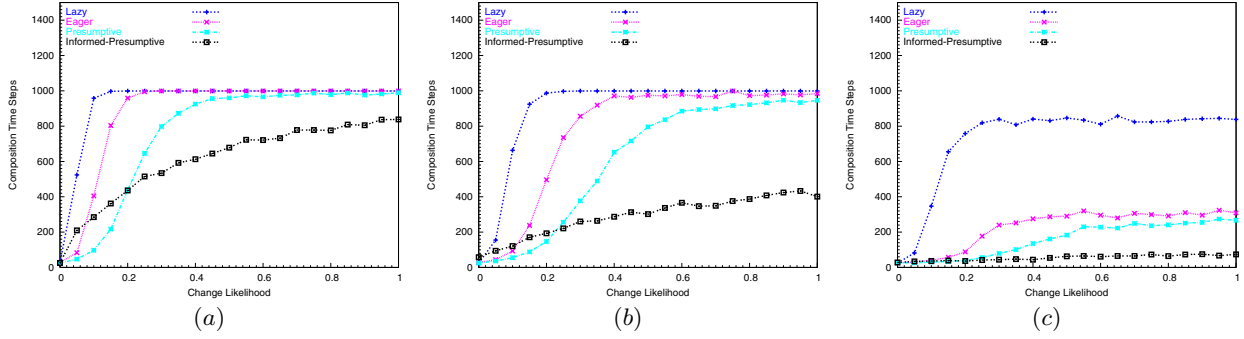


Figure 1: The average composition time for the four strategies with a volatility ratio of (a)1, (b).5, (c).33.

denote the difference as  $E(\bar{p}^a)$ . This allows us to model the problem of finding the vectors satisfying Eq. ?? as a gradient descent where we descend down the error surface,  $E(\bar{p}^a)$ , until we reach the minimal plateau ( $E(\bar{p}^a) = 0$ ). For performing the gradient descent, we update  $\bar{p}^a$  in the following way:

$$\bar{p}^a \leftarrow \bar{p}^a + \Delta \bar{p}^a \quad (4)$$

where

$$\Delta \bar{p}^a = -\eta \nabla E(\bar{p}^a) \quad (5)$$

Here,  $\eta$  is the step size,  $0 \leq \eta \leq 1$ . The negative sign indicates that we take a step in the direction of the reducing gradient. For our application, the definition for  $\nabla E(\bar{p}^a)$  is the vector:

$$\nabla E(\bar{p}^a) = \left[ \frac{\partial E(\bar{p}^a)}{\partial \bar{C}^a}, \frac{\partial E(\bar{p}^a)}{\partial \bar{A}^a} \right] \quad (6)$$

Fortunately, Theorem ?? simplifies the task of finding  $\nabla E(\bar{p}^a)$ .

**THEOREM 1.** *As  $V^a$  is a linear function of the QoS parameters of a service,  $a$ , the gradient  $\nabla E(\bar{p}^a)$  is constant.*

Theorem ?? allows us to perform gradient descent with minimal computational overhead, as we show later.

### 3. EXPERIMENT

Our experimental evaluation focuses on comparing composition times when using the informed-presumptive approach and the previously existing strategies. These are:

- The *eager* strategy, which on service expiration, issues a query to the service for the revised information. It suspends the WSC until the revised information is received and backtracks if there is a difference in the QoS value.
- The *lazy* strategy, which on service expiration, issues a query and continues the WSC until *completion*, at which point the revised information is processed. It backtracks if there is a difference in the QoS value.
- The *presumptive* strategy, which on service expiration, issues a query and continues the WSC until the *revised information is received*. It backtracks if there is a difference in QoS value.

Among these strategies, the presumptive approach was shown to perform the best.

For our experiments, we utilized a travel planning scenario in a simulated volatile environment, and implemented the above strategies in addition to the informed-presumptive.

Each component of a travel plan (ie. booking a plane ticket, reserving a hotel room, and renting a car) is a single step in the composition. At each step, we gave a choice of 10 vendor Web services,

with varying cost and availability values. Let the time to compose a step in the composition (without any backtracks) be  $t_{comp}^s$ , the expiration time for a service  $a$  be  $t_{exp}^a$ , and the query lag time for a service be  $t_{lag}^a$ . We use two measurements of volatility, (a) the volatility ratio,  $\frac{t_{comp}^s}{t_{exp}^a}$ , which signifies how often a service expires during the composition of a single step, and (b) the likelihood that the QoS parameter of a service will change upon expiration.

### 4. RESULTS

Figure ?? shows the results of our experiments for volatility ratios 1, 0.5, and 0.33. In Figure ??(a), we see that when services often expire during composition, the informed-presumptive approach shows lower composition times on average than the other strategies. However, at a low likelihood of change ( $\leq 20$  percent), the informed-presumptive performs slightly worse as precision errors due to overstepping in the gradient descent become visible. In Fig. ??(b), the informed-presumptive has a significantly better average composition time than the others. For lower volatility levels, such as 0.33 (Figure ??c), the informed-presumptive continues to outperform the strategies, although the services expire less often. The improved composition times of the informed-presumptive are achieved by eliminating unnecessary backtracks employed by the other strategies.

### 5. CONCLUSIONS AND FUTURE WORK

We have introduced a new approach for managing WSC in volatile environments that outperforms previous state-of-the-art composition strategies. We show that a significant time speed-up may be gained by predicting whether changed parameters induce changes in the composition, thus eliminating unnecessary backtracks.

When complex value functions and dependencies are introduced to the composition, performing gradient descent is likely to become a time-consuming task. Thus, our future work will explore implementing our approach on more complex compositions.

### 6. REFERENCES

- [1] V. Agarwal, G. Chafle, K. Dasgupta, N. Karnik, A. Kumar, S. Mittal, and B. Srivastava. Synth: A system for end to end composition of web services. *JWSR*, 3(4), 2005.
- [2] T.-C. Au, U. Kuter, and D. S. Nau. Web service composition with volatile information. In *ISWC*, 2005.
- [3] T.-C. Au and D. Nau. Reactive query policies: A formalism for planning with volatile external information. In *CIDM*, 2007.