# Title

Collaborative Reliability Prediction of Service-Oriented Systems

# Citation

Zibin Zheng, Michael R. Lyu

# Abstract

Author proposes a collaborative reliability prediction approach, which employs the past failure data of other similar users to predict the Web service reliability for the current user, without requiring real-world Web service invocations. Author also presents a user-collaborative failure data sharing mechanism and a reliability composition model for the service-oriented systems.

# Issues

Without sufficient past failure data, without internal information of the service components, and influenced by the unpredictable communication links, how to make effective and accurate reliability prediction on the service-oriented systems?

# Approach

Author proposes a collaborative framework and a collaborative reliability prediction approach, which employs past failure data of similar service users.

## Collaborative Framework

Instead of contributing videos (YouTube) or knowledge (Wikipedia), the service users are encouraged to contribute their individually observed Web service past failure data. To speedup the process, client-side middleware can be designed for the service users to automatically record the failure information of Web service invocations and contribute this information to a centralized server.

## Reliability Prediction

First, Author defines the performance of a Web service as its average failure probability (pi), as below:

, (1)

where is failure probability of Web service candidate observed by the service user , is the number of service users, and is the average failure probability of the Web service candidate .

After that, author calculates the similarity of the service users with the current user based on the past failure data of their commonly-invoked Web services. Then, a set of similar users are identified. After that, the invocation failure probabilities of Web services are predicted for the current user by using the invocation failure probabilities observed by similar users. Finally, the reliability of the service-oriented system is predicted by aggregating the failure probabilities of the service components.

### Phase 1: Similarity Computation

By a relationship matrix among service users and Web services, Person Correlation Coefficient (PCC), computes the similarity between service user and service user , as below:

, (2)

where is a set of commonly invoked Web services by both user and user , is the failure probability of Web service observed by the service user , and represents the average failure probability of all the Web services invoked by user .

Similar to the above approach, the similarity between Web service and Web service is defined:

, (3)

where is a set of service users who invoke both the Web services and , and is the average failure probability of Web service .

### Phase 2: Similar User Selection

In author’s approach, the service users who have negative correlation (negative PCC values) with the current user will be excluded. Thus, a set of similar service users can be identified by:

, (4)

where is the largest PCC value with the current user , is to exclude the dissimilar users, and can be calculated by Equation (2). A set of similar Web services with the current Web service can also be identified by:

, (5)

where is the largest PCC value with the current Web service and can be calculated by Equation (3).

### Phase 3: Failure Probability Prediction

Employing the similar users, the user-based approaches (named as UPCC) predict the missing value by the following equation:

, (6)

where and are average failure probabilities of different Web services observed by user and , respectively, and is the significant weight of the similar user , which is defined as:

, (7)

Item-based approaches (named as IPCC) predict the missing value by:

, (8)

where and are average failure probabilities of the Web service and observed by different service users, respectively, and is the significant weight of the similar Web service , which defined as:

, (9)

Considering that a missing value may not have similar users or similar items, author computes the as below:

, (11)

where is the Equation (6), is the Equation (8), and is a user-defined parameter for determining how much the missing value prediction relies on the similar users or the similar Web services.

### Phase 4: Reliability Aggregation

Failure probabilities of the basic compositional structures is introduced in the following:

* Sequence.

, (12)

where is the number of sequential tasks and is the failure probability of the task.

* Branch.

, (13)

where is the number of branches, is the failure probability of the branch, and is the execution probability of the branch ().

* Loop.

,(14)

where is the failure probability of the task in the loop structure, li is the probability of executing the loop for times, is the maximum looping times, and ). When , the composed invocation failure probability is 0 since the task is not executed.

* Parallel.

, (15)

where is the probability of the parallel branch will task.

With assumption of independent task failure in the service flow, a service flow could be decomposed to the basic compositional structures. Author predicts the reliability of the service flow using exponential reliability function:

, (17)

where p is the composed failure probability of the service flow, f is the execution frequency of the service flow, and t is the time period for which the reliability is to be calculated.

# Conclusion

In this paper, Author proposes a collaborative reliability prediction approach for service-oriented systems. The main idea is to exploit the past failure data of the similar users for predicting Web service failure probabilities for the current user. The predicted failure probabilities of the elementary Web services are composed using different compositional structures to predict the reliability of the whole service-oriented system.

By experiment, result indicates the effectiveness of this approach.

# My Idea

1. When there are no similar users or similar services, the prediction may be inaccurate. Because there is no evidence that a number of service users, having a similar behavior for some services, will also have a similar behavior for a particular service. But I have no idea to solve this problem.
2. Maybe we could make some parameter self-adaptive, like in neighborhood-based approach. Just according to the experiment result, maybe there is some relation between and density of matrix.
3. We could consider the updates of services into the approach. For example, we could weight the time, which leads to that newer the data is, greater the weight is.