

Cost-Based Prevention of Violations of Service Level Agreements in Composed Services Using Self-Adaptation

Philipp Leitner, Schahram Dustdar
Distributed Systems Group
Vienna University of Technology
Argentinierstrasse 8, 1040, Wien, Austria
{lastname}@infosys.tuwien.ac.at

Branimir Wetzstein, Frank Leymann
Institute of Architecture of Application Systems
University of Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany
{lastname}@iaas.uni-stuttgart.de

Abstract—Providers of composite Web services face the challenge of having to comply to SLAs, which are agreements governing the minimum performance that customers can expect from a composite service. In this work, a framework for optimizing adaptations of service compositions with regards to SLA violations has been developed. The framework, dubbed PREvent (Prediction and Prevention of SLA Violations Based on Events), uses techniques from the areas of machine learning and heuristic optimization to construct models for prediction of SLA violations at runtime, and to decide which adaptation actions may be used to improve overall performance in a composition instance. An optimizer component decides, whether applying these changes makes sense economically (i.e., whether the costs of violating the SLAs are bigger than the adaptation costs). If this is the case, the respective actions are applied in an automated way. At its core, PREvent is a self-optimizing system in the sense of autonomic computing, with the target of minimizing the total costs of adaptations and SLA violations for the service provider.

Keywords—service-based computing; autonomic computing; heuristic optimization

I. INTRODUCTION

Service-based applications have seen large industrial and research interest in the last years. Increasingly, non-functional aspects and quality-of-service (QoS) are becoming essential features of service management. In the business world, QoS is typically defined within legally binding Service Level Agreements (SLAs), represented. SLAs contain Service Level Objectives (SLOs), i.e., concrete numerical QoS objectives, which the service needs to fulfill. If SLOs are violated, agreed upon monetary consequences go into effect. For this reason, providers have an interest in monitoring SLAs and preventing violations, either by using post mortem analysis and optimization [1], or by runtime prediction of performance problems [2]. We argue that the latter is more powerful, allowing to prevent violations before they have happened by timely application of runtime adaptation actions [3], [4]. However, preventing SLA violations is not for free. Therefore, a tradeoff between preventing violations and the inherent costs of doing so exists. We formalizing this tradeoff as an optimization problem, with the goal of minimizing the total costs (of violations and

applied adaptations) for the service provider. Furthermore, we introduce a self-optimizing system in the autonomic computing sense, which uses prediction mechanisms based on machine learning techniques and automated adaptation to optimize the aforementioned total costs.

II. COST-BASED COMPOSITION OPTIMIZATION

We dubbed this self-optimizing system PREVENT (Prediction and Prevention of SLA Violations Based on Events). We sketch the overall architecture of PREVENT in Figure 1.

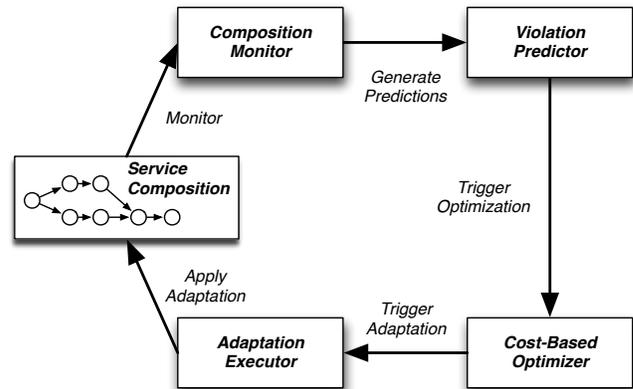


Figure 1. Overall Framework

PREVENT follows the seminal steps of **M**onitoring, **A**nalysis, **P**lanning, and **E**xecution [5] of the MAPE loop. Monitoring is implemented using the event-driven monitoring pattern [6]. The monitoring runtime data (e.g., QoS information) is then fed into machine learning models (e.g., artificial neural networks, decision trees) to generate predictions of SLA violations ahead of time [7]. These predictions are the input to the Cost-Based Optimizer component, which decides, based on generated predictions, existing SLAs, and available runtime adaptations, which adaptations are cost-optimal for the service provider [8]. These adaptations may prevent all or a subset of the predicted violations, but do not necessarily do so (if it is cheaper for the provider to just pay

the penalty than to adapt). Finally, the selected adaptations are applied to the running service composition, for instance using the notion of fragment-based adaptation [3].

We consider the Cost-Based Optimizer component as the core of PREVENT. For each composition instance with predicted violations, this component needs to find the subset of adaptations that minimizes the predicted total costs (TC) for the service provider. We have formalized this optimization problem as in Equation 1, with A^* being the selected subset of adaptation actions.

$$TC(A^*) \approx v(A^*) + \sum_{s_x \in S} e_{s_x}^i + \sum_{a_x \in A^*} c(a_x) \rightarrow \min! \quad (1)$$

Essentially, Equation 1 consists of three terms: (1) $v(A^*)$ is a penalty term, which defines the costs of applying conflicting adaptations as ∞ , (2) the term $\sum_{s_x \in S} e_{s_x}^i$ represents the (estimated) penalty payments for SLA violations that the provider will have to pay after adaptation, and (3) the term $\sum_{a_x \in A^*} c(a_x)$ captures the costs of adaptation. Please refer to the original publication [3] for details on this optimization problem.

The Cost-Based Optimizer has all necessary information to evaluate Equation 1 at runtime for any set of actions A^* . However, finding the A^* that minimizes $TC(A^*)$ is still far from trivial, as this equation is discrete and cannot be optimized analytically. Hence, we analyzed a number of different algorithms for efficiently finding the best adaptations to apply. More concretely, we researched the Branch-and-Bound algorithm for deterministically solving smaller problem instances (i.e., instances with a smaller number of available adaptations). For larger problem instances, we turned to heuristic optimization, e.g., variations of local search and genetic algorithms. As reported in [8], we achieved the best results using Greedy Randomized Adaptive Search Procedure [9] (GRASP) and memetic algorithms [10] (a variation of genetic algorithms with locally optimized population members).

III. CONCLUSIONS

In this extended abstract, we have outlined the PREVENT framework for autonomically preventing SLA violations in service compositions. The unique feature of PREVENT is that it keeps the total costs for the service provider minimal, in that it attempts to prevent violations only if the benefit will outweigh the costs for the provider. In the implementation of the PREVENT model, we used a combination of event-driven monitoring, machine learning based prediction, and optimization. As part of our future research, we will go beyond the discussed S-Cube results and extend the PREVENT model with support for preventing violations of aggregated SLAs, i.e., SLAs which are defined over a period of time instead of each composition instance individually. Furthermore, we will investigate what new types of runtime

adaptation are being opened up by the current buzz of cloud computing.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement 215483 (S-Cube).

REFERENCES

- [1] L. Bodenstaff, A. Wombacher, M. Reichert, and M. C. Jaeger, "Analyzing Impact Factors on Composite Services," in *Proceedings of the 2009 IEEE International Conference on Services Computing (SCC '09)*. Los Alamitos, CA, USA: IEEE Computer Society, 2009, pp. 218–226.
- [2] L. Zeng, C. Lingenfelder, H. Lei, and H. Chang, "Event-Driven Quality of Service Prediction," in *Proceedings of the 6th International Conference on Service-Oriented Computing (ICSOC'08)*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 147–161.
- [3] P. Leitner, B. Wetzstein, D. Karastoyanova, W. Hummer, S. Dustdar, and F. Leymann, "Preventing SLA Violations in Service Compositions Using Aspect-Based Fragment Substitution," in *Proceedings of the International Conference on Service-Oriented Computing (ICSOC'10)*, 2010.
- [4] R. Kazhamiakin, B. Wetzstein, D. Karastoyanova, M. Pistore, and F. Leymann, "Adaptation of Service-Based Applications Based on Process Quality Factor Analysis," in *Proceedings of the 2nd Workshop on Monitoring, Adaptation and Beyond (MONA+)*, 2009, pp. 395–404.
- [5] J. O. Kephart and D. M. Chess, "The Vision of Autonomic Computing," *IEEE Computer*, vol. 36, no. 1, pp. 41–50, 2003.
- [6] P. Leitner, A. Michlmayr, F. Rosenberg, and S. Dustdar, "Monitoring, Prediction and Prevention of SLA Violations in Composite Services," in *Proceedings of the IEEE International Conference on Web Services (ICWS'10)*. Los Alamitos, CA, USA: IEEE Computer Society, 2010, pp. 369–376.
- [7] P. Leitner, B. Wetzstein, F. Rosenberg, A. Michlmayr, S. Dustdar, and F. Leymann, "Runtime Prediction of Service Level Agreement Violations for Composite Services," in *Proceedings of the 3rd Workshop on Non-Functional Properties and SLA Management in Service-Oriented Computing (NFPSLAM-SOC'09)*, 2009, pp. 176–186.
- [8] P. Leitner, W. Hummer, and S. Dustdar, "Cost-based optimization of service compositions," *IEEE Transactions on Services Computing (TSC)*, 2012.
- [9] T. Feo and M. Resende, "Greedy Randomized Adaptive Search Procedures," *Journal of Global Optimization*, vol. 6, pp. 109–133, 1995.
- [10] N. Radcliffe and P. Surry, "Formal Memetic Algorithms," *Evolutionary Computing*, vol. 865, pp. 1–16, 1994.