Jan Jürjens, Kurt Schneider (Hrsg.): Software Engineering 2017,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2017 99

Considering Transient Effects of Self-Adaptations in Model-Driven Performance Analyses

Christian Stier[1] Anne Koziolek[2]

Abstract: Self-adaptive systems reconfigure themselves to meet requirements under changing user load. Model-driven performance analyses for self-adaptive systems enable software architects to evaluate whether a self-adaptive system meets requirements under varying user load. It is essential to the efficiency of a self-adaptive system that it adapts its configuration at the right time. The effectiveness of adaptations depends not only on the time when an adaptation decision is made but also on its execution time. The execution of adaptations can cause additional stress on the system. This can further deteriorate system performance. Existing model-driven analyses do not consider these transient effects. We present an approach that enables systematic modeling and analysis of transient effects in software performance analyses. We apply our approach to a horizontally scalable media hosting application. By considering the transient effects of *scale-outs* we were able to increase prediction accuracy for response times of the applications services. Further experiments demonstrated that our approach enables detection and resolution of design deficiencies of self-adaptive systems.

Keywords: Software Performance Engineering, Software Performance Prediction, Component-Based Software Engineering, Design Time, Software Architecture

1 Modeling the Interdependence of System and Reconfiguration Performance

In our work on *Considering Transient Effects of Self-Adaptations in Model-Driven Performance Analyses* [SK16] we outline an approach for the systematic specification and analysis of the performance impact of reconfigurations. Self-adaptive software systems adapt their structure, configuration and deployment to meet QoS requirements under changing user load. The efficiency and effectiveness with which a self-adaptive system reacts to changes in user load depends upon the current and expected context in which the decision is made. If the autoscaler of a horizontally scalable application scales the number of provisioned resources too late to react to a burst in user load, QoS goals might not be met due to resource contention. On the other hand, provisioning resources to become available before they are actually needed can lead to additional cost. Transient effects refer to the performance impact of reconfigurations on the QoS of a software system in a transient phase.

Design-time software performance analyses such as SimuLizar [BLB] enable software architects to identify QoS requirement violations for self-adaptive systems. This enables software architects and system operators to reason on how well the system is able to adapt.

[1] FZI Research Center for Information Technology, stier@fzi.de
[2] Karlsruhe Institute of Technology, koziolek@kit.edu

Existing analyses neglect the effect of transient effects on QoS. Thus, they provide inaccurate and unreliable QoS predictions.

Our approach enables a systematic analysis of transient effects by linking the specification of reconfiguration mechanisms and their transient effects. Our language describes reconfigurations as reusable *Self-Adaptation Actions*, a concept used by many self-adaptation frameworks [CG12; Hu16]. Every action is specified as a sequence of parametrized steps. We distinguish three main types of steps. *BranchingAdaptationStep* models conditions of a reconfiguration. *ResourceDemandingStep* models the performance impact caused by enacting the reconfiguration. To model the performance impact we leverage the Palladio Component Model (PCM). *EnactAdaptationStep* describes the effect of the action on the system configuration. The effects is described via a parametrized model transformation.

2 Analyzing Transient Effects of Self-Adaptive Software Systems

We propose an analysis methodology for considering transient effects in software performance analyses. We integrated our methodology with the simulation-based analysis of self-adaptive systems of SimuLizar [BLB]. Once a reconfiguration mechanism decides to enact an Adaptation Action, our *Action Model Interpreter* executes the action within the analysis. The Interpreter traverses and executes the steps of an action.

In our evaluation we investigate to which extent the consideration of transient effects improves the prediction accuracy and quality of model-based analyses of self-adaptive software systems. For this, we used a horizontally scaling web-based media hosting application. The evaluation compared the baseline SimuLizar with a version extended by our approach. By considering the time required to execute scale-outs, we were able to reduce the median response time prediction error over the full experiment scenario from 22.5% to 16.5%. In transient phases the effect on prediction error was significantly higher, reducing the error from 14.6% to 5.4%. The significantly higher accuracy in transient phases enabled us to identify and address design deficiencies in the system under investigation.

References

[BLB] Becker, M.; Luckey, M.; Becker, S.: Performance Analysis of Self-Adaptive Systems for Requirements Validation at Design-Time. In: QoSA 2013. ACM.

[CG12] Cheng, S.-W.; Garlan, D.: Stitch: A language for architecture-based self-adaptation. Journal of Systems and Software 85/12, pp. 2860–2875, 2012.

[Hu16] Huber, N.; Brosig, F.; Spinner, S.; Kounev, S.; Bahr, M.: Model-Based Self-Aware Performance and Resource Management Using the Descartes Modeling Language. IEEE Transactions on Software Engineering/99, pp. 1–1, 2016.

[SK16] Stier, C.; Koziolek, A.: Considering Transient Effects of Self-Adaptations in Model-Driven Performance Analyses. In: QoSA 2016. ACM, Venice, 2016.