An approach to predictive analysis of self-adaptive systems in design time

Patrícia Araújo de Oliveira
Advised by Francisco Durán and Ernesto Pimentel

University of Málaga, Spain,
{patricia, duran, ernesto}@lcc.uma.es

Abstract. Predictive analysis methods offer the possibility of estimating the impact of design decisions, which may help in the accomplishment of operational optimal results, before the deployment of the system is made, and therefore minimizing the required maintainance effort and cost. However, current predictive methods are not effective when used on self-adaptive systems, and specifically when used on cloud environments, because of its complexity and dynamic nature. The main goal of this thesis project is to investigate different methods for the specification of adaptive systems, and to propose techniques and tools for the modeling of self-adaptive systems and environments, considering adaptation mechanisms, and approaches for the estimation of different Quality of Service (QoS) metrics that help in the analysis of the systems to be developed. Specifically, we will provide generic mechanisms for the modeling of adaptive systems and environments, the definition of metrics as transformation rules, and tools using such system specifications for their analysis. We will focus on the kind of systems and adaptation mechanisms we find in the cloud, and will evaluate our proposal on state-of-the-art cloud applications. We will present an approach of predictive analysis, based on graph transformation, which provides the capability of taking decisions about elasticity-related QoS from the definition of an adaptive model of the system and the specification of adaptation mechanisms described in a formal language to control elasticity in cloud applications. The proposed approach will enable the simulation of cloud environments and their elasticity features at design time, which allows the prediction of different QoS metrics in cloud scenarios, and provides the capability of specifying and tuning elasticity monitoring, constraints, and strategies at different levels.

1 Research problem and motivation

An important feature of self-adaptive systems, such as cloud platforms, is the capability to react to changes dynamically, which brings new challenges from the perspective of predictive analysis. Elasticity is one of the key issues in Cloud Computing, where in fact, elasticity refers not only to resources but also to quality and cost, and how they can influence each other when changing the context. Therefore, having the capability of predicting problems related with performance constraints, scalability issues or reliability risks, becomes especially relevant when adaptive systems are being modelled. Predictive information related with Quality of Service (QoS) metrics allows the adoption of decisions before the deployment phase, so preventing problems and mitigating the impact of bad design or management decisions. In order to get this kind of useful information, we need to have precise and appropriate models both of the application to be analysed and of the context (cloud platform or self-adaptive system) where the application is going to be deployed. The more accurate the model is, the more precise the prediction. In addition to the possibility of getting predictive information, the same scenario may also help in the calibration of quality parameters, providing support to estimate optimal values for them.

A number of different predictive analysis approaches can be found in the literature, including techniques based on stochastic networks, Petri nets, statistical methods or simulation. However, current predictive analysis tools present limitations to deal with dynamic architectures. The main limitations exhibited by most current tools comes from the fact that they do not allow changes in the model along the analysis process, operating just on static structures. This is the case, for instance, of the Palladio tool [1], one of the currently more successful predictive analysis frameworks, and widely used both in industry and academia. Thus, for instance, the Palladio Simulator can be used to predict QoS properties (performance and reliability) from software architecture models.

Although Palladio group is introducing the cloud and adaptive systems issues in some recent work [2] [3] the limitation of using static architectures has not still been solved. There have been other attempts to deal with these issues, such as those in [4] to build support for dynamic systems on Palladio. Other tools such as D-Klaper [5], MEDEA citefalkner2016model or the one described in [6] have proposed alternative solutions for performance prediction of dynamic systems. However, much work needs still to be made to solve the problem in a convenient way as it is explained in Section 3.

2 Research challenges

With the aim of extending the Palladio's predictive capabilities to support dynamic systems, we propose to use the e-Motions implementation of the Palladio Component Model (PCM) described in [7]. Given the metamodel of Palladio and its operational semantics expressed in terms of graph-transformation rules, this implementation allows to analyse (static) systems modeled in Palladio Bench. However, the relevance of this e-Motions specification is the capability of integrating new adaptation rules, which extend the behaviour of Palladio, making the analysis of dynamic systems feasible, and increasing its expressiveness [8].

In order to explain the contribution intended of this thesis project, the diagram in Figure 1 summarizes the relationship between the main processes and elements of the proposal. In e-Motions, a DSL is specified by its syntax (a metamodel) and a behavior (an operational semantics described as a set of graph transformation rules). The systems to be analyzed are specified using Palladio, and specifically the Palladio Component Model (PCM). Models conforming to the PCM are composed of four different submodels, which correspond to different and complementary views of the system. The Palladio language is specified in e-Motions by taking an extended PCM, denoted PCM^*, which includes definitions for tokens and dynamics of systems, and its behavior. As presented in [7], static systems defined in the Palladio Bench (models $\mathsf{M_{app}}$ conforming to the PCM) can be loaded and analyzed in e-Motions using the DSL $\mathsf{PCM}^* + \mathsf{Beh_{Palladio}}$.

To deal with adaptive systems, the behavior to be used is extended with adaptation mechanisms, specified as additional e-Motions transformation rules, $\mathsf{Beh_{Adaptation}}$. To control the application of the adaptation operations, SYBL (Simple Yet Beautiful Language) [9] is used, which is a language for controlling elasticity in cloud applications in runtime. For a specific model $\mathsf{M_{app}}$, the SYBL annotations provided are represented as a model extension $\mathsf{Ctrl_{SYBL}}$ so that the extended model $\mathsf{M_{app}} + \mathsf{Ctrl_{SYBL}}$ can be used to analyze the performance of the described adaptive system using the DSL $\mathsf{PCM}^* + (\mathsf{Beh_{Palladio}} + \mathsf{Beh_{Adaptation}})$.

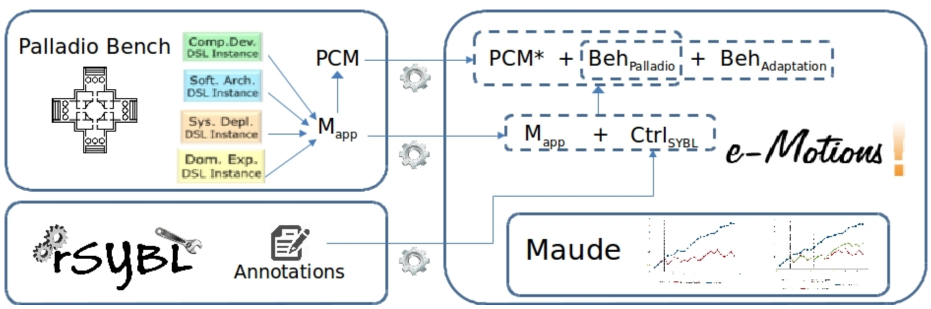

Fig. 1. Architecture

Our proposal will try to offer the techniques and tools to allow the modeling of self-adaptive systems, and specifically in cloud infrastructures, and their analysis, so that a better estimation of the satisfaction of the requirements of systems can be carried on, supporting a better selection of resources and a better calibration of the operational parameters.

Our claim is that by extending the capacity of the Palladio system, and specifically its PCM and tooling, will allow us to perform more precise statistical analysis of self-adaptive systems, including those reconfiguration algorithms usually found in cloud environments.

3 Related work

In this section, we discuss other approaches that use predictive strategies, most of them for performance prediction at runtime or at design time.

Huber et al. propose in [4] a DSL to describe the behaviour of self-adaptive systems based on strategies, tactics and actions. This work is part of the Descartes project, which uses Palladio PCM for their design time phases. However, they focus on runtime performance analysis, not in predictive analysis of applications at design time.

SimuLizar [3] is an extension of Palladio for the performance analysis of self-adaptive systems at design time. However, the simulation scope is limited to only a set of rules that are triggered between the static environment models, which prevents from testing all possible reachable states of the systems.

D-Klaper [5] is a tool for model-driven performance engineering which can be applied to self-adaptive systems. It uses an intermediate language to provide software design models, which can then be analyzed. However, the D-Klaper language does not support the modeling of adaptation rules, nor the transformation of input models.

MEDEA [10] is an approach that proposes the performance prediction at the beginning of its life cycle for this, modeled the workloads with the resource consumption, capturing the CPU, memory and disks. However, although using models to represent the system, this is used to generate executable code for real hardware and middleware deployments and the results of the executions are presented to the expert through specific context views that indicate whether the design meets the performance requirements, not acting entirely at design time.

Johnsen et al. present in [6] an approach model-based prediction to compare the effect, in terms of performance and accumulated cost, of selecting different instance types for virtual servers from Amazon Web Services (AWS), for this their used a highly configurable modeling framework for applications running on Apache YARN, the ABS-YARN, which using the executable semantics of Real-Time ABS, defined in Maude, as a simulation tool. However, they do not model the application but use values obtained from real measurements.

CloudScale method [2] aims to analyze and identify the scalability problem in design time. This approach presents the ScaleDL languages, which consist in the set of languages required to allow software architects managing scalability, elasticity, and cost-efficiency. This set of languages includes the technologies mentioned above Simulizar and Descartes. Although it is a robust method, the Palladio limitations - as no possibility of modifications of the initial models - remain present in the approach.

4 Proposed solution and preliminary results

Building on the already available e-Motions specification of Palladio, we have already specified several adaptation mechanisms by providing appropriate adaptation rules as graph transformation rules. Specifically, we can specify a system in Palladio, including descriptions of its components architecture, environment and usage model, and perform its QoS analysis in e-Motions using transformation rules that describe the adaptation mechanisms available in the system and its environment.

We were able to operate simulations and perform predictive analysis taking into account some the adaptation mechanisms, and thus showing the feasibility of the approach. So far, we have been able to model the dynamic adaptation of the system in accordance to its continuous monitoring by creating rules for: load balancing, variable usage, the scale in/out nodes and the scale up/down resources capacity (specifically CPU).

As early experiments, presented in [8] and [11], we were able to perform simulation-based predictive analysis of adaptive systems. Our approach is able to operate simulations and perform predictive analysis taking into account different adaptation mechanisms. For the case study used we observed the response time, throughput at resource usage of the system in order to show the feasibility of the approach.

The prediction of QoS metrics is one of the most relevant issues when gathering knowledge of applications and their environments, compared to other solutions presented in the literature [12]. However, existing prediction methods do not consider specific cloud metrics and, therefore, they are not capable of managing other particular cloud features, such as self-provisioning on demand, measured usage, network access, resource pooling, and elasticity. Properties related to scalability, elasticity and efficiency are essential to achieve a dynamic adaptation in a cloud scenario, specifically for resource allocation and pay-per-use. Thus, we need to take into account these new metrics [13], and also a taxonomy of different sources of uncertainty present in the models of self-adaptive systems and the different ways of managing them [14].

5 Plan for evaluation and validation

Building on the knowledge we have gathered so far, we will specify other mechanisms of adaptation available for cloud systems in more ambitious case studies. We will consider other QoS metrics, including not only performance metrics, but also others related to feasibility, costs, and security.

With this thesis, we intend to offer the techniques and tools to allow the modeling of self-adaptive systems, and specifically cloud infrastructure, and their analysis, so that a better estimation of the satisfaction of the requirements of systems can be carried on, supporting a better selection of resources and a better calibration of the operational parameters.

To perform the analysis of a dynamic system, it is necessary to consider their capacity to process and manage different workflows, react through variations of the usage, and to carry on the necessary changes when components have assigned different workloads. Following standard techniques, we will model workloads based on real uses of systems, and will use this information to perform our simulations.

We will evaluate our proposal modeling real applications running in real cloud environments, and will verify that the results produced by our predictive analysis match the actual behavior of the real system executed in the cloud. We will use standard techniques, such as the Kolmogorov-Smirnov test [15] and other similar ones, for the comparison of the probability distributions.

References

1. Happe, J., Koziolek, H., Reussner, R.: Facilitating performance predictions using software components. IEEE Software **28**(3) (2011) 27–33
2. Becker, S., Brataas, G., Lehrig, S.: Engineering Scalable, Elastic, and Cost-Efficient Cloud Computing Applications: The CloudScale Method. Springer International Publishing (6 2017)
3. Becker, M., Becker, S., Meyer, J.: Simulizar: Design-time modeling and performance analysis of self-adaptive systems. Software Engineering **213** (2013) 71–84
4. Huber, N., van Hoorn, A., Koziolek, A., Brosig, F., Kounev, S.: S/t/a: Metamodeling run-time adaptation in component-based system architectures. In: e-Business Engineering (ICEBE), 2012 IEEE Ninth International Conference on, IEEE (2012) 70–77
5. Grassi, V., Mirandola, R., Randazzo, E.: Model-driven assessment of QoS-aware self-adaptation. In: Software Engineering for Self-Adaptive Systems. Springer (2009) 201–222
6. Johnsen, E.B., Lin, J., Yu, I.C.: Comparing AWS deployments using model-based predictions. In Margaria, T., Steffen, B., eds.: Proceedings 7th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA). Volume 9953 of Lecture Notes in Computer Science., Springer (2016) 482–496
7. Moreno-Delgado, A., Durán, F., Zschaler, S., Troya, J.: Modular DSLs for flexible analysis: An e-Motions reimplementation of Palladio. In: European Conference on Modelling Foundations and Applications (ECMFA), Springer (2014) 132–147
8. de Oliveira, P.A., Moreno-Delgado, A., Durán, F., Pimentel, E.: Towards the predictive analysis of cloud systems with e-Motions. In: XX Ibero-American Conference on Software Engineering (CIbSE). (2017)
9. Copil, G., Moldovan, D., Truong, H.L., Dustdar, S.: Sybl: An extensible language for controlling elasticity in cloud applications. In: Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on, IEEE (2013) 112–119
10. Falkner, K., Szabo, C., Chiprianov, V.: Model-driven performance prediction of systems of systems. In: Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems, ACM (2016) 44–44
11. de Oliveira, P.A., Durán, F., Pimentel, E.: Graph-transformation for the performance analysis of elastic systems. In: 15th International Workshop on Foundations of Coordination Languages and Self-Adaptative Systems. (2017)
12. Chinneck, J., Litoiu, M., Woodside, M.: Real-time multi-cloud management needs application awareness. In: Proceedings of the 5th ACM/SPEC International Conference on Performance Engineering. ICPE '14, New York, NY, USA, ACM (2014) 293–296
13. Becker, M., Lehrig, S., Becker, S.: Systematically deriving quality metrics for cloud computing systems. In: Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering, ACM (2015) 169–174
14. Perez-Palacin, D., Mirandola, R.: Uncertainties in the modeling of self-adaptive systems: A taxonomy and an example of availability evaluation. In: Proceedings of the 5th ACM/SPEC International Conference on Performance Engineering. ICPE '14, New York, NY, USA, ACM (2014) 3–14
15. Hollander, A., Wolfe, D.: Nonparametric statistical methods. wiley-interscience. New York, NY (1999)