PALLADIO OPTIMIZATION SUITE: QOS OPTIMIZATION FOR COMPONENT-BASED CLOUD APPLICATIONS

Michele Ciavotta
Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria.
michele.ciavotta@polimi.it
Danilo Ardagna
Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria.
danilo.ardagna@polimi.it
Anne Koziolek
Karlsruhe Institute of Technology, Institute for Program Structures and Data Organization.
anne.koziolek@kit.edu

Abstract. One important issue in software engineering is to find an effective way to deal with the increasing complexity of software computing system. Modern software applications have evolved in terms of size and scope. Specific tools have been created to predict the Quality of Service (QoS) at design-time. However, the optimization of an architecture usually has to be done manually, resulting in an arduous and time-consuming process. For this reason, we present the Palladio Optimization Suite (POS), a collection of complementary plugins realized to run atop Palladio Bench with the aim of automatizing the exploration of the space of possible architectures by means of advanced search paradigms.

Model-Driven, Cloud, QoS, Optimization

1. Introduction

One of today's issues in software engineering is to find new effective ways to deal with the increasing complexity of software computing system. Modern software applications have evolved not only in terms of size and scope, but also in the criticality of the services supported. Another factor to consider in this change is the emergence and the success of Cloud computing, which has many interesting features to offer but, at the same time, it may introduce some non-negligible issues and new challenges in application development. Such a scenario calls for dependable software systems able to guarantee the achievement of Quality of Service (QoS) requirements, such as performance and availability at design-time. To reach this goal, the use of software architecture (SA) has emerged as an appropriate level for dealing with software qualities and several efforts have been devoted to the definition of methods and tools able to evaluate quality at SA level. However, each method can usually only assess quality attributes (e.g., performance or availability) for fully defined architectures. Palladio [2] is a software solution for SA description, and forecast of non-functional requirements. Even if the support to QoS analysis is valuable, the space of design alternatives for a single application is usually very large and the task of finding the most suitable architecture is often arduous and time demanding; for this reason solutions able to guide the user have been proposed. In this paper we present the Palladio Optimization Suite (POS), a collection of complementary plugins realized to run atop Palladio Bench with the aim of automatizing the exploration of the space of possible architectures by means of advances search paradigms. We also present a showcase in which the suite has been used to obtain a cost-effective and QoS-aware design of a cloud application.

The remainder of the paper is organized as follows. Section 2 showcases the Palladio Optimization Suite. A possible workflow with preliminary experimental results are presented in Section 3, whilst conclusions are drawn in Section 4.

2. Palladio Optimization Suite

The Palladio Optimization Suite, currently comprises two complementary solutions: PerOpteryx, based on genetic algorithms for multi-objective, designed for trade-off analysis and SPACE4Clouds addressing detailed configuration of a multi-cloud deployment.

2.1. **PerOpteryx.** PerOpTeryx[1] stands for *Performance Optimizer* and is a tool designed for multi-objective optimization of component-based applications [4]; it internally implements the NSGA-II evolutionary algorithm to explore the architectural space of the application under design. Its main steps are described below. The process starts by randomly generating an initial population from a candidate solution defined by the user in Palladio Component Model (PCM) format, the individuals are then modified along degrees of freedom instances. three degree of freedom types are considered here: (1) allocation of components, (2) server farm configuration, and (3) component selection. The evolutionary search then iterates the following steps:
Reproduction**:** New candidate solutions are derived by "mutation" or "cross-over" or they are randomly created. Duplicate candidates are removed from the population and replaced by candidates randomly generated.
Evaluation**:** Each unevaluated candidate is evaluated for each quality attribute of interest.
Selection**:** the population is reduced by just keeping the n most promising candidates based on the NSGA-II selection strategy. The result of the optimization is a set of Pareto optimal architectures.

2.2. **SPACE4Cloud.** SPACE4Cloud[2] (System PerformAnce and Cost Evaluation on Cloud) is an integrated environment for model-driven design-time QoS assessment and optimization of Cloud applications [3] It takes in input models in extended PCM format [1] describing the application under development in terms of functionalities, QoS requirements (e.g., average response time of *login* service below 500ms or 95% of *report generetion* service below 2s), and end-user workload profile defined over a 24-hour time horizon. Such models are converted in Layered Queueing Networks and evaluated in terms of cost and performance. The tool implements a local-search-based metaheuristic with two optimization layers:
Higher level: a Tabu-inspired mechanism decides the set of clouds alongside a VM type for each application tier.
Lower level: a local search mechanism generates a feasible solution with an adequate number of VMs for each tier and each hour of the day.
The objective is seeking for the configuration that minimizes the execution costs

[1]www.palladio-simulator.com/tools/add_ons/peropteryx/
[2]https://github.com/deib-polimi/modaclouds-space4cloud/

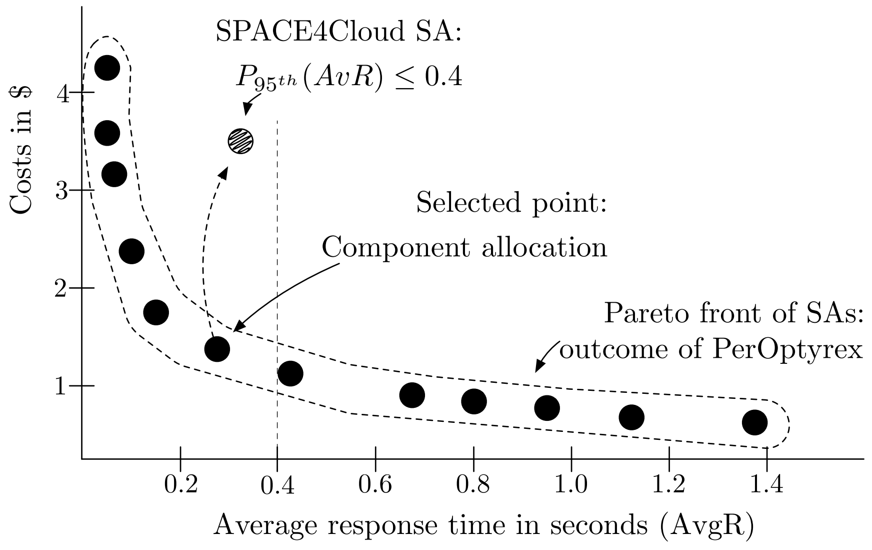

FIGURE 1. POS: example of workflow

fulfilling at once the QoS requirements. The outcome of this module is a new set of models describing the Cloud deployment in terms of type and amount of virtual resources to allocate over a daily horizon.

3. Workflow and Preliminary Experiments

In this section we introduce a possible workflow that exploits POS for the design-time optimization of a cloud application. To this end, we use a business reporting system (BRS), which lets users retrieve reports and statistical data about running business processes, as a case study. It is a 4-tier system consisting of 6 software components. The components are allocated on four different server farms (see [5] for more details). The proposed workflow is showcased in Figure 1. First we use PerOpteryx to obtain a preliminary cost-performance trade-off related to the peak workload. The outcome is a Pareto set of distinct SAs (i.e. component allocations and number of servers), optimized for the peak workload, among which we select the one that shows an adequate performance level (e.g. the average response time below 0.5 sec) minimizing the execution costs. The output of this phase is a complete PCM model, that is further decorated with a 24-hour estimate of the incoming workload and with a set of finer grained constraints (e.g. to use a certain cloud provider and to force the 95^{th} percentile of the response time to be below a threshold, say 0.4 s) . This new set of models is fed into SPACE4Clouds, which optimizes over the daily horizon choosing the type and number of VMs for each tier. For this reason the returned hourly cost can higher with respect to those of the solution selected among those of the PerOpteryx. The execution time of the two tools can be very different because dissimilar is the approach adopted and the size of the search space; whist PerOpteryx requires several hours for the considered example exploring a wider set of design, SPACE4Cloud only takes less that 30 minutes, exploiting the component allocation identified by PerOpteryx but extending its solution considering workload fluctuations during a reference working day.

4. Conclusions

In this paper, we propose a suite for multi-attribute QoS optimization of component based cloud applications. The core idea of our approach is the combination of an evolutionary optimization with a local-search-based approach. PerOpteryx allows a wide exploration of the design space to determine preliminary insights between performance and costs. In the second step, SPACE4Clouds allows to determine the optimal configuration in a Cloud deployment spanning over 24 hours and by considering also more advanced constraints (i.e., predicating on the percentiles of the application component QoS). Ongoing work is focused on the integration of the two tools and on the evaluation of the results in an industry setting. Future work will extend the proposed approach for the design-time optimization of data-intensive applications.

5. Acknowledgments

The research reported in this article is partially supported by the European Commission grants no. FP7-ICT-2011-8-318484 (MODAClouds) and H2020-ICT-2014-1-644869 (DICE).

References

[1] D. Ardagna, M. Ciavotta, M. Miglierina, G. Gibilisco, G. Casale, J. Pérez, F. D'Andria, and R. S. González. MODAClouds Deliverable D5.2.2, 2014.

[2] S. Becker, H. Koziolek, and R. Reussner. The palladio component model for model-driven performance prediction. *Journal of Systems and Software*, 82(1):3–22, 2009.

[3] D. Franceschelli, D. Ardagna, M. Ciavotta, and E. Di Nitto. Space4cloud: A tool for system performance and costevaluation of cloud systems. In *Proceedings MultiCloud '13*, pages 27–34, 2013.

[4] A. Koziolek. *Automated improvement of software architecture models for performance and other quality attributes*, volume 7. KIT Scientific Publishing, 2014.

[5] A. Koziolek, D. Ardagna, and R. Mirandola. Hybrid multi-attribute qos optimization in component based software systems. *Journal of Systems and Software*, 86(10):2542–2558, 2013.