Engineering of Self-Aware IT Systems and Services: State-of-the-Art and Research Challenges

Samuel Kounev

Institute for Program Structures and Data Organization
Karlsruhe Institute of Technology (KIT)
76131 Karlsruhe, Germany
skounev@acm.org

Modern IT systems have highly distributed and dynamic architectures composed of loosely-coupled services typically deployed on virtualized infrastructures. Managing system resources in such environments to ensure acceptable end-to-end application Quality-of-Service (QoS) while at the same time optimizing resource utilization and energy efficiency is a challenge. The adoption of Cloud Computing technologies, including Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS), comes at the cost of increased system complexity and dynamicity. This makes it hard to provide QoS guarantees in terms of performance and availability, as well as resilience to attacks and operational failures [8]. Moreover, the consolidation of workloads translates into higher utilization of physical resources which makes the system much more vulnerable to threats resulting from unforeseen load fluctuations, hardware failures and network attacks.

We present an overview of our work-in-progress and long-term research agenda focusing on the development of novel methods, techniques and tools for the engineering of so-called *self-aware* IT systems and services[1] [6, 4, 7]. The latter are designed with built-in online QoS prediction and self-adaptation capabilities used to enforce QoS requirements in a cost- and energy-efficient manner. The current focus is on performance, availability and efficiency aspects, however, long-term we are planning to consider further QoS properties such as reliability and fault-tolerance. Self-awareness, in this context, is defined by the combination of three properties that IT systems and services should possess:

1. *Self-reflective*: i) aware of their software architecture, execution environment and the hardware infrastructure on which they are running, ii) aware of their operational goals in terms of QoS requirements, service-level agreements (SLAs) and cost- and energy-efficiency targets, iii) aware of dynamic changes in the above during operation,
2. *Self-predictive*: able to predict the effect of dynamic changes (e.g., changing service workloads or QoS requirements) as well as predict the effect of possible adaptation actions (e.g., changing service deployment and/or resource allocations),

[1] http://www.descartes-research.net

3. *Self-adaptive*: proactively adapting as the environment evolves in order to ensure that their QoS requirements and respective SLAs are continuously satisfied while at the same time operating costs and energy-efficiency are optimized.

Our approach to the realization of the above vision is based on the use of *online* service architecture models integrated into the system components and capturing all service aspects relevant to managing QoS and resource efficiency during operation [2, 10, 7]. In contrast to black-box models, the modeling techniques we are working on are designed to explicitly capture all relevant aspects of the underlying software architecture, execution environment, hardware infrastructure, and service usage profiles. In parallel to this, we are working on self-aware service platforms designed to automatically maintain models during operation to reflect the evolving system environment. The online models will serve as a "mind" to the running systems controlling their behavior, i.e., deployment configurations, resource allocations and scheduling decisions. To facilitate the initial model construction and continuous maintenance during operation, we are working on techniques for automatic model extraction based on monitoring data collected at run-time [1, 5, 3].

The online service architecture models are intended to be used during operation to answer QoS-related queries such as: What would be the effect on the QoS of running applications and on the resource consumption of the infrastructure if a new service is deployed in the virtualized environment or an existing service is migrated from one server to another? How much resources need to be allocated to a newly deployed service to ensure that SLAs are satisfied while maximizing energy efficiency? What QoS would a service exhibit after a period of time if the workload continues to develop according to the current trends? How should the system configuration be adapted to avoid QoS problems or inefficient resource usage arising from changing customer workloads? What operating costs does a service hosted on the infrastructure incur and how does the service workload and usage profile impact the costs? We refer to such queries as *online QoS queries*.

The ability to answer online QoS queries during operation provides the basis for implementing novel techniques for self-aware QoS and resource management [7, 2, 10]. Such techniques are triggered automatically during operation in response to observed or forecast changes in the environment (e.g., varying service workloads). The goal is to *proactively* adapt the system to such changes in order to avoid anticipated QoS problems, inefficient resource usage and/or high system operating costs. The adaptation is performed in an autonomic fashion by considering a set of possible system reconfiguration scenarios (e.g, changing VM placement and/or resource allocations) and exploiting the online QoS query mechanism to predict the effect of such reconfigurations before making a decision [2].

Each time an online QoS query is executed, it is processed by means of the online service architecture models which are composed dynamically after determining which specific parts of the system are relevant to answering the query. Given the wide range of possible contexts in which the online service mod-

els can be used, automatic model-to-model transformation techniques (e.g., [9]) are used to generate tailored prediction models on-the-fly depending on the required accuracy and the time available for the analysis. Multiple prediction model types (e.g., queueing networks, stochastic Petri nets, stochastic process algebras and general-purpose simulation models) and model solution techniques (e.g., exact analytical techniques, numerical approximation techniques, simulation and bounding techniques) are employed here in order to provide flexibility in trading-off between prediction accuracy and analysis overhead.

Self-Aware Service Engineering [4, 6] is a newly emerging research area at the intersection of several computer science disciplines including Software and Systems Engineering, Computer Systems Modeling, Autonomic Computing, Distributed Systems, Cluster and Grid Computing, and more recently, Cloud Computing and Green IT (see Figure 1). The realization of the described vision calls for an interdisciplinary approach considering not only technical but also business and economical challenges. The resolution of these challenges promises to reduce the costs of ICT and their environmental footprint while keeping a high growth rate of IT services.

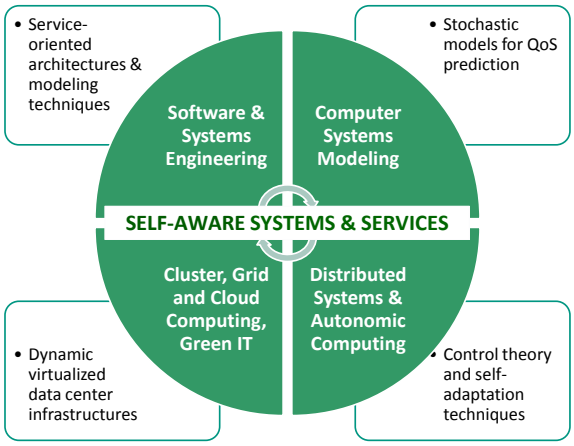

Fig. 1. Self-Aware Service Engineering

References

1. F. Brosig, N. Huber, and S. Kounev. Automated Extraction of Architecture-Level Performance Models of Distributed Component-Based Systems. In *26th IEEE/ACM International Conference On Automated Software Engineering (ASE 2011), November 6-11, Oread, Lawrence, Kansas*, 2011.
2. N. Huber, F. Brosig, and S. Kounev. Model-based Self-Adaptive Resource Allocation in Virtualized Environments. In *SEAMS'11: 6th International Symposium*

on Software Engineering for Adaptive and Self-Managing Systems, May 23-24, Waikiki, Honolulu, Hawaii, USA. ACM Press, 2011.

3. N. Huber, M. von Quast, M. Hauck, and S. Kounev. Evaluating and Modeling Virtualization Performance Overhead for Cloud Environments. In *International Conference on Cloud Computing and Service Science (CLOSER 2011), May 7-9, Noordwijkerhout, The Netherlands*, 2011.
4. S. Kounev. Self-Aware Software and Systems Engineering: A Vision and Research Roadmap. In *GI Softwaretechnik-Trends, ISSN 0720-8928. Proceedings of Software Engineering 2011 (SE 2011), Nachwuchswissenschaftler-Symposium, February 21-25, Karlsruhe, Germany*, 2011.
5. S. Kounev, K. Bender, F. Brosig, N. Huber, and R. Okamoto. Automated Simulation-Based Capacity Planning for Enterprise Data Fabrics. In *4th International ICST Conference on Simulation Tools and Techniques, March 21-25, Barcelona, Spain*, 2011.
6. S. Kounev, F. Brosig, and N. Huber. Self-Aware QoS Management in Virtualized Infrastructures (Poster Paper). In *8th International Conference on Autonomic Computing (ICAC 2011), June 14-18, Karlsruhe, Germany*, 2011.
7. S. Kounev, F. Brosig, N. Huber, and R. Reussner. Towards self-aware performance and resource management in modern service-oriented systems. In *Proceedings of the 7th IEEE International Conference on Services Computing (SCC 2010), July 5-10, Miami, Florida, USA*. IEEE Computer Society, 2010.
8. S. Kounev, P. Reinecke, K. Joshi, J. Bradley, F. Brosig, V. Babka, S. Gilmore, and A. Stefanek. Providing Dependability and Resilience in the Cloud: Challenges and Opportunities. In A. Avritzer, A. van Moorsel, K. Wolter, and M. Vieira, editors, *Resilience Assessment and Evaluation*, Dagstuhl Seminar 10292. Springer Verlag, 2011.
9. P. Meier, S. Kounev, and H. Koziolek. Automated Transformation of Palladio Component Models to Queueing Petri Nets. In *19th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2011), July 25-27, Singapore*, 2011.
10. R. Nou, S. Kounev, F. Julia, and J. Torres. Autonomic QoS control in enterprise Grid environments using online simulation. *Journal of Systems and Software*, 82(3):486–502, Mar. 2009.