SPACE4CLOUD: A DEVOPS ENVIRONMENT FOR MULTI-CLOUD APPLICATIONS

Michele Guerriero
Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria.
michele.guerriero@polimi.it
Michele Ciavotta
Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria.
michele.ciavotta@polimi.it
Giovanni Paolo Gibilisco
Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria.
giovannipaolo.gibilisco@polimi.it
Danilo Ardagna
Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria.
danilo.ardagna@polimi.it

Abstract. Cloud computing has been a game changer in the design, development and management of modern applications, which have grown in scope and size becoming distributed and service oriented. New methodologies have emerged to deal with this paradigm shift in software engineering. Consequently, new tools, devoted to ease the convergence between developers and other IT professional, are required. Here, we present SPACE4Cloud, a DevOps integrated environment for model-driven design-time QoS assessment and optimization, and runtime capacity allocation for Cloud applications.

Model-Driven, Cloud, QoS, design-time, runtime, DevOps

1. Introduction

In recent years we have witnessed a paradigm shift in the software creation and management. Projects have progressively grown in size and scope, and the legacy model of standalone applications has lost much of its relevance in favor of more flexible, distributed, and web-based architectures. Another factor to consider in this change is the emergence and the success of Cloud computing. Overall, new ideas for application development and management have appeared, which resulted in DevOps, *i.e.*, a software development method based on collaboration, more often on a real convergence, between software developers, system administrators, and performance engineers. Under these circumstances, tools that simplify the early quality evaluation at design-time and systems for managing the quality at runtime have become especially important. In this work we propose SPACE4Cloud, a collection of tools developed within the MODAclouds[1] EU FP7 project for design-time modeling and analysis, and runtime quality management of multi-Cloud applications.

The rest of the paper is organized as follows: in Section 2 we introduce SPACE4Cloud; some experimental results are presented in Section 3 whereas conclusions are finally drawn in Section 4.

[1] www.modaclouds.eu

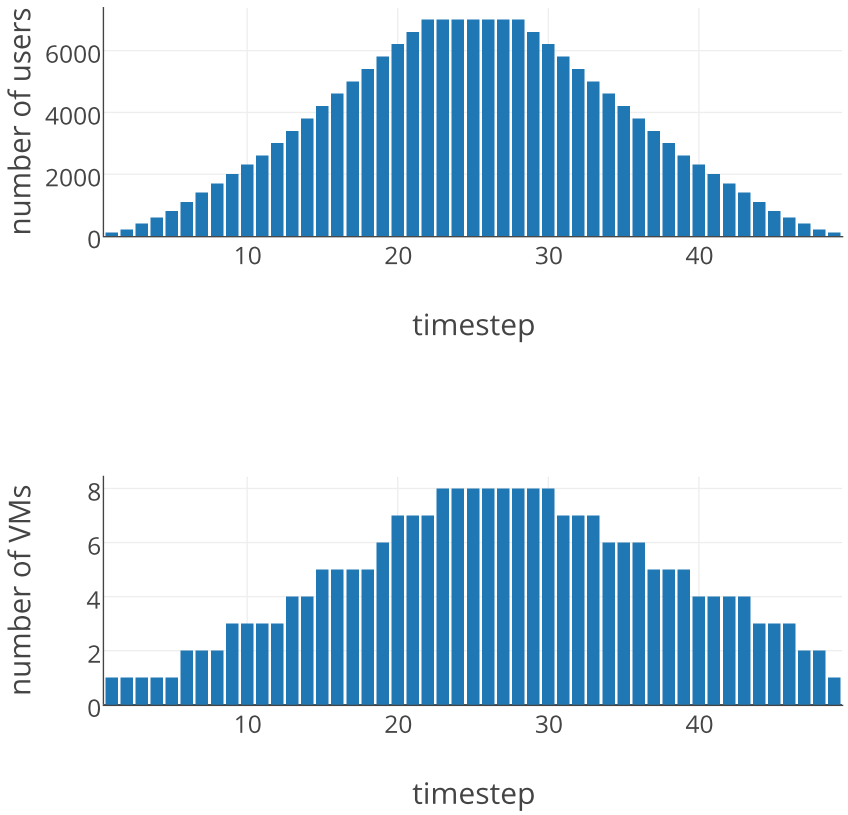

Figure 2. Varying number of VMs

2. SPACE4Cloud

SPACE4Cloud (System PerformAnce and Cost Evaluation on Cloud) is an integrated environment for model-driven design-time QoS assessment, optimization, and runtime capacity allocation of Cloud applications. It is composed of two main tools: at design-time **SPACE 4Cloud**^{Dev} takes in input models in extended PCM format [2] describing the application under development in terms of functionalities, Quality of Service (QoS) requirements, and end-user workload profile defined over a 24-hour time horizon. Such models are converted in Layered Queueing Networks [4] and evaluated in terms of cost and performance. The tool is also able to perform, through a local-search-based metaheuristic, a fully automatized exploration of the space of possible Cloud offers, seeking for the configuration that minimizes the execution costs fulfilling at once the QoS requirements. The outcome of this module is a new set of models describing the Cloud deployment and the runtime adaptation actions. These models are, in turn, fed into **SPACE4Cloud**^{Ops}, which is the tool in charge of guaranteeing at runtime the pledged QoS levels by enacting suitable policies. This solutions implements the so-called Receding Horizon Control paradigm. Simply put, it solves a Mixed Integer Linear Programming (MILP) optimization problem over a finite time window; this process generates a set of scaling actions to be implemented within the MODAClouds Runtime Platform [3], using a suitable IaaS interface, in order to obtains a satisfactory QoS during the time window, yet only the adaptations for the first time step are enforced. Eventually, the process is repeated considering the second time step as the beginning of a new time window. The MILP problem, in its basic formulation, is a capacity allocation problem in charge to determine the number of VMs suitable to serve the expected incoming workload, minimizing costs and guaranteeing that the QoS requirements are met. The optimization model, based on queueing theory results, is characterized by performance parameters that are continuously updated at runtime by an appropriate monitoring system (see [3]) in order to cope with the time-varying behavior of the Cloud. Moreover, since the workload varies over time and it can not be known in advance, the allocation problem is solved based on a prediction, also provided by the monitoring system.

The entire SPACE4Cloud environment is implemented in Java, released under Apache License 2.0[2] [3]

3. Experimental Results

In this section we briefly describe some of the experiments performed to prove the soundness of our approach. The experiments are all performed using a simplified single-tier web application and leveraging Amazon EC2 services. Embracing a model-base approach, we start by modeling our application and the operational environment at design-time using the extended PCM format. Along this path, initial estimates of service times are derived by performing some preliminary experiments on a prototype environment and relying on state-of-the-art parameter estimation techniques [5]. In this way we could describe the application and the runtime environment, and feed with the resulting models **SPACE4Cloud**^{Dev}. Once the application has been automatically deployed in the MODAClouds runtime environment, **SPACE4Cloud**^{Ops} starts creating the optimization model and enacting the runtime control loop with a 5 minute timescale against a synthetic workload generated by Apache JMeter[4]. A timescale of 5 minutes has been set up since it has been proved in [1] to provide better performance with respect to larger ones, essentially due to more accurate workload predictions available at this scale. Moreover in [1] a comparison with a heuristic currently implemented by some IaaS providers is reported, showing that our approach always provides better solutions. Here, focusing on the effectiveness of our approach, a basic experiment carried out using the 4-hour workload profile shown in Figure 1 is presented; the incoming workload is basically a ramp up with a maximum number of users equal to 7,000, followed by a ramp down. Figure 2 shows the number of running VMs varying in the range [1, 8] following the shape of the workload, proving the effectiveness of the adaptive mechanism provided by **SPACE 4Cloud**^{Ops}.

4. Conclusions

In this work we presented a joint DevOps environment for design-time modeling and optimization, and runtime control for Cloud applications. The aim of the tool

[2] github.com/deib-polimi/modaclouds-space4cloud

[3] github.com/deib-polimi/modaclouds-autoscalingReasoner

[4] jmeter.apache.org

is to minimize the execution costs of Cloud applications providing QoS guarantees by design. The most distinguished characteristic of the Cloud, such as variable workload, congestion due to multi-tenancy, and performance variability, are considered. Future work will be devoted mainly to extend the runtime adaptive actions to PaaS platforms and multi-Cloud applications, both already managed at design-time. Moreover, a feedback mechanism will be implemented in MODAClouds runtime environment to provide a better estimation of design-time parameters. In this way the user, dealing with an application model closer to reality, can further improve the deployment, discovering and solving possible performance bottleneck. Finally, the tool will be extended for design-time modeling and optimization of data intensive applications within the framework of the DICE[5] project.

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, and R. Lancellotti. A receding horizon approach for the runtime management of iaas cloud systems. In *SYNASC-MICAS 2014*.

[2] D. Ardagna, M. Ciavotta, M. Miglierina, G. Gibilisco, G. Casale, J. Pérez, F. D'Andria, and R. S. González. MODAClouds D5.2.2 - MODACloudML QoS abstractions and prediction models specification, 2014.

[3] G. Iuhasz, S. Panica, G. Casale, W. Wang, P. Jamshidi, D. Ardagna, M. Ciavotta, D. Whigham, N. Ferry, and R. S. González. MODAClouds D6.4.2 - runtime environment final release, 2015.

[4] J. A. Rolia and K. C. Sevcik. The method of layers. *IEEE Trans. Softw. Eng.*, 21(8):689–700, Aug. 1995.

[5] L. Zhang, X. Meng, S. Meng, and J. Tan. K-scope: Online performance tracking for dynamic cloud applications. In *ICAC 2013*.

[5] www.dice-h2020.eu