Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A.A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

AN ARCHITECTURE FOR AN AUTOSCALING CLOUD-BASED SYSTEM FOR SIMULATION EXPERIMENTATION

Simon J. E. Taylor
Anastasia Anagnostou

Modelling & Simulation Group
Department of Computer Science
Brunel University London
Uxbridge, UB8 3PH, UK

Tamas Kiss

Centre for Parallel Computing
University of Westminster
London, W1W 6XH, UK

Gary Pattison
Shane Kite

Saker Solutions
Upper Courtyard, Ragley Hall
Alcester, Warwickshire, B49 5NL, UK

Jozsef Kovacs
Peter Kacsuk

MTA SZTAKI
Budapest
Hungary

ABSTRACT

More and more simulation applications need high performance computing to deliver the results from experimentation in a timely manner. Cloud computing presents an attractive cost-effective alternative to using a local computing cluster. Normally a user would decide how many cloud computing resources to hire, provision them and then use them for experimentation. However, it may be the case that the user has paid for many instances that were not used. We have proposed the Microservice-based Cloud Application-level Dynamic Orchestrator (MiCADO) to automatically orchestrate and scale cloud computing applications. This article describes the architecture of a version of MiCADO that has been adapted for simulation experimentation.

1 INTRODUCTION

There is a growing demand for faster simulation experimentation and optimisation. Cloud computing presents an attractive alternative to expensive computing clusters. Based on experiences from developing a cloud computing platform for simulation (Taylor, et al. 2018), we have produced a platform that enables cloud resources to be automatically increased, or scaled, according to the needs of an application. This is the Microservice-based Cloud Application-level Dynamic Orchestrator (MiCADO) (Kiss, et al. *in press*). This paper describes how MiCADO is being used to support simulation experimentation.

2 THE MICROSERVICE-BASED CLOUD APPLICATION-LEVEL DYNAMIC ORCHESTRATOR (MICADO) AND SIMULATION EXPERIMENTATION

MiCADO is based on microservices that decouple independent components from a monolithic application. The concept of MiCADO is to monitor the needs of a microservice (e.g. the time being taken for a simulation experiment to complete) and then to launch new instances of that microservice on cloud to cope with demand (e.g. multiple cloud-based simulation microservices that process experiments in parallel). Figure 1 shows the MiCADO architecture.

The Application layer contains the application that requires cloud resources (the simulation package or frontend). The Application Definition layer defines a functional architecture of the application using

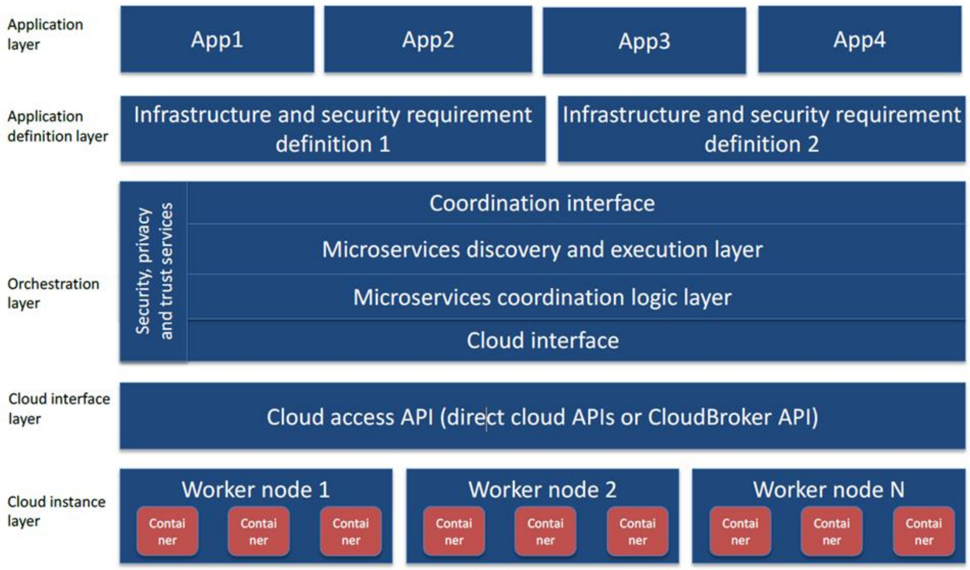

Figure 1: The MiCADO Generic Architecture

an application template that defines the required infrastructure, interconnectivity and Quality of Service specifications. The Orchestration layer consists of four horizontal and one vertical sub-layers: a Coordination interface API for orchestration control (e.g. control over how microservices are run), a Microservices discovery and execution layer to manages microservices execution, a Microservices coordination logic layer that decides when to launch/shut down cloud instances to run new microservices, a Cloud interface API that abstracts cloud access from the above layers (MiCADO can use many different clouds), and a Security layer providing support for advanced security policy management. The Cloud access API is used to actually launch and shut down cloud instances (potentially on multiple clouds). Finally, the Cloud instance layer contains the actual cloud instances provided by IaaS cloud providers.

To use MiCADO for simulation experimentation a developer would first create a simulation microservice. An application template would then be specified and would include the maximum runtime for a particular set of experiments. MiCADO would then launch one instance of the microservice and montor how long simulation experimentation was taking. If the deadline was not going to be met, MiCADO would launch a new microservice/instance. This process would continue until experimentation was complete. In this way the optimum number of cloud instances (and cost) would be used.

3 SUMMARY

This paper has briefly described the MiCADO architecture and how it is used for simulation experimentation.

ACKNOWLEDGMENTS

This work was initially partially supported by the EU H2020 Cloud Orchestration at the Level of Application (COLA) project (Contract no. 731574).

REFERENCES

Kiss, T. P. Kacsuk, J. Kovacs, B. Rakoczi, A. Hajnal, A. Farkas, G. Gesmier, G. Terstyanszky. *In press.* "MiCADO—Microservice-based Cloud Application-level Dynamic Orchestrator", *Future Generation Computer Systems*, https://doi.org/10.1016/j.future.2017.09.050.

Taylor, S.J.E., T. Kiss, A. Anagnostou, G. Terstyanszky, P. Kacsuk, J. Costes, N. Fantini. 2018. *Future Generation Computer Systems* 88: 524-539.