Abstract: Automated Scalability of Cloud Services and Jobs

Tamas Kiss, Gabor Terstyanszky, Osama Abu Oun, James DesLauriers, Gregoire Gesmier, Gabriele Pierantoni,
University of Westminster, London UK

Jozsef Kovacs, Peter Kacsuk, Eniko Nagy, Attila Farkas
MTA-SZTAKI, Budapest, Hungary

ABSTRACT

Many scientific and commercial applications require access to computation, data or networking resources based on dynamically changing requirements. Users and providers both require these applications or services to dynamically adjust to fluctuations in demand and serve end-users at required quality of service (performance, reliability, security, etc.) and at optimized cost. This may require resources of these applications or services to automatically scale up or down.

The European funded COLA (Cloud Orchestration at the Level of Application) project aims to design and develop a generic framework that supports automated scalability of a large variety of applications. Learning from previous similar efforts and with the aim of reusing existing open source technologies wherever possible, COLA elaborated a modular architecture called MiCADO (Microservices-based Cloud Application-level Dynamic Orchestrator) [1] that provides optimized deployment and run-time orchestration for cloud applications.

MiCADO is built from well-defined building blocks implemented as microservices. This modular design supports various implementations where components can be replaced relatively easily with alternative technologies. The generic, technology independent architecture diagram of MiCADO is represented in Figure 1. Building blocks, both on the MiCADO Master and also on the MiCADO Worker Nodes are implemented as microservices. The current implementation uses widely applied technologies, such as Docker Swarm as Container Orchestrator [2], Occopus as Cloud Orchestrator [3], and Prometheus [4] as the Monitoring System.

The user facing interface of MiCADO is a TOSCA (Topology and Orchestration Specification for Cloud Applications, an OASIS standard) [5] based description of the desired topology and its associated scalability and security policies. This interface can then be embedded to existing GUIs, custom web interfaces or science gateways.

The first prototype implementations of MiCADO show promising results on various application types. The two main targeted application categories are cloud-based services where scalability is achieved by scaling up or down the number of containers and virtual machines based on load, performance and cost, and the execution of a large number of (typically parameter sweep style) jobs where a certain number of these jobs need to be executed by a set deadline.

This work was partially supported by the COLA Cloud Orchestration at the level of Applications project, Project No. 731574.

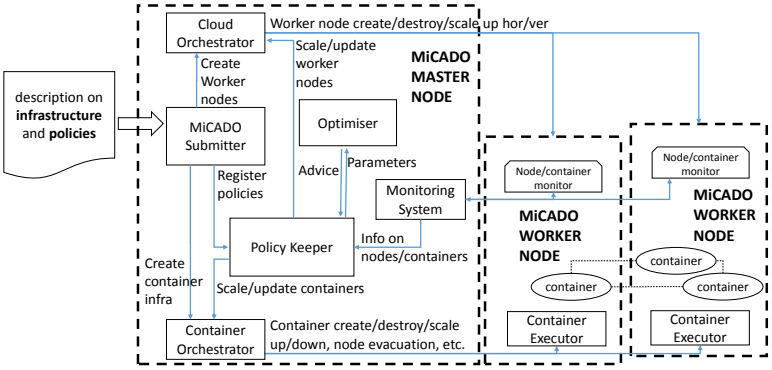

Figure 1 – MiCADO Generic Architecture Diagram

Direct involvement of industry partners assures that the results of COLA are prototyped on real application scenarios. Three near production quality demonstrators and twenty further proof of concept case studies are being implemented using MiCADO and demonstrating its applicability in case of both service and job type scalability. Some of the applications prototyped are directly related to services utilized in science gateways, such as the Data Avenue service of WS-PGRADE [6].

Keywords—application level cloud orchestration; automated scalability; microservices; container techonlogies

REFERENCES

[1] Kiss T, et al., MiCADO – Microservice-based Cloud Application-level Dynamic Orchestrator, in Future Generation Computing Systems, https://doi.org/10.1016/j.future.2017.09.050, 2017

[2] Docker Swarm, [online] Available from: http://www.docker.com

[3] Kovács J, Kacsuk P., Occopus: a Multi-Cloud Orchestrator to Deploy and Manage Complex Scientific Infrastructures, Journal of Grid Computing, Volume 16, Issue 1, 2018, pp. 19-37.

[4] Prometheus, [Online] Available from: https://prometheus.io/

[5] OASIS Topology and Orchestration Specification for Cloud Applications Version 1.0, [online] Available from: http://docs.oasis-open.org/tosca/TOSCA/v1.0/TOSCA-v1.0.html

[6] Hajnal A, Farkas Z, Kacsuk P: Data Avenue, Remote Storage Resource Management in WS-PGRADE, in proceedings of IWSG 2014, 6th International Workshop on Science Gateways, IEEE, 25 August 2014, **DOI:** 10.1109/IWSG.2014.7