Cloud Orchestration at the Level of Application

Project Acronym: **COLA**

Project Number: **731574**

Programme: **Information and Communication Technologies**
Advanced Computing and Cloud Computing

Topic: **ICT-06-2016 Cloud Computing**

Call Identifier: **H2020-ICT-2016-1**
Funding Scheme: **Innovation Action**

Start date of project: 01/01/2017 Duration: 30 months

Deliverable:

D4.3 Upgraded COLA development testbed and production infrastructure

Due date of deliverable: 31/12/2018 Actual submission date: 30/01/2019

WPL: Peter Gray

Dissemination Level: PU

Version: draft

1 Table of Contents

2 List of Figures and Tables

Figures

Tables

3 Status, Change History and Glossary

Status:	Name:	Date:
Draft:	Peter Gray	05/12/2018
Reviewed:	József Kovács	25/01/2019
Approved:	Tamas Kiss	30/01/2019

Table 1 - Status Change History

Version	Date	Author	Modification
v0.1	05/12	CloudSigma	Structure document
v0.2	10/12	CloudSigma	Ongoing Assessment of requirements
v0.3	19/12	CloudSigma	Troubleshooting issues and maintenance
v0.4	7/01	CloudSigma	Improvements and upgrades to the infrastructure layer
v0.5	11/01	University of Westminster	Performance comparisons
v0.6	14/01	CloudBroker, ScaleTools	Improvements and upgrades to the cloud access layer
v0.7	16/01	CloudSigma	Introduction
v0.8	18/01	CloudSigma	Summary of the transition from testbed infrastructure to production infrastructure
v0.9	21/01	CloudSigma	Upcoming tasks in WP4
v1.0	22/01	CloudSigma	Conclusion
v1.1	23/01	CloudSigma	Figures/tables, glossary, references.

Table 2 - Deliverable Change History

4 Glossary

ADT	Application Description Template
API	Application Programme Interface
DNAT	Destination Network Address Translation
DNS	Domain Name Server
ICMP	Internet Control Message Protocol
GUI	Graphical User Interface
SQL	Structured Query Language
SNAT	Stateful Network Address Translation
TCP	Transmission Control Protocol
YAML	Yet Another Markup Language / YAML Ain't Markup Language
QoS	Quality of Service

Table 3 – Glossary

5 Introduction

This deliverable covers the work done since the submission of D4.2 to move from testbed infrastructure to production ready infrastructure. We document the technical support provided to project partners to ensure performant utilisation of the underlying IaaS cloud infrastructure by the MiCADO services layer. The performance benchmarking results that were reported in D4.2, as well as issue resolution from one-to-one interaction with project partners have formed the basis for the successful provisioning of the underlying cloud infrastructure that can now be considered production ready as we head into the final phase of the project. Where this deliverable differs from the last, is that our focus has been more on functionality and user-experience rather than performance benchmarking.

In Section 6 we briefly cover the continuous assessment of requirements and outline our approach to dealing with related issues raised by project partners. We provide more details about this in Section 7 by outlining the specific troubleshooting activities and general maintenance activities of the cloud infrastructure. We continue in Section 8 to describe the improvements and upgrades to the production cloud infrastructure provided by CloudSigma in relation to the feedback received from the project partners. In Section 9 we provide a summary of the work carried out as we move from development testbed to production cloud. We continue in Section 10 to outline the ongoing tasks that will be implemented from now until the end of the project and beyond. We end this report by concluding our achievements over the duration of the project.

This deliverable continues on from the previous deliverable and relates mainly to work carried out as part of Task 4.3: Deployment, operation and support of production infrastructure. As this is the last deliverable covering the advancement from testbed to production infrastructure, we measure our success against all the objectives under WP4. The results outlined in this deliverable inform Task 4.5 regarding the implementation of necessary enhancements to the cloud access layer, WP5 regarding QoS and scaling of services, and WP6 to advise on price/performance optimisation. The following WP4 deliverable will concentrate on the enhancements to the cloud access layer.

6. Ongoing assessment of requirements

As reported in D4.2, we established early in the project what the high-level requirements common to the use-cases were. These were extrapolated from analysis of the outputs documented in the following deliverables:

1. D8.1: Business and Technical Requirements of COLA Use-Cases, and D8.2: Customisation and Further Development of Software Applications. The high-level requirements common to the use-cases were extrapolated and assessed to understand whether the performance of the current MiCADO implementation in the cloud is likely to meet the demands of the applications.

2. D5.4: First Set of Template and Services of Use Cases. The application-level performance requirements were extrapolated from the Application Description Templates (ADTs) and definition of services relating to each use-case.

Requirements were also defined by SZTAKI towards the CloudBroker platform including functionality such as cloud-init support and port range handling. This will be presented in D4.4 Upgraded CloudBroker Platform in M30.

Since the publishing of the aforementioned deliverables, our approach was to communicate directly with partners within the project and to deal with issues on a case by case basis. Any issues directly relating to the cloud infrastructure provided by CloudSigma were dealt with by creating a ticket in their issue tracking platform JIRA. Issues were then slotted into their internal bi-weekly sprints and resolved whenever possible. We outline these issues in more detail in the following section.

7. Troubleshooting Issues and Maintenance

Each partner providing testbed infrastructure to the project has been continuing to provide technical support as well as scheduling regular maintenance of their respective clouds. As a result, the testbed infrastructure has been more than adequate for the project needs.

As the need to move from testbed infrastructure to production infrastructure became more important, the focus on identifying any technical issues relating to CloudSigma's production cloud became a priority. SZTAKI and UoW identified a number of issues while doing some extended development and testing of MiCADO on CloudSigma. They also noted some functional limitations for partners implementing WP6 tasks. A list was compiled and provided to CloudSigma. In the table below, we present these issues along with the corresponding resolution and status. It should be noted that not all issues can be resolved immediately due to technical limitations of the core infrastructure. However, issues marked as critical have been prioritised and will be resolved before the end of the project.

Issue Identified	Impact	Intended resolution	Status
Google Chrome not supported by the Web UI at https://zrh.cloudsigma.com/ui/ Error message: "We have encountered some technical difficulties, please contact support via live chat or email." Chrome version: 69.0.3497.100 (Official Build) (64-bit), OS version: Windows 10.	Low	Issue escalated through CloudSigma technical support to the front-end developers.	In progress
CloudSigma firewall opens up all ports in case at least one port is requested by the user to open.	High	This is due to the limitations of the underlying ebtables based firewall. This functionality will not be included immediately.	Pending review

Missing ICMP configurability in CloudSigma firewall.	Medium	This is not intended functionality.	Closed
Contextualisation of Windows virtual machine is not yet supported.	High	In order to make the necessary configurations on boot, cloud-init is used for Linux based deployment. This functionality is originally not intended to be used in Windows so we have come up with a bespoke solution to this problem.	Resolved
Instantiated virtual machines produces high fluctuation in performance.	High	Some fluctuation in performance is expected as CloudSigma by nature is a shared cloud environment. However, due to a recent hardware upgrade across all data centres, performance is now more consistent.	Resolved
Occasionally CloudSigma DNS server outage happens.	Low	Issue escalated to CloudSigma technical support.	In progress
Occasionally *very* long _drive cloning_ and _node starting_ times.	Medium	This was caused by issues in the software stack.	Resolved
No information in API docs about "max_retries_exceeded" error response returned by CloudSigma API.	Medium	Further details required. Likely related to the same software stack issues that were later resolved.	In progress

Table 4 – Issues identified

At the time of submitting this deliverable; 3 issues have been resolved, 1 issue has been closed as it is not in scope, 2 issues are in progress and due to be resolved before the end of the project, and one issue will not be resolved by the end of the project as it involves an upgrade that CloudSigma is unable to implement within the time remaining.

In the following section we provide information about the improvements and upgrades that resulted from the feedback from partners.

Further work is being done on improving the communication between cloud access layer and cloud infrastructure upon provisioning. In some edge cases the cloud access layer is not correctly handling the output of the cloud provider API. These changes will be described in more detail in deliverable 4.4.

Additionally, research was done as to what the optimal CPU clock size of the virtual machines should be. It is based on the physical CPU core size of the CloudSigma hardware

hosts CPUs and reported for consideration in person during a project meeting to the developers of the cloud access layer. The virtual machine size is preset, so any modifications in the parameters needs to be implemented in the cloud access layer.

In the duration of the project and due to the movement to a production infrastructure, more hardware hosts were added to CloudSigma's Zurich location, where the MiCADO tests and operation are being performed. The general elasticity of the cloud concept allowed CloudSigma to seamlessly introduce upgrades to the platform, without the need for a migration or any actions on the user side. The upgrades included more hosts with improved hardware specifications, improvements to the network infrastructure and various maintenance actions in our datacenter. CloudSigma runs its own proprietary cloud stack and the introduction of more hardware costs had unexpected side effects for the underlying software. This resulted in several issues, observed by the MiCADO development team and the experiment partners:

- intermittent failures to perform some calls to the CloudSigma API
- abnormally high execution times for some background tasks, resulting in incorrect resource status
- intermittent unavailability of cloud resources

These issues were intermittently experienced for the total duration of one month. During that time CloudSigma provided an alternative account to the MiCADO developers in another datacenter, while the development team rectified the problematic behaviour of the production infrastructure. Afterwards, all MiCADO operations resumed as expected and enjoyed better performance compared to the test phase.

8. Improvements and upgrades to the infrastructure layer and the cloud access layer

In this section we outline the improvements and upgrades to the production cloud provided by CloudSigma. In most cases any improvements made impact all of CloudSigma's data centre locations as they are made to the software stack. However, the hardware upgrade for improving performance relates only to CloudSigma's Zurich data centre location.

We also provide some details about the various improvements and upgrades to the CloudBroker platform. More substantial information will be provided in the Deliverable 4.4: Upgraded CloudBroker Platform due to be submitted in month 30 of the project.

8.1 Server context in Windows

Server context is a way for the VM to get information on the way it was set-up, i.e. get it's definition. Unlike most cloud public cloud providers, CloudSigma's server context is communicated over a virtual serial port device, which on UNIX-like operating system would usually appear as /dev/ttyS1 and on Windows as COM2. While the general practice for cloud providers is to allow the retrieval of this information via a network protocol, CloudSigma's approach means that this information can be accessed by the VM before networking is

established and with the use of cloud-init the networking itself can be configured as per the user definition.

Having the server definition accessible by the VM can be useful in various ways. For example, it is possible to easily determine from within the VM, which network interfaces are connected to public and which to private network. Another use is to pass some data to initial VM setup scripts, like setting the hostname to the VM name or passing ssh public keys through server metadata.

At first sight, it might be confusing with the presence of both ‘server context’ and ‘server metadata’. ‘server metadata’ is really a subset of ‘server context’. The ‘server metadata’ itself is a key-value store for user-defined data on a server definition. The ‘server context’ on the other hand is one step above. It includes the full server definition, as well as the server metadata, along with attached drives definitions.

The above described method has been widely used on CloudSigma for Unix based operating systems, but using it under Windows had not been attempted. This posed further challenges because of operating system specifics and differences. CloudSigma’s development team made extensive tests of the behaviour of the virtualised serial port in Windows and eventually came up with a solution that handles the operating system specifics and outputs the necessary information from the VM definition.

Further development of a custom Python script was needed to deliver the cloud-init functionality requested by SZTAKI. CloudSigma developed the solution which combines serial port access and the ability to provision arbitrary files and execute commands based on a YAML definition identical to the cloud-init one, which is the basic contextualisation tool for MiCADO on Unix based operating systems.

8.2 Consistent performance

Some partners were experiencing fluctuation in performance on CloudSigma. While some inconsistency is to be expected in a shared cloud environment, CloudSigma decided to make some hardware upgrades to their infrastructure in their Zurich data centre and match performance with the companies newer data centres in Frankfurt. UoW took the opportunity to re-run some performance tests against the recent hardware upgrades. The tests were re-run on 7 instances via CloudBroker. A total of 120 jobs were run. We include jobs 1-48 in the graph below, which is indicative of the results across all the testing.

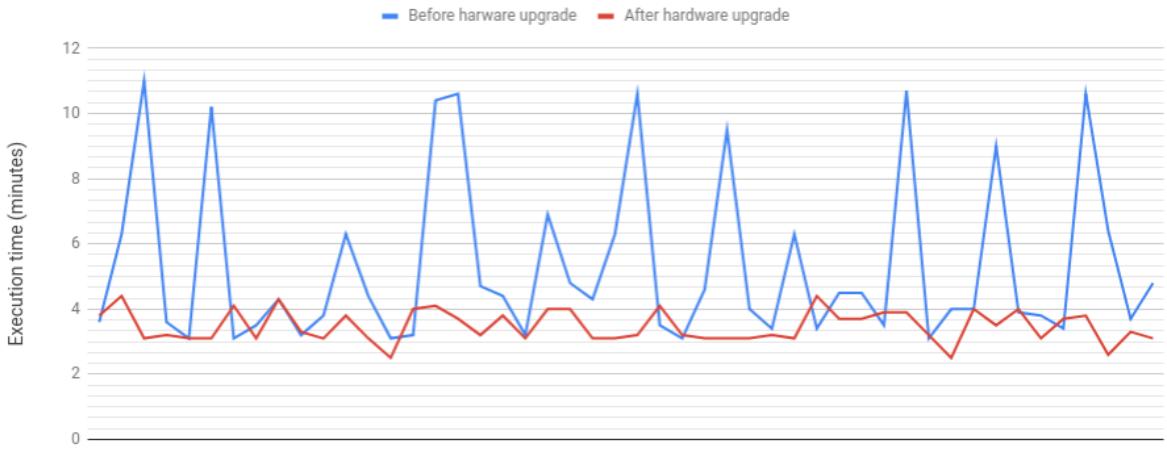

Figure 1: CloudSigma performance results before and after hardware upgrade

We can clearly see that the execution time has improved and is much more consistent. This is also in line with results performed on CloudSigma's newer Frankfurt data centre.

8.3 CloudSigma firewall functionality

Currently, it is not possible to filter ports. This is due to CloudSigma's firewall being based on the ebtables program, a filtering tool for a Linux-based bridging firewall. It enables transparent filtering of network traffic passing through a Linux bridge. The filtering possibilities are limited to link layer filtering and some basic filtering on higher network layers. Advanced logging, MAC DNAT/SNAT and brouter facilities are also included. The ebtables tool can be combined with the other Linux filtering tools (iptables, ip6tables and arptables) to make a bridging firewall that is also capable of filtering these higher network layers. This is enabled through the bridge-netfilter architecture which is a part of the standard Linux kernel.

To summarise, it is not possible to DENY connections to all ports and ALLOW just one single port. Also by denying all ports this would create issues for programs that make outbound connections as well. When they establish a TCP session the incoming traffic would still be filtered as ebtables is stateless and would not recognize an established connection state.

This upgrade is pending review by CloudSigma's management team as it will have a commercial impact. If it is deemed critical and is in-line with CloudSigma's existing business model it will be scheduled in their development roadmap. Until this upgrade is implemented, CloudSigma has advised partners not to use the CloudSigma firewall for these needs.

8.4 Changes to the CloudBroker platform

The operation, maintenance and support for the CloudBroker platform has been ongoing, while a number of changes and upgrades were made to the platform over the last year. These changes and upgrades will be reported in full at month 30 as part of Deliverable D4.4: Upgraded Platform. A summary of the changes for this period follows:

- Cloudbroker Azure, EGI and new CloudSigma regions adapters development

- CloudBroker Resource Catalogue and Resource Monitoring features development
- Ongoing bug fixes to the CloudBroker platform
- Backup procedure for the COLA server was established (nightly backups of user data as well as latest database state)
- CloudBroker platform technologies were updated.
- Backup procedure for the COLA server was established (nightly backups of user data as well as latest database state)
- Multiple upgrades to the adapters were performed to improve operation of the integrated clouds.
- SICS cloud (version 2) was integrated in the CloudBroker platform.
- Several images were generated to support performance testing.
- Multiple upgrades / updates to improve performance and functionality of the COLA CloudBroker platform.

9. Summary of the transition from Development Testbed to Production Infrastructure

Over the last year we have made efforts to align functionality and performance of the production infrastructure with outputs from other WPs to help us transition from development testbed to a fully optimised production infrastructure and to ensure optimal commercial exploitation and sustainability beyond the project end.

All cloud providers within the project (CloudSigma, Westminster, SZTAKI, SICS) have provided extensive technical support for their respective clouds in preparation for the final production-quality demonstrators. While the academic clouds are still available for testing, CloudSigma offers production infrastructure that meets the demands of real-world deployments along with the standard commercial requirements such as 24/7 technical support and a comprehensive SLA, which will come into force after commercialisation. Figure # shows what the production infrastructure will look like at this point.

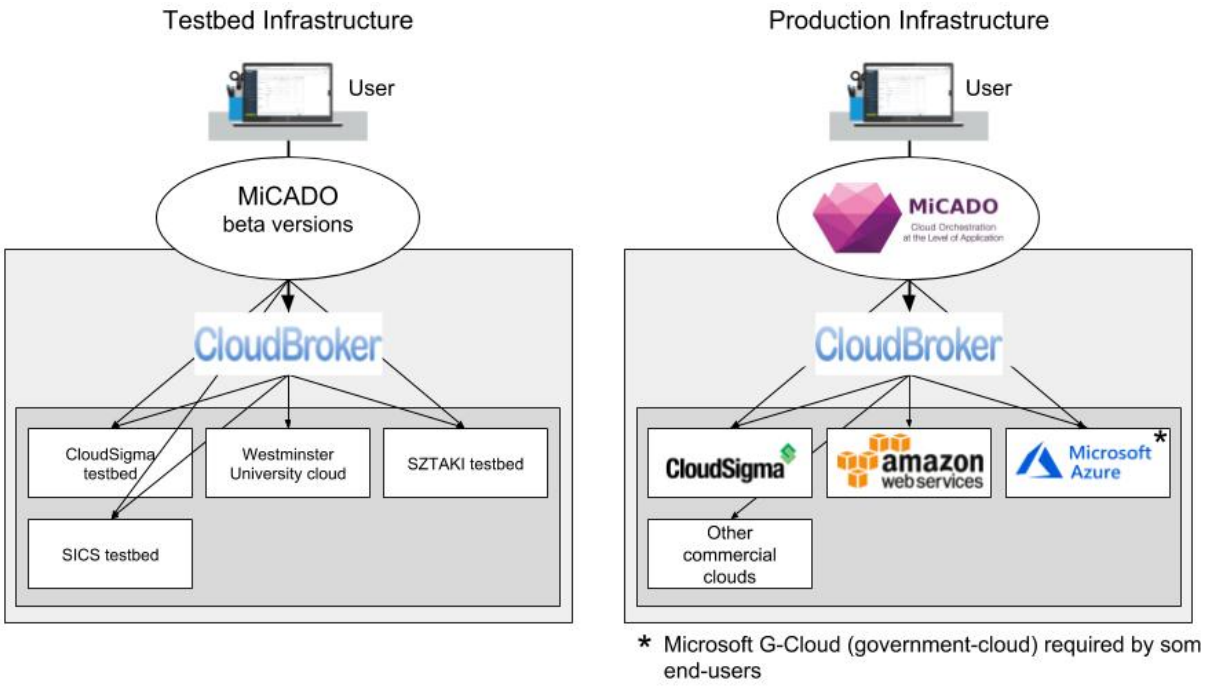

Figure 2: Testbed infrastructure, production infrastructure comparison

The operating system and software packages of the underlying CloudSigma production infrastructure were upgraded based on the performance testing and analysis performed by UoW and SZTAKI; presented in D4.2 and re-run for this deliverable (see Section 8.2).

The underlying hardware hosts operating system was upgraded to the latest Ubuntu LTS release at the time - 18.04. This allowed the upgrade of the kernel and the virtualization stack to the latest available versions. The benefits of this is an increase in performance, stability and security.

In addition to that, multiple new hardware hosts were introduced to the cloud. This allows for a greater elasticity of the production infrastructure and an improved performance. The servers that were introduced had the following general specifications: HPE Generation 10 servers with Intel Xeon Gold series CPU units.

To ensure interoperability, MiCADO is now able to connect to multiple cloud middleware/providers such as Amazon EC2, CloudSigma, OpenStack and OpenNebula (infrastructure layer), as well as the CloudBroker Platform (cloud access layer). CloudSigma and AWS are now fully integrated into the CloudBroker Platform. Integration of Microsoft Azure is planned to meet the needs of many organisations in the public sector that may require Microsoft G-Cloud (Government Cloud).

Due to the work undertaken in Work Package 4 and the interaction with other Work Packages, we now have a clear understanding of the infrastructure requirements, which will be used in the onboarding of additional commercial cloud providers, allowing for smooth integration into the CloudBroker Platform at any stage after the completion of the project. Microsoft Azure is one such provider.

10. Upcoming tasks in WP4

Over the final months of the project we will be concentrating on tasks leading towards exploitation and commercialisation of the results. All cloud providers will continue to maintain their cloud testbeds and to provide technical support to project partners. CloudSigma will continue provide access to their commercial cloud in Zurich. CloudSigma will also include both OCCOPUS and MiCADO into the Devops tools section of the CloudSigma web app. Finally, we will work towards submission of the final deliverable in Work Package 4, D4.4: Upgraded CloudBroker Platform.

11 Conclusion

In this deliverable we have covered the work done to maintain the testbed infrastructure and to prepare a fully functional and performance production infrastructure. More specifically, we outlined the improvements and upgrades to CloudSigma's commercial cloud in accordance with the requirements that arose during the project. Like the previous Work Package 4 delilverables, this one informs WP5 with regard to QoS and scaling services, WP6 to ensure price performance optimisation, and provides the developers of the core MiCADO components a baseline for further testing. As a result, application developers and end-users will be able to set QoS, security, performance and economic requirements, and make modification to the requirements on the fly. Furthermore, an optimized price/performance ratio will make the cloudification of applications more feasible for SMEs.

11 References

N/A