A software workbench for developing, deploying and controlling time-critical cloud applications

Paul Martin¹, Junchao Wang¹, Huan Zhou¹, Yang Hu¹, Arie Taal², Andrew Jones², Ian Taylor², Vlad Stankovski³, George Suciu³, Alexandre Ullisses³, Elisa Morón López³, Cees de Laat³, Zhiming Zhao³

¹ University of Amsterdam, Netherlands ² Cardiff University, UK ³ MOG Technologies SA, Portugal

Background

Time-critical applications are required to respond immediately to a range of events that may occur at runtime. Often the quality of service (QoS) given directly impacts business value (e.g. for multimedia platforms) or public safety (e.g. for disaster response). Many such applications are distributed and highly demanding. Cloud environments provide on-demand virtualised infrastructure that could support such applications, but there is a lack of tools for exerting fine-grained control over software-defined infrastructure and applications at runtime.

The software workbench for interactive, time-critical and highly self-adaptive cloud applications (SWITCH) is a Horizon 2020 project that will provide tools for managing the complete lifecycle of time-critical applications within the Cloud [1], explicitly linking user-level QoS with programmable infrastructure and autonomous runtime monitoring and control.

Architecture

The SWITCH workbench will provide tools for developing, deploying and controlling the execution of time-critical applications, supporting every stage of the application lifecycle. It will realise an application-infrastructure co-programming and control model that relates application logic, QoS constraints, and developments in programmable infrastructure.

The workbench has three subsystems:

* The SWITCH Interactive Development Environment (SIDE), to specify applications for deployment on Cloud.
* The Dynamic Real-time Infrastructure Planner (DRIP), to plan and provision applications on virtual infrastructure.
* The Autonomous System Adaptation Platform (ASAP), to monitor and interfere in the execution of applications.

The modularity of SWITCH allows components to be replaced as new Cloud standards and technologies come into existence.

Lifecycle

The SWITCH application lifecycle is split into a number of interlinked phases:

1. Application composition and verification.
2. Resource selection and infrastructure planning.
3. SLA negotiation.
4. Infrastructure provisioning.
5. Application deployment.
6. Application execution and runtime management.
7. Runtime monitoring and diagnosis.
8. Runtime adaptation.
9. Runtime visualisation and feedback.

Pilot cases

SWITCH has three pilot cases, each with different time-critical requirements and business case for moving into the Cloud:

* A collaborative business communication platform that provides browser-based real-time communication and collaboration that scales to demand while minimising costs.
* An elastic disaster early warning system that continuously monitors a sensor network, identifies events in progress, and upscales facilities in anticipation of user demands.
* A cloud studio for directing and broadcasting live events that manages the streaming of video feeds and the production of the broadcast stream virtually rather than on-site.

These cases all potentially potentially demonstrate the advantages of permitting developers access to software-defined infrastructure and networking in Clouds.

Information linking

Most of the information objects used by SWITCH can be realised using existing standards such as WS-BPEL, TOSCA, INDy and OCCI. However the innovations of SWITCH should be technology agnostic where possible. In order to provide a solution that is not intrinsically tied to specific standards, an abstract reference model for SWITCH is provided that can be linked with the conceptual vocabulary of those standards.

References


Acknowledgements

This research has received funding from the European Union’s Horizon 2020 research and innovation program under grant agreements 643963 (SWITCH project), 654182 (ENVRIplus project), and 676247 (VRE4EIC project).