Re-VAMP load testing with CLIF for continuous integration on the cloud

Bruno Dillenseger, Xavier Etchevers
Orange Labs

Beyond the automatic build of your application components, would it be possible to automate the instantiation of necessary virtual machines over the cloud, the installation of necessary middleware, the deployment and the configuration of components, for both the application under test and the load injection system?

If you liked the last year's talk about CLIF meeting Jenkins, you will enjoy their adventure with VAMP!
CLIF: OW2's load testing framework

- **generic/extensible**
  - IP, VoIP, database, mobile networks, custom protocols...

- **flexible**
  - Eclipse, Java Swing, command line, Maven, Jenkins

- **advanced**
  - Millions of virtual users
  - 1000+ distributed load injectors
  - continuous research transfer

- **mature**
  - more than 10 years feedback

http://clif.ow2.org
Load tests in continuous integration with the CLIF plug-in for Jenkins

- Automated testing
- Automated reporting
  - per-test report
  - trends through tests
- The CLIF plug-in
  - brings load testing and performance monitoring capabilities

http://jenkins-ci.org/
Pending issues for load testing in continuous integration

- **System Under Test**
  - Once built, automatically deploy and start the application for load testing
  - Including IP addresses settings for distributed applications

- **Load Injection**
  - Automatically deploy and configure load injectors
  - Automatically provide the load test scenarios with the target IP address

- **On which infrastructure?**
VAMP: Virtual Appliance Management Platform

→ VAMP deploys virtualized, distributed applications over the cloud

- pluggable IaaS drivers (e.g OpenStack)
- multi-IaaS support with the Sirocco driver
- generating and publishing virtual machines images through the UForge technology
- component-based description of the application's architecture
- advanced/autonomic management
  - self-repair, elasticity (to appear)
Deployment life-cycle with VAMP

**Images Provisioning**
- Packaging
- Publication in IaaS

**Installation**
- Transfert
- Intégration
- (Post-)configuration

**Generation**
- Virtual appliance generated

**Publication**
- Virtual images published

**Instantiation**
- Virtual machines instantiated

**Configuration**
- Applicative elements configured

**Activation**
- Application activated

Applicative elements (binaries, data, packages …)
VAMP's extensions to OVF

Image (software stack)
- **Id**
- **Format**
- **OS** *(id, distribution, version, arch.)*
- **Package** *(id, version)*
- **Data** *(id, type source, destination)*
- **State** defined | generating | generated | published

Virtual Machine
- **Id**
- **Cpu**
- **Memory**
- **NetworkItf** *(…)*
- **Disk** *(id, capacity, imageld)*
- **State** stopped | started | failed

Applicative Component
- **Name**
- **Interface** *(name, signature, role)*
- **Attribute** *(name, value)*
- **VirtualNode**
- **State** stopped | started | failed

Binding
- **Client**
- **Server**
- **State** defined | established
OVF graphical modeler for VAMP
Back to load testing in Continuous Integration

➔ Supporting tools
  ▪ Jenkins for automating tests and reports
  ▪ CLIF for load injection
  ▪ VAMP for deploying the test application and the load injection system over a IaaS

➔ Input
  ▪ tested application build chain
  ▪ VAMP-enhanced OVF description
    • application and load injectors combined
  ▪ CLIF test scenario
CLIF and the application under test in the OVF modeler
Overall process

**OVF {application + load injectors}**

**CLIF scenario (workload definition)**

**Continuous Integration server**

**VAMP**

**[optional]**

**Sirocco**

multi-IaaS support

**UForge**

VM image generation and publication

IaaS provider

virtual machines instantiation, components configuration and start

load injectors

application

5th November 2014
It's time for demo!
It's time for conclusion... and questions?

- Demonstration of load testing (CLIF) in continuous integration (Jenkins) deployed on cloud (VAMP)

- References
  - CLIF http://clif.ow2.org
  - Sirocco http://sirocco.ow2.org
  - VAMP svn://svn.forge.ow2.org/svnroot/sirocco/vamp
  - UForge http://www.usharesoft.com

- Developed in the context of the Open Cloudware PaaS project http://www.opencloudware.org
Very good question!

The VAMP manager creates and repairs Deployment Managers. It may be inside or outside the IaaS.

Configurator Agents self-configure the application. The applicative components are configured through dedicated wrappers.

The Deployment Manager (1 per application) creates applicative VMs and recreates them when they fail.

The VAMP manager

deployment manager

configurator agent

VAMP manager

VM0

deployment manager

configurator agent

VM1

configurator agent

VM2

Control Plane (VAMP)

Applicative Plane (legacy application)

Message Bus