Testing a VOIP user agent

Christos Tranoris
Electrical and Computing Engineering
University of Patras
Patras, Greece
http://nam.ece.upatras.gr
tranoris@ece.upatras.gr

I. SHORT USE CASE DESCRIPTION

A Federation Customer wants to test his own VOIP client application against operation conformance. Specifically the application's performance regarding:

- Sound and recording capabilities and conformance
- Dialtone correct operation and conformance

The customer wants a simple setup of an Asterisk server and 2 VOIP user agents configured to use this VOIP server. Then he will test his VOIP user agent with the Asterisk service and the configured VOIP user agents. The Sound and recording capabilities and conformance will be accomplished through and ECHO VOIP service and the dialtone correct operation and conformance against a DTMF VOIP service.

The customer's test suite consists of the following two test cases:

1) From the customer VOIP application a call to an extension number ie #301 will be made towards the Asterisk server in order then to be redirected to the ECHO VOIP service.
2) From the customer VOIP application a call to an extension number ie #401 will be made towards the Asterisk server in order then to be redirected to the DTMF VOIP service.

Additionally a third machine will participate that will keep any logging of all the services

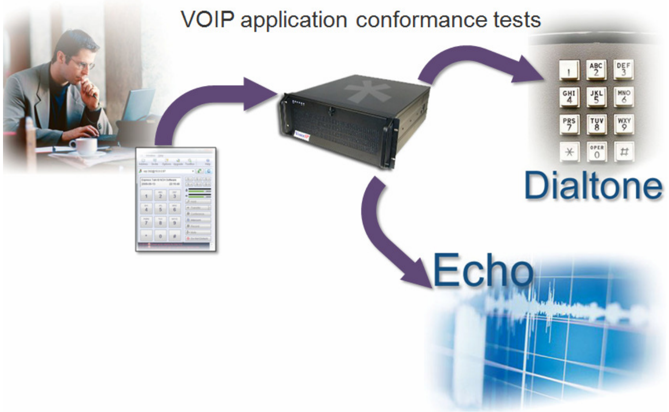


Figure 1: Testing the VOIP agent

Here is a brief list of customer's requirements:

Hardware	Machines capable of hosting the VOIP user agents
OS	Linux on all machines
Installed Services	Server: Asterisk 1.6.0 Clients: ECHO client based on Asterisk, DTMF client based on Asterisk
Availability	3 days
Target CPU Utilization	Server: 100%, Clients: -
Connectivity	-
QoS	Best effort

The requirements of the test exploit different aspects of the whole PII federation architecture:

1) From user design through provisioning details
2) A mixture of physical and Virtual resources
3) OS deployment
4) Services provisioning
5) Decisions that Teagle should make transparently from user
6) Core scenario steps from this use case could be used also in other scenarios

A. *Technical environment*

The tools to deploy, monitor and run the experiments are those offered by the Panlab architecture [1]. A Web Portal is available where customers and providers can access services, a visual Creation Environment which is called "Virtual Customer Testbed (VCT) tool" where a customer can define requested services, a repository which keeps all persistent information like resources, partners, defined federation scenarios, etc. Experimenters can browse through the resource registry content and can select, configure, deploy and access reserved resources. The above components interact with each other in order to offer a service called "Teagle".

Some additional components for integrating testbeds that belong to various administrative domains, are the following: the Panlab Testbed Manager (PTM) which is responsible for configuring the domain's resources. PTM implements the so called Resource Adaptation Layer where Panlab partners "plug-in" their Resource Adapters (RA). A Resource Adapter (a concept similar to device drivers) wraps a domain's resource API in order to create a homogeneous API defined by Panlab.

Figure 2 depicts the deployed testbed and its resources. For this scenario many resources of the whole federation participate, offered by the following testbed providers: University of Patras (Testbed A), EICT (logging/Customer) and Octopus (Testbed B).

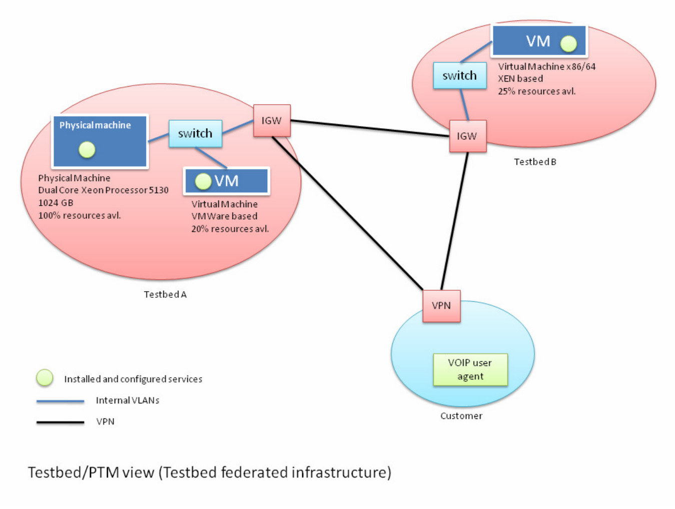


Figure 2: Deployed Testbed view

UoP Testbed consists of a solely physical machine, equipped with Two Intel(R) Xeon(R) E5320 Quad Core CPU @1.86GHz, for a total of 8 cores and 4Gb of memory. Such a hardware configuration allow us to instantiate easily up to 8 VMs without concurrency problems. The network physical configuration consists in 3 physical gigabit network adapters plugged on the same switch.

EICT has deployed a dedicated Panlab test environment to work on the development and to conduct functional and integration tests for the Panlab software components: a XEN server for hosting virtual machines and a laptop running a Windows based VoIP client for demonstration purposes. Furthermore, the EICT Panlab testbed was designed to implement a resource adapter for controlling Ethernet based switching fabrics. Thus, a dedicated switching device was added for testing and verification of the created code base.

Octopus network is a live telecom operator level (carrier grade) testing environment with its own service farm. The Octo testbed provides creation of virtual machines on a VMware vSphere host with a configurable Echo VoIP service. Octo provides the following Linux servers in order to provision virtual machines via Teagle.

- Linux server for PTM on GlassFish server and resource adaptation layer on Oscar OSGi
- IGW server
- OCTO service engine

Figure 3 displays the VCT design in the VCT tool. More details and specifications of Panlab's components can be found at [1].

B. *Results*

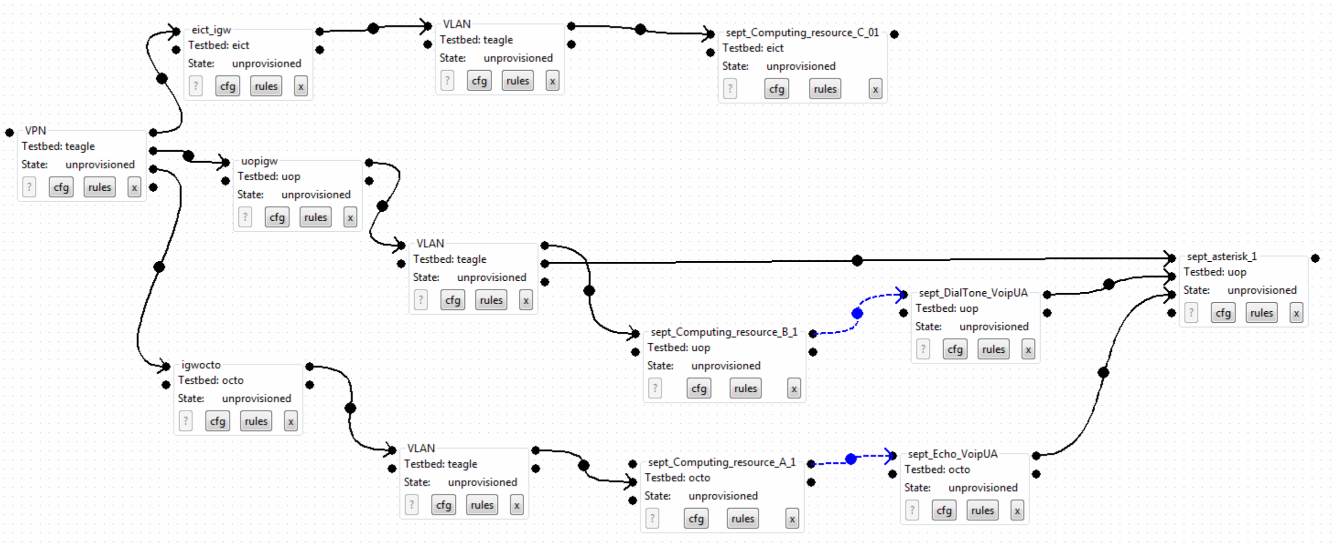
From the first fully use case implementation from design of the VCT to provisioning and running the test, some lessons learned where gathered:

- A further improvement of requirements and quality of the tools is needed
- All software tools need a deployable form
- Provisioning process could be improved from time perspective (i.e. parallelization of actions)
- Status of deployed components and RAs (alive, deployed, etc)
- Development of the reservation system
- A more sophisticated environment for constructing RAs is needed

II. TEST BED AVAILABILITY

The resources for creating similar scenarios are going to be available under the Panlab Office offerings. We expect to make more resources available as demand increases.

[1] Website of Panlab and PII European projects, supported by the European Commission in its both framework programmes FP6 (2001-2006) and FP7 (2007-2013): http://www.panlab.net



eict_igw
Testbed: eict
State: unprovisioned

VLAN
Testbed: teagle
State: unprovisioned

sept_Computing_resource_C_01
Testbed: eict
State: unprovisioned

VPN
Testbed: teagle
State: unprovisioned

uopigw
Testbed: uop
State: unprovisioned

VLAN
Testbed: teagle
State: unprovisioned

sept_asterisk_1
Testbed: uop
State: unprovisioned

sept_DialTone_VoipUA
Testbed: uop
State: unprovisioned

sept_Computing_resource_B_1
Testbed: uop
State: unprovisioned

igwocto
Testbed: octo
State: unprovisioned

VLAN
Testbed: teagle
State: unprovisioned

sept_Computing_resource_A_1
Testbed: octo
State: unprovisioned

sept_Echo_VoipUA
Testbed: octo
State: unprovisioned

Figure 3: The VCT design