Virtualization based testbeds widely used for the creation of network environments needed to test protocols and applications. However, complexity of present networks and protocols arises the need of very complex network testbeds, made of tens or hundreds of virtual machines.

Need for tools to support the design, deployment and management of large virtual network scenarios over clusters of servers.

Apart from large scale initiatives, several tools are available for small group or individual use:
- VNUML (www.dit.upm.es/vnuml)
- Netkit (http://wiki.netkit.org)
- MLN (http://min.sourceforge.net)
- Marionette (http://marionette.sourceforge.net)
- CORE (http://core.nrl.navy.mil/core)

However, none of them support neither individual use: tools are available for small group or educational environments, nor the diversity of virtual machines operating systems needed for complex testbeds.

Introduction

EDV: VNUML Distributed Deployment

In EDV project, a partnership between Telefonica I+D and UPM, a wrapper application to VNUML was developed to allow the distributed deployment of virtual scenarios over clusters of servers.

Distributed Deployment Architecture

EDV segments virtual scenarios into sub-scenes deployed to the different servers, interconnecting them by means of VLANs.

EDV includes basic segmentation algorithms (restrictions, round robin and weighted round robin) as well as an API to allow the user to develop new segmentation algorithms.

VNUML/EDV Limitations

Limitations of VNUML and EDV tools:
- Only Linux virtual machines (User Mode Linux limitation). Performance problems.
- Inability to manage virtual machines individually.
- Autocorification and command execution limited.
- Distributed version (EDV) limit: manual network configuration for dispense clusters, lack of monitoring tools, etc.

All these limitations led us to create:

VNX overall goal is the creation of a tool to allow the deployment of large virtual network scenarios over a federated cluster environment made of dispense nodes interconnected by means of layer 2/3 tunnels over Internet.

Each node is composed of:
- Virtualization servers running different types of hypervisors
- Physical (non-virtual) equipment to allow the creation of hybrid virtual scenarios

VNX Architecture

VNX overall goal is the creation of a tool to allow the deployment of large virtual network scenarios over a federated cluster environment made of dispense nodes interconnected by means of layer 2/3 tunnels over Internet.

Each node is composed of:
- Virtualization servers running different types of hypervisors
- Physical (non-virtual) equipment to allow the creation of hybrid virtual scenarios

VNX architecture: VNX

Plug-ins developed:
- libvirt, the Linux standard API for virtualization (libvirt.org), provides access to most of virtualization platforms supported in Linux (KVM, Xen, UML, etc.)
- Dynamips, to support emulated CISCO routers
- UML, which includes old VNUML code

The internal API is a simplified version of libvirt API, with the addition of a primitive to execute commands inside virtual machines.

VNX internal API

New autocofiguration and command execution mechanism created. Based on the OVF approach: a dynamic odfom to virtual machines with an XML file with autocofiguration parameters and commands to execute.

VNX Implementation

First version of VNX available:
- libvirt support. Tested with Linux (Ubuntu, Fedora, CentOS), FreeBSD and Windows (XP and 7).
- Dynamips and Olive router emulation support
- Virtual machine individual management (start, stop, restart, reboot, suspend, etc)
- OVF-like autocofiguration and command execution support
- Plug-in architecture to add extensions to VNX
- Distributed deployment support (EDV)
- Library of root filesystems available:
  - Ubuntu, Fedora, CentOS, FreeBSD, etc.
- VNX is mostly written in Perl (around 25000 lines of code): Windows autocof daemon written in C++.

Conclusions and Future Work

Future work includes:
- Full OVF support
- Dynamic scenarios (adding or deleting machines, network migration)
- Graphical user interface
- New virtual machine types (e.g. Android)
- Plug-in to control physical equipment
- Better network emulation capabilities
- Testbeds over the cloud: deploy virtual scenarios over cloud infrastructures

Acknowledgment

This research was partially supported by S21Sec and the Centra for the Development of Industrial Technology (CDTI) as part of the SEGURIG project (http://www.computogps.com), within the CENIT program, with reference CENIT-2007-2004, as well as by the Spanish Ministry of Science and Innovation (National Plan for Research, Development and Innovation) grant TEC2008-0659; as part of ARCO Project.

Contact Information

David Fernández
Dpto. Ingeniería de Sistemas Telemáticos
Universidad Politécnica de Madrid
Avda. Complutense, 30
28040 MADRID - SPAIN
E-mail: david@dit.upm.es
http://www.dit.upm.es/vnx