

Distem: a Software Defined Testbed

Cristian Ruiz, Emmanuel Jeanvoine and Lucas Nussbaum

Inria Nancy, MADYNES Team

September 14, 2016 – Journees Cloud



UNIVERSITÉ
DE LORRAINE



Evaluation of network algorithms or protocols

Two classical approaches for validation:

- **Formal**: equations, proofs, etc.
- **Experimental**, on a scientific instrument.

In reality

Given the size and complexity of today's systems, it is difficult to carry out complete analytic studies. **So, the experimental approach is commonly seen.**

Experimental validation is required

It is achieved using three different methodologies:

- ① Simulation
- ② Real execution
- ③ Emulation

Outline

- ① Distem
- ② Some studies performed with Distem
- ③ Enabling the evaluation of SDN based applications
- ④ Conclusions

An emulator for distributed systems

Take your **real application**

Run it on a **cluster**

And use **Distem** to **alter the platform**
so it **matches the experimental conditions you need**



Heterogeneous nodes
Long distance networks
Faults, perf. variations
Grid, Cloud, P2P features
...

Distem features

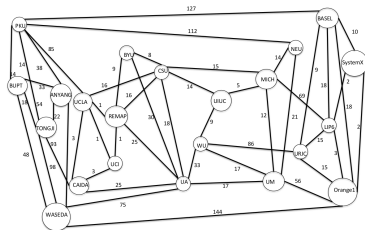
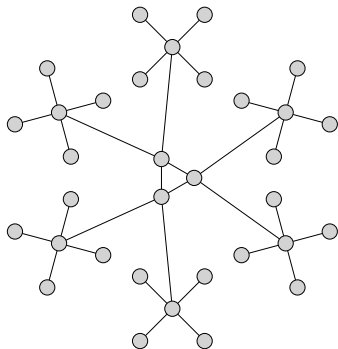
The features of Distem include:

- It achieves large scale virtual platforms:
 - e.g. 40.000 virtual nodes emulated over 168 physical nodes
- emulation of CPU performance, network topologies, I/O speed

Distem uses modern Linux functionality:

- Linux containers
- control groups
- CPU frequency scaling
- traffic control
- I/O throttling

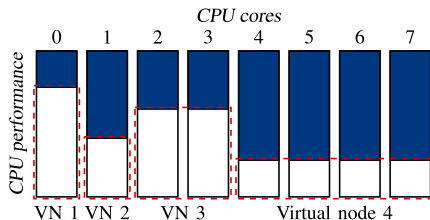
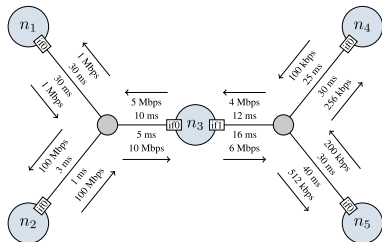
Complex network topologies



- We use VXLAN to create the links. L2 network are encapsulated into L3 networks preventing ARP storm in network equipment
- Management network is provided to communicate with each virtual node deployed
- We use Alevin¹ to map efficiently virtual topologies

¹<https://sourceforge.net/p/alevin/wiki/home/>

Event injection framework



- Heterogeneous conditions can be created and updated dynamically : CPU frequencies, different IO and network capabilities
- This is useful to achieve complex experiments where the platform is modified, like it could happened in reality
- These modifications can be injected using a deterministic behavior or using a probabilistic distribution

Failure injection framework

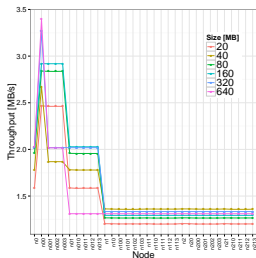
- We take into account failures that provoke a loss of the node (very common failures)
- Nodes can be lost in three different ways:
 - **Graceful**: the node is shut down cleanly, using an operating system command
 - **Soft**: the node is forced to shut down
 - **Hard**: the node failed abruptly
- We do not take into account byzantine failures

Outline

- ① Distem
- ② Some studies performed with Distem
- ③ Enabling the evaluation of SDN based applications
- ④ Conclusions

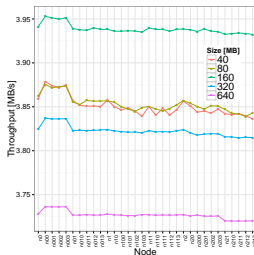
In networking caching of Information Centric Networks

Throughput for different nodes on a ICN network mapped over a dandelion topology (32 vnodes)



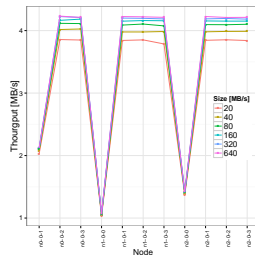
(a) All nodes get the file at the same time.

Cache is deactivated



(b) All nodes get the file at the same time.

Cache is activated



(c) Leaf nodes get the file in a sequential manner.

Cache is activated

HPC load balancer

Charm++ running Stencil3D using 128 processes in the heterogeneous platform

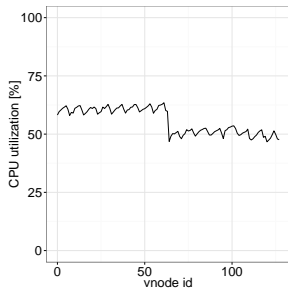


Figure: LBOff
Walltime: 341 secs

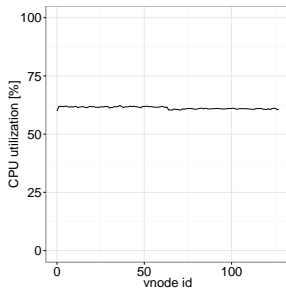


Figure: RefineLB
Walltime: 320 secs

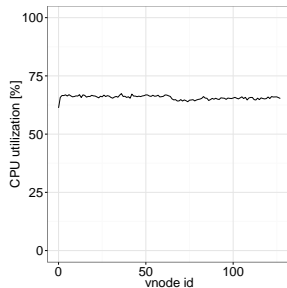


Figure: Hybrid
Walltime: 356 secs

Outline

- ① Distem
- ② Some studies performed with Distem
- ③ Enabling the evaluation of SDN based applications
- ④ Conclusions

Solves some limitations of legacy networks. SDN consist of four innovations:

- ① Separation of the control and data planes
- ② Centralization of the control plane
- ③ Programmability of the control plane
- ④ Standardization of application programming interfaces (API)

Challenges for evaluating SDN applications and configurations

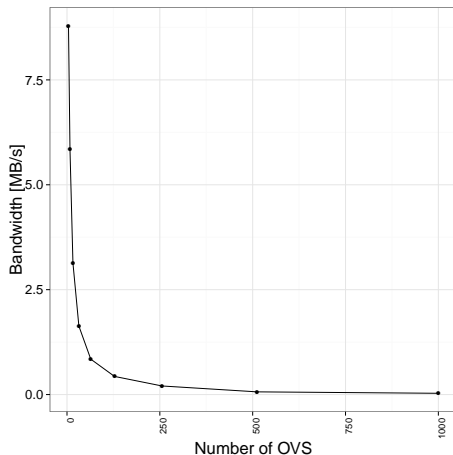
Motivation

Because the operation and intelligence of the network are fully controlled by an OpenFlow controller the correctness and efficiency of the functions implemented by the controller must be fully tested before its use in a production network.

Environments for developing SDN-based services and protocols

- Emulators: Mininet, Estinet
- Simulators: ns-3, fs-sdn
- Emulation of large data center networks
- Accuracy and efficiency
- Easily transition between prototype and test environment

Using Distem for evaluating POX controller performance



Linear chain of OpenvSwitches, 1000 virtual switches emulated using 16 physical machines

Outline

- ① Distem
- ② Some studies performed with Distem
- ③ Enabling the evaluation of SDN based applications
- ④ Conclusions

Conclusions

- It allows to build complex network topologies using a programmatic approach
- It allows to deploy the experiment on a desired number of physical resources transparently
- It offers a standard way to control the software stack through the use of tarballs that are widely used by LXC and Docker
- It provides an event injection framework which help experimenters to change platform characteristics in a reproducible way and to be integrated easily into scripts

Distem offers good facilities to test and prototype SDN-based application in real setting and under large topologies

Emulation of resources:

- Network parameters and topology
- CPU performance

Scalability and efficiency:

- Sharing of resources using virtualization
- Thousands of nodes can be deployed in a few minutes

To try it out

<http://distem.gforge.inria.fr>