Porting the Distem Emulator to the CloudLab and Chameleon testbeds

Cristian Ruiz, Emmanuel Jeanvoine, Lucas Nussbaum

Porting the Distem Emulator to the CloudLab and Chameleon testbeds

Cristian Ruiz, Emmanuel Jeanvoine, Lucas Nussbaum

RESEARCH REPORT

N° 8955

September 2016

Project-Team MADYNES

ISRN INRIA/RR--8955--FR+ENG

ISSN 0249-6399

Porting the Distem Emulator to the CloudLab and Chameleon testbeds

Cristian Ruiz[*†‡], Emmanuel Jeanvoine[*†‡], Lucas Nussbaum[*†‡]

Project-Team MADYNES

Research Report n° 8955 — September 2016 — 11 pages

Abstract:
The Distem emulator was designed in the context of the Grid'5000 testbed. In this paper, we describe the experience of porting Distem to two testbeds: CloudLab and Chameleon, in order to uncover possible issues when deploying Distem on platforms different from Grid'5000. It also provides some insight on differences between the design of each of those three testbeds, and their impact on experimenters.

Key-words: testbeds, distem emulator, porting

[*] Inria, Villers-lès-Nancy, F-54600, France
[†] Université de Lorraine, LORIA, F-54500, France
[‡] CNRS, LORIA - UMR 7503, F-54500, France

Portage de l'émulateur Distem sur les plates-formes expérimentales CloudLab et Chameleon

Résumé :
L'émulateur Distem a été conçu dans le contexte de la plate-forme Grid'5000. Dans cet article, nous présentons notre retour d'expérience sur le portage de l'émulateur Distem vers deux autres plates-formes expérimentales: CloudLab et Chameleon, dans le but de mettre en évidence des problèmes potentiels liés au fait d'utiliser une plate-forme autre que Grid'5000. Ce travail permet aussi de mettre en évidence des différences dans le design de ces trois plates-formes, et leur impact sur les expériences.

Mots-clés : testbeds, émulateur Distem, portage

1 Introduction

The Distem emulator was designed in the context of the Grid'5000 testbed, and some of its design choices are expected to be strongly tied to that testbed. In this paper, we describe the experience of porting Distem to two testbeds: CloudLab and Chameleon, in order to uncover possible issues when deploying Distem on platforms different from Grid'5000. Specifically, we tested the installation procedure, and whether the underlying infrastructure would impact the different features of Distem. We performed the following tests:

- One node installation;
- Installation on virtual and bare metal nodes;
- Multinode environment;
- *VXLAN* network between Distem vnodes.

No general background on Distem is provided in this paper. The reader is encouraged to read [3, 1].

In the next section, we review Distem's installation process. Then, in Section 3 we describe CloudLab and the results of our tests. Similarly, in Section 4 we will present Chameleon. Finally, we present our conclusions and lesson learned in Section 5.

2 Overview of Distem Setup

2.1 Distem installation

Distem can be installed in two ways: manually or using the `distem-bootstrap` script. These two methods are described below:

2.1.1 Steps to install distem by hand

This installation procedure at the moment is only valid in *Debian*. First, you should add the Distem repository to the file `/etc/apt/sources.list`

```
deb http://distem.gforge.inria.fr/deb ./
deb-src http://distem.gforge.inria.fr/deb ./
```

Then, build and install the Distem package:

```
apt-get update
apt-get build-dep -y --force-yes distem
apt-get install git rake

git clone https://gforge.inria.fr/git/distem/distem.git
cd $REPO_PATH/
rake snapshot
dpkg -i ../distem_1.2+git20160216145812_amd64.deb

apt-get install -q -y --force-yes -f
```

Before starting Distem, you have to make sure that you can log in into the nodes via SSH without password.

2.1.2 distem-bootstrap script

This script will launch the installation procedure in several machines in parallel and it will setup the whole infrastructure (daemon initialization). It needs only a list of machines that are accessible via SSH as `root`.

2.2 Automating tests

After initializing Distem on the nodes, you can use Distem commands (on the command line) to create the desired infrastructure. To automate the creation of virtual infrastructures, we write a script in ruby which uses the Distem Ruby library. This script in shown in Listing 1

```ruby
require 'distem'

nodelist = []
NB_VNODES = 1 # Number of vnodes per pnode
Distem.client do |cl|

  puts 'Creating virtual network'

  cl.vnetwork_create('vnetwork','192.168.0.0/22')

  puts 'Creating containers'

  count = 0

  pnodes = cl.pnodes_info

  private_key = IO.readlines('/root/.ssh/id_rsa').join
  public_key = IO.readlines('/root/.ssh/id_rsa.pub').join

  ssh_keys = {'private' => private_key,'public' => public_key}
  pnodes.each do |pnode|

   pnode_list = []

    NB_VNODES.times do
     nodename = "vnode-#{count}"
     pnode_list.push(nodename)
     count += 1
    end

    res = cl.vnodes_create(pnode_list,
     {'host' => pnode[0],
      'vfilesystem' =>{'image' => 'file:///root/jessie-mpich-lxc.tar.gz','shared' => true},
      'vifaces' => [{'name' => 'if0', 'vnetwork' => 'vnetwork'}]},
     ssh_keys)

    nodelist+=pnode_list
  end

  puts 'Starting containers'
  cl.vnodes_start(nodelist)

  puts 'Waiting for containers to be accessible'
  start_time = Time.now

  cl.wait_vnodes()

  puts "Initialization of containers took #{(Time.now-start_time).to_f}"

end
```

Listing 1: Ruby script for testing Distem

3 Tests on the CloudLab testbed

CloudLab is a *meta-cloud*, that is a facility for building clouds. Currently the platform is distributed across three sites at the university of Wisconsin, Clemson University, and the university of Utah. It gathers approximately 5000 cores with different architecture: ARM and x64. More detailed information about hardware characteristics can be found at `https://www.cloudlab.us/hardware.php`. Users have access to two types of environments:

- Virtual machiens using the XEN hypervisor
- Physical machines (x86_64, ARM)

3.1 Starting a new profile

The first step in *CloudLab* assuming you have already created an account and joined a project is to start an experiment. You will be redirected to a web page where you can select a profile that fits your needs and start from it. There are already a lot of profiles to start from, all of them offering a wide range of software already configured and different topologies (number of nodes and network links). For testing Distem, we start from a simple profile which includes just one machine with an *Ubuntu* distribution already installed.

Once the profile has been selected, we proceed to choosing a site to instantiate the experiment on: Utah, Clemson or Wisconsin. Only the sites that are compatible with the chosen profile are shown.

You will be able to see the progress of initialization of your experiment from the web interface which has been nicely designed. In this web interface you have access to a web-based console for interacting with your nodes once the deployment has finished. Additionally, in this same web interface, it is indicated how you can access your machines via SSH. For bare metal deployments, you are provided with different public IP addresses that you can access from anywhere (e.g., your local machine). In the case of Xen based deployments a combination of public IP address and ports are given. Normally, you will use the same IP address and change the port in order to access the desired machine.

In the first experiment, we used just one machine in order to install Distem. We perform the installation of Distem manually as indicated in Section 2. Then, we start Distem manually by typing:

```
distemd -d &> /tmp/distem.log& # launching daemon
distem --coordinator host=c220g2-011313.wisc.cloudlab.us --init-pnode c220g2-011313.wisc.cloudlab.us
```

Or using the IP address of the machine:

```
distem --coordinator host=128.104.222.133 --init-pnode 128.104.222.133
```

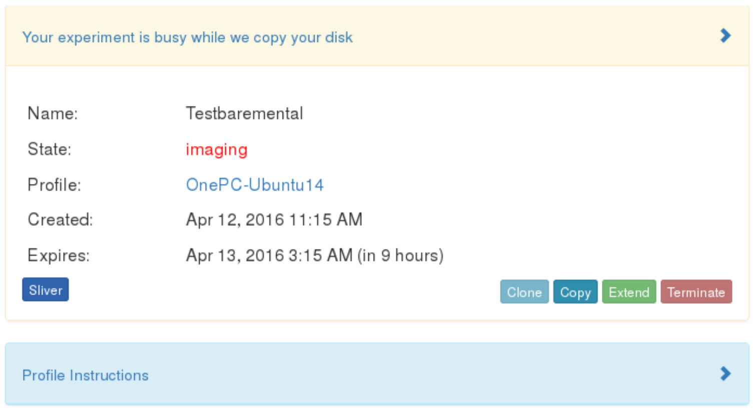
We deploy a *vnode* for testing using the following commands

```
distem --create-vnetwork vnetwork=vnetwork,address=10.144.0.0/22
distem --create-vnode vnode=node-1,pnode=172.17.14.2,rootfs=file:///root/jessie-mpich-lxc.tar.gz,\
sshprivkey=/root/.ssh/id_rsa,sshpubkey=/root/.ssh/id_rsa.pub
distem --create-viface vnode=node-1,iface=if0,vnetwork=vnetwork
distem --start-vnode node-1
```

This procedure could be automated using the script `distem-bootstrap` for bare metal based deployments:

```
bundle exec ruby examples/distem-bootstrap --distem-version wheezy -g -f /tmp/machine_file
```

After making sure that Distem installation works without problem, we save the state of our experiment as a profile. For that, we click on the option *clone* which will create a new profile in our account. This processes will snapshot the disk contents creating a new image that will be associated to our profile. A screenshot of the process is shown in Figure 1.



Your experiment is busy while we copy your disk

Name:	Testbaremental
State:	imaging
Profile:	OnePC-Ubuntu14
Created:	Apr 12, 2016 11:15 AM
Expires:	Apr 13, 2016 3:15 AM (in 9 hours)

Figure 1: Disk snapshotting

This normally takes several minutes and at the end of the process we will be notified via email. If we look at our profiles, there should be a new available profile.

3.2 Multi-node profile

We edited the created profile in order to add a new node with the same characteristic. We used a web-based profile editor which is shown in Figure 2. Using this profile editor we were able to produce different multi-node profiles to test Distem, they are summarized in Section 3.4. For starting Distem in a multi-node environment manually, we do:

```
distem --coordinator host=172.17.4.19 --init-pnode 172.17.4.19,172.17.9.9
```

For bare metal deployments, we could automate using `distem-bootstrap`:

```
bundle exec ruby examples/distem-bootstrap --distem-version wheezy -g -f /tmp/machine_file
```

For Xen-based deployments, it was not possible for containers located in different machines to communicate, as detailed in Section 3.5. Therefore, we have to add a dedicated network link. This created a new network interface in the machine, the network interface is configured in a private network which can

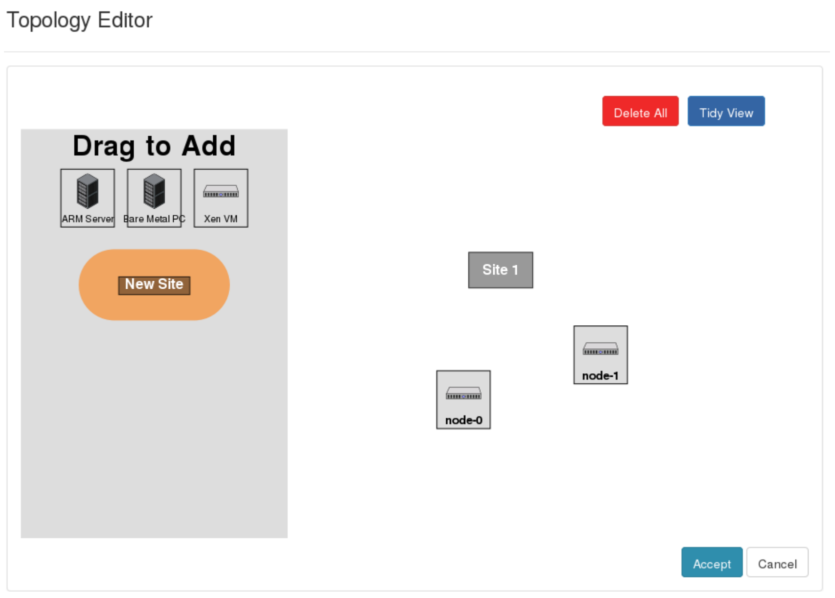


Figure 2: Profile editor

be specified on the profile. We should now use this new interface when deploying the virtual infrastructure. To do that we made the following modifications to the script in Listing 1:

```
cl.vnetwork_create('vnetwork','192.168.0.0/22', :root_interface => "eth1")
```

3.3 Testing VXLAN

In order to activate *VXLAN* networking support[2] on Distem, we changed the following line in the script that build the virtual infrastructure:

```
cl.vnetwork_create('vnetwork','192.168.0.0/22',:network_type => 'vxlan')
```

Before deploying the virtual infrastructure, we need to update the version of the `iproute2` package provided by Ubuntu 14.04. More details on this procedure are available in Section 3.5. The deployment of *VXLAN* networks did not solve the communication problems we had for the multinode deployment using Xen VMs with no additional network link. For all the other cases, *VXLAN* networks worked without any problem.

3.4 Profiles created

1. `distem-demo` : Two virtual machines running Ubuntu14.04 and configured with a network link, for the Xen hypervisor;

2. `distem-arm` : Two bare metal machines running Ubuntu 15.10 configured with a network link, for the ARM architecture;

3. `distem-x64` : Two bare metal machines running Ubuntu14.04, for the x64 architecture.

These profiles are publicly accessible from CloudLab.

3.5 Problems encountered

This section summarizes the problems encountered during the deployment of Distem on CloudLab.

3.5.1 Compatibility with Ubuntu 14.04

We found two problems when deploying Distem on Ubuntu 14.04 which consequently were fixed:

1. Incompatibility with version 1.0.8 of *LXC*;
2. Mechanism used to detect that the *cgroups* file system is mounted.

Another problem was the non-initialization of the SSH service inside containers. Therefore, we should start the service by hand using the following command to log in into the container:

```
lxc-attach -t $VNODE_NAME
```

This problem is probably linked to the version of *LXC* used, more investigation is needed.

3.5.2 Network problems

1. When using Xen infrastructure, it seems that the default virtual network card cannot be configured in promiscuous mode. Therefore, packets that are not addressed to the VM MAC address are dropped. This prevents the communication of containers located in different virtual machines. To solve this problem, we add another network interface.
2. The fact that VMs are accessed using a combination of IP address and port breaks how we handle results using the library Net::SSH::Multi [1]. The hostname is used as a key in the Hash used for storing the results.
3. We found an inconsistency on how hostnames are assigned. The execution of the two following commands returns two different values:

```
root@node-0:~# host 128.104.222.180
180.222.104.128.in-addr.arpa domain name pointer c220g2-011128.wisc.cloudlab.us.
```

```
root@node-0:~# hostname
node-0.distem.disco-cloudlab-pg0.wisc.cloudlab.us
```

   This breaks Distem when we want to clean up our environment executing the command `distem -q`.

[1] `http://www.rubydoc.info/github/ruby-cute/ruby-cute/master/Net/SSH/Multi`

4. By default, we do not have access to the machines using the `root` user and this is necessary for Distem. This can be easily achieved by typing:

```
cat .ssh/id_rsa.pub >> .ssh/authorized_keys
```

3.5.3 *iproute2* version

For setting up *VXLAN* networking in Distem, a *UDP* port is configured using the `dstport` flag. This flag will only run on version 3.16+ of `iproute2`. This version of `iproute2` was not available in Ubuntu 14.04 which normally ships version 3.13. We update the version of the package to 3.16+ using the following commands:

```
wget http://launchpadlibrarian.net/212469986/iproute2_3.16.0-2ubuntu1.1_amd64.deb
dpkg -i iproute2_3.16.0-2ubuntu1.1_amd64.deb
```

4 Tests on Chameleon testbed

Chameleon is a configurable experimental environment for large-scale cloud research. It provides two kinds of platforms to perform experiments: bare metal and virtual machines managed by OpenStack. More information about the hardware description can be found here[2]

4.1 Deployment using OpenStack and KVM

One way of deploying experiments is to use OpenStack. Complete instructions for deploying an instance using the OpenStack interface can be found in the Chameleon User Guide[3]. You can choose the number of instances (VMs) to deploy. The first step is to deploy just one instance to test Distem. After deploying it, you have to setup several parameters before actually being able to login in. You have to enable a public IP address and then add a rule for enabling SSH connections. A default network is added. Then, we can access our instance via SSH using the public configured IP address. The username *cc* must be used:

```
ssh -i cloud.key cc@129.114.110.226
```

Distem worked successfully on one node, following the same procedure as the one used in Section 2 and Section 3. We snapshot the disk state and we create a new image. This new image is accessible for post deployments.

4.1.1 Multi-node setup

We used the disk image generated previously for instantiating two virtual machines. For that, we click on *launch instance* and then we choose the flavor *small* which is configured with a disk of 20 GB. We put the instance count to 2. Using the image created, we were able to successfully deploy two nodes. One of the

[2]`https://www.chameleoncloud.org/about/hardware-description/`
[3]`https://www.chameleoncloud.org/docs/user-guides/openstack-kvm-user-guide/`

reserved nodes has to be used as a gateway to access the other nodes. Therefore, `distem-bootstrap` cannot be used transparently, unless we use another machine just to execute this script.

The disk image will contain the SSH keys and the file `authorized_keys`, so we should probably erase old entries in the file `~/.ssh/known_hosts` in order to avoid a fingerprint mismatch. After initializing Distem using the same procedure in Section 3 and deploying a virtual infrastructure, the *vnodes* are not able to communicate. This resembles the situation in CloudLab when using the Xen hypervisor. However, there is no possibility of adding another network link as in CloudLab.

4.2 Deployment using Bare metal

All the steps are well described here[4]. We have to correctly configure our timezone, in order to successfully perform a reservation. After making the reservation, we instantiate a physical machine. All the process takes place in an OpenStack-like web interface.

The process of creating a snapshot of the disk is rather complicated given that it is not supported natively by the Ironic bare metal provisioner. You have to execute several commands before being able to create the snapshot. Here [5] you can find the procedure for doing that on *Ubuntu* 14.04.

4.2.1 Multinode

We deployed successfully Distem on a multinode environment, and containers located in different machines were able to communicate.

4.3 VXLAN test

We tried to setup a *VXLAN* but we run into the same problem of `iproute2` version as described in Section 3.5. We update the package as already described in Section 3.5.3 and we modify the following line in the script (Listing 1), in order to deploy with *VXLAN*.

```
cl.vnetwork_create('vnetwork','10.144.0.0/22',:network_type => 'vxlan')
```

The use of *VXLAN* enables the communication of containers located in different machines. This works for all types of configurations tested.

4.4 Problems encountered

1. It was impossible to bring networking to containers located in different VM using OpenStack KVM. This was solved partially by using *VXLAN*.

2. Up-to-date Ubuntu images were not available, and we could not upgrade to another version.

[4] `https://www.chameleoncloud.org/docs/bare-metal-user-guide/`

[5] `https://www.chameleoncloud.org/docs/user-guides/ironic/%23%snapshotting_an_instance`

5 Conclusions

These series of tests were a useful use case for providing an idea of how the chosen experimental testbeds work and the different implications when performing experiments. We found that CloudLab offers the best user experience through a nicely conceived graphical interface. This helps users to easily deploy and configure a complex experiment. Additionally, CloudLab encourages reproducibility of experiments by providing a system of templates to describe an experiment. The templates can be shared and created without any additional cost for users.

Chameleon manages the deployment of an experiment using an OpenStack-like interface, which is expected as a large share of the Chameleon software stack comes from the OpenStack project. However, this currently limits the number of additional facilities specially suited to experimentation. Also, for a user not familiarized already with OpenStack, the deployment of experiments could seems complex at the beginning. There still exists some problems and features that are not supported such as hard disk snapshoting on baremetal machines and options for configuring and managing multiple network links.

One difference between both testbeds and Grid'5000 is the lack of access points or gateways from where all the machines of the tested are reachable. In the case of CloudLab, machines are accessible in different ways using ports or public ips which depend on the type of environment (virtual machines or baremetal). This could limit the size of experiments due to the availability of public ips. Additionally, we found that even if Grid'5000 seems less friendly for a newcomer than the other testbeds presented in this report, it has a large level of configurability. It offers several possibilities to configure the hardware to fit an experimenter's needs. Finally, after having the experience of deploying our emulator Distem in three different testbeds: Grid'5000, CloudLab and Chameleon. We observed clearly the trade-off between user experience and the level of control we have in the hardware.

6 Acknowledgments

We want to thank the *CloudLab* and *Chameleon* teams for providing us with access to their platforms.

References

[1] T. Buchert, L. Nussbaum, and J. Gustedt. Methods for Emulation of Multi-core CPU Performance. In *HPCC-2011*, September 2011.

[2] Tomasz Buchert, Emmanuel Jeanvoine, and Lucas Nussbaum. Emulation at Very Large Scale with Distem. In *SCALE Challenge, held in conjunction with CCGRID'2014*, 2014.

[3] Luc Sarzyniec, Tomasz Buchert, Emmanuel Jeanvoine, and Lucas Nussbaum. Design and Evaluation of a Virtual Experimental Environment for Distributed Systems. In *PDP2013*, Belfast, Royaume-Uni, February 2013.

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399