On the Performance of Linux Container with Netmap/VALE for Networks Virtualization

Maurizio Casoni, Carlo Augusto Grazia, **Natale Patriciello**

Department of Engineering *Enzo Ferrari*
University of Modena and Reggio Emilia

Modena, 15 October 2013

Talk overview

1. Introduction
   - Problem
   - State of the Art
2. Architecture
   - Linux Virtual Ethernet
   - LXC with Netmap and VALE
3. Environment
4. Results
   - Test Environment
   - Incremental loss-free throughput
   - Maximum serial loss-free throughput
   - Maximum parallel loss-free throughput
   - Maximum sender load
5. Netmap/VALE with ns-3
6. Conclusions
7. Future Works

Problem

what

Performance comparison between interconnection technologies on Virtual Machines.

- No care (for now) of what is currently inside V.M.
- Linux focused (FreeBSD maybe.. but not Windows!)

why

Improve NetBoxes network emulation

- NetBox: a solution for emulating complex networks based on Linux Container
- Can be configured to talk over plain Ethernet protocol with other self instances or real-world equipment
- Interconnecting NetBoxes plays a crucial role on its scalability

State of the Art

typical solution

use of Virtual Ethernet (veth) and Virtual Bridges through ordinary system calls

weaknesses

- big per-packet overhead
- dynamic memory allocation
- memory copies between user and kernel space
- consequence: low throughput (near 500 Kpps with 64 byte per packet)

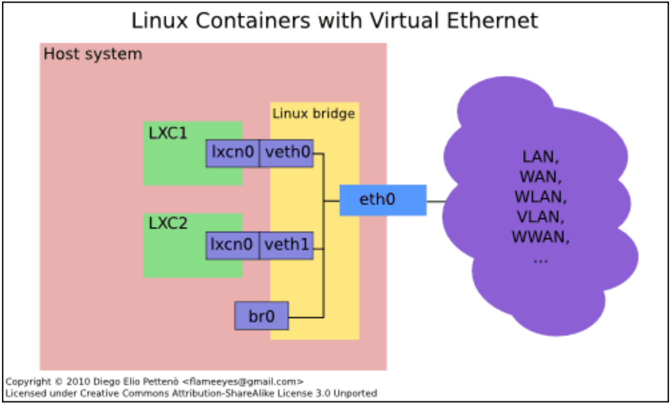
LXC with Virtual Ethernet

Virtual Ethernet

Software object that can process outgoing and incoming packets

- simple tunnel driver that works at the link layer
- looks like a pair of ethernet devices interconnected with each other
- require **bridges** to communicate with other virtual or real interfaces
- managed through standard **ip** tool

LXC with Virtual Ethernet



LXC

LXC

lightweight virtual system mechanism

- builds from chroot to implement complete virtual systems
- resource management
- isolation mechanisms

Reverse paradigm

Instead of retrofitting efficiency onto full isolation, LXC started out with an efficient mechanism (existing Linux process management) and added isolation

Netmap

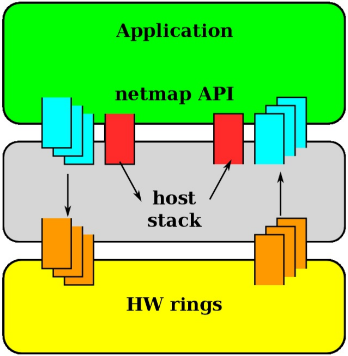
Netmap

A novel framework for fast packet I/O

- memory map preallocated buffers
- software buffers matches NIC ring buffers
- I/O batching

Changes over API

Little changes to fix weakness of standard API (but programs should be adapted)

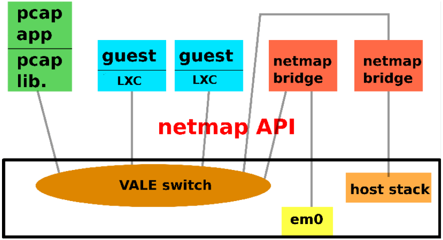


VALE

VALE

Software Virtual Local Ethernet whose ports are accessible using the netmap API

- designed to interconnect virtual machines
- can be used as a fast local bus
- works as learning bridge
- accessible through Netmap API



System model

Nodes

- sender: sends packets over its output interface
- forwarder: receive packets from its input interface and forward them over its output interface
- receiver: count how many packets it receives over its input interface

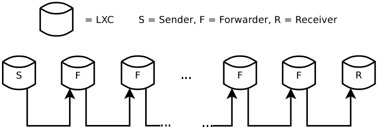
System model

Serial setup

representative of many network scenarios EU 7th FP Large Scale Integrated Project "A holistic approach towards the development of the first responder of the future"

What is it?

- chain of forwarders
- first and last forwarders connected to a sender and a receiver



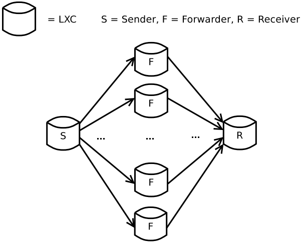
System model

Parallel setup

models another interaction from the same EU 7th FP Large Scale Integrated Project

What is it?

- forwarders works in parallel
- only one sender and one receiver



Test Environment

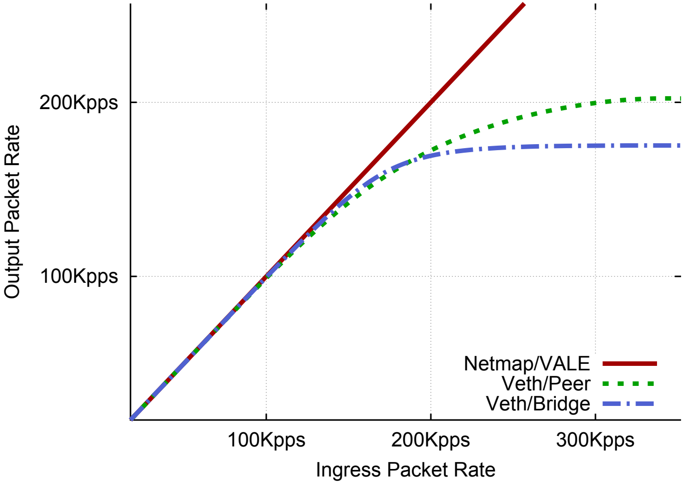
- UNIX-based open tool, running on LXC instances
- backend supported:
  - Netmap / VALE
  - Virtual Ethernet as peer
  - Virtual Ethernet with Virtual Bridges
- small footprint:
  - no correctness checks
  - packets are allocated and constructed before any measurement
  - measurement made by reading Time Stamp Counter

Incremental loss-free throughput

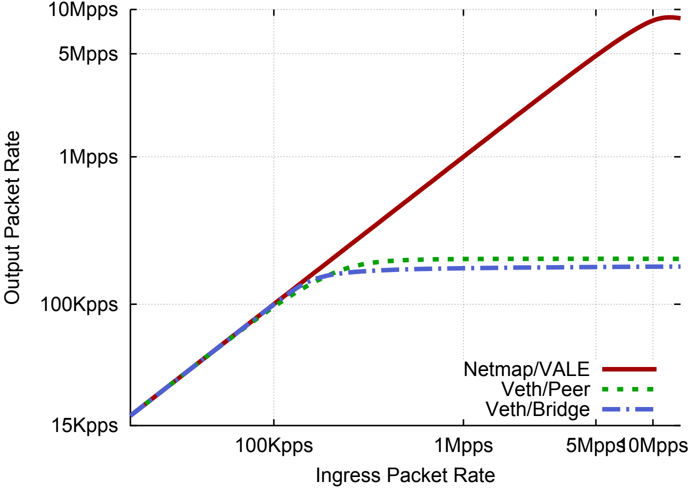
Experiment setup

- nodes placed in series
- slowly increase of sending rate
- measuring receiver rate
- where is the saturation point?

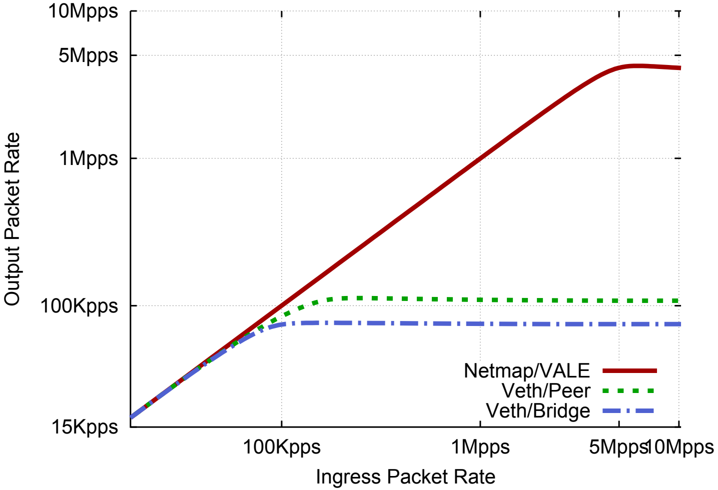
Incremental loss-free throughput - 1 forwarder



Incremental loss-free throughput - 1 forwarder



Incremental loss-free throughput - 6 forwarders

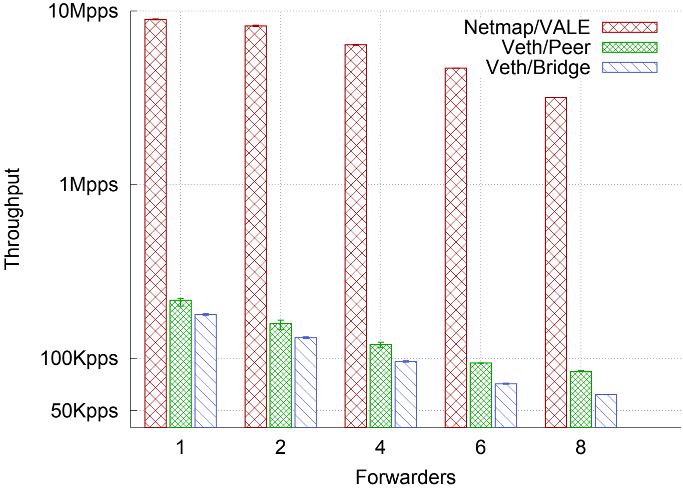


Maximum serial loss-free throughput

Experiment setup

- serial setup
- from 1 to 8 forwarders
- maximum sending rate, without loss at receiver side

Maximum serial loss-free throughput

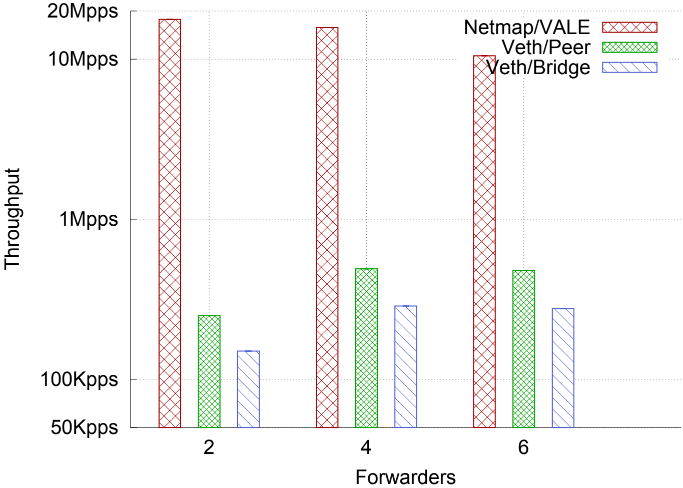


Maximum parallel loss-free throughput

Experiment setup

- parallel setup
- from 2 to 4 forwarders
- measuring receiver rate

Maximum parallel loss-free throughput

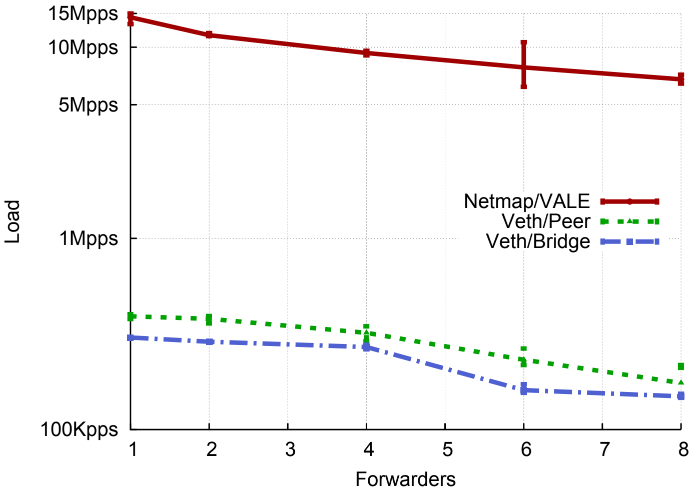


Maximum sender load

Experiment setup

- serial setup
- from 1 to 8 forwarders
- sending rate measurement

Maximum sender load



What is ns-3 ?

ns-3

a discrete-event network simulator, targeted primarily for research and educational use

Why ns-3?

- preferred, open simulation environment for our networking research
- it can be used as realtime network emulator
- wireless or wired IP simulations which involve models for Wi-Fi, WiMAX, or LTE for layers 1 and 2 and and a variety of static or dynamic routing protocols such as OLSR and AODV for IP-based applications

Adding Netmap/VALE support to ns-3

Improve ns-3 real-world interaction

10 Gbps (64 byte per-packet) at software layer with commodity hardware !

Improve ns-3 emulation range

VALE supports up to 70 Gbps of throughput between its ports

Not all comes for free..

- **API change:** we need to use Netmap API inside ns-3 code
- **Paradigm change:** adapt ns-3 code to work with I/O batching and preallocated buffers

Preliminary results

Emulation range: before..

- 70 Kpps (64 byte per-packet) between two ns-3 instances

Emulation range: after

- 2.5 Mpps (64 byte per-packet) between two ns-3 instances

Some comments

- Our focus is in emulation range, real-world interaction will follow for free after proper coding of emulation's code
- Packet processing code is quite expensive in ns-3, even at layer 2 (after all is a simulator)
- Upper layers not aware of I/O batching capabilities

Conclusions

Experiment

- we measured performance with a Netmap/VALE and a Veth backend
- beta version of Netmap/VALE in ns-3 (should be code-reviewed by the community)

Netmap/VALE

it improves performance

- less overhead
- more throughput
- we can now emulate a wide range of networks
- **sidenote:** we should change applications not only at code level, but also as paradigm

Future Works

- Netmap/VALE with ns-3 (partially done!)
- benefits for NetBoxes interconnected with physical devices
- layer 3 and layer 4 modification (working with packet bursts)

thank you
for the attention