Empowered by Innovation

Towards the Superfluid (Network) Cloud

Felipe Huici
[felipe.huici@neclab.eu]

NEC Laboratories Europe

Motivation

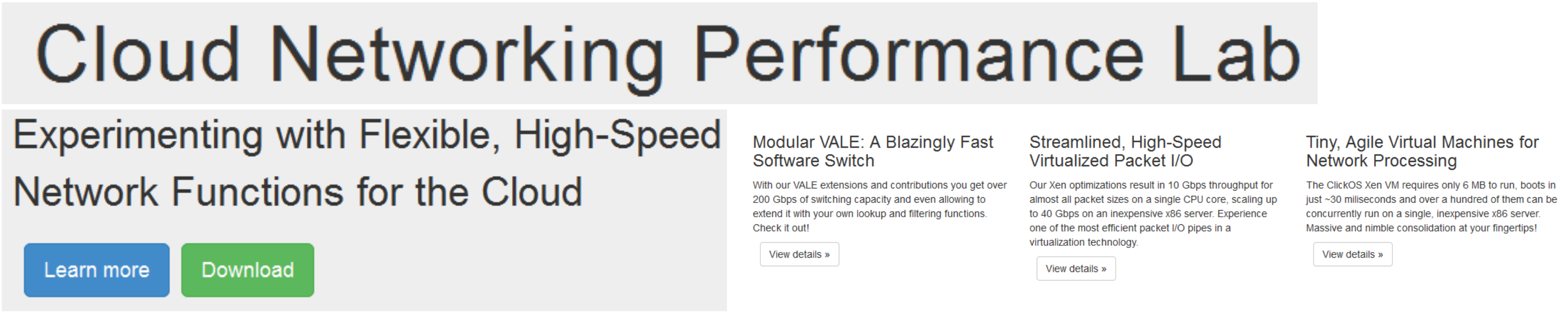
- Virtualization and cloud deployments have brought great benefits
  - OPEX/CAPEX reduction (fewer servers, lower cooling and power costs)
  - Faster deployment
  - Better disaster recovery
  - Flexibility through migration
  - Isolation, multi-tenancy

- Can we improve things further, making the cloud more “fluid”?
  - High consolidation (Hundreds? Thousands of VMs?)
  - On-the-fly service instantiation (in milliseconds)
  - Fast migration (hundreds of milliseconds?)
  - High throughput (10-40+ Gb/s)

Talk Overview

- Novel technologies and optimizations
  1. ClickOS: High performance NFV
  2. Minicache: Virtualized content caches
  3. VALE: High performance, modular, energy efficient SW switch
  4. Massive consolidation: thousands of VMs on a single server

- Check out our open source portal!
  - http://cnp.neclab.eu/



1. ClickOS: High Performance NFV*

ClickOS and the Art of Network Function Virtualization
NSDI 2014

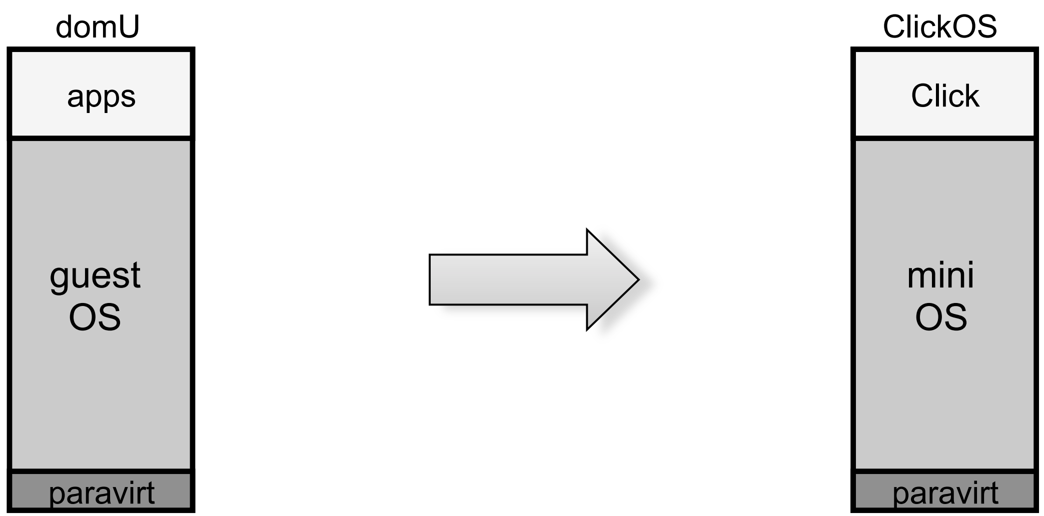
NFV: Shifting Middlebox Processing to Software

- Can share the same hardware across multiple users/tenants
- Reduced equipment/power costs through consolidation
- Safe to try new features on a operational network/platform
- But can it be built using commodity hardware while still achieving high performance?
- ClickOS: tiny Xen-based virtual machine that runs the Click modular router software

From Thought to Reality - Requirements

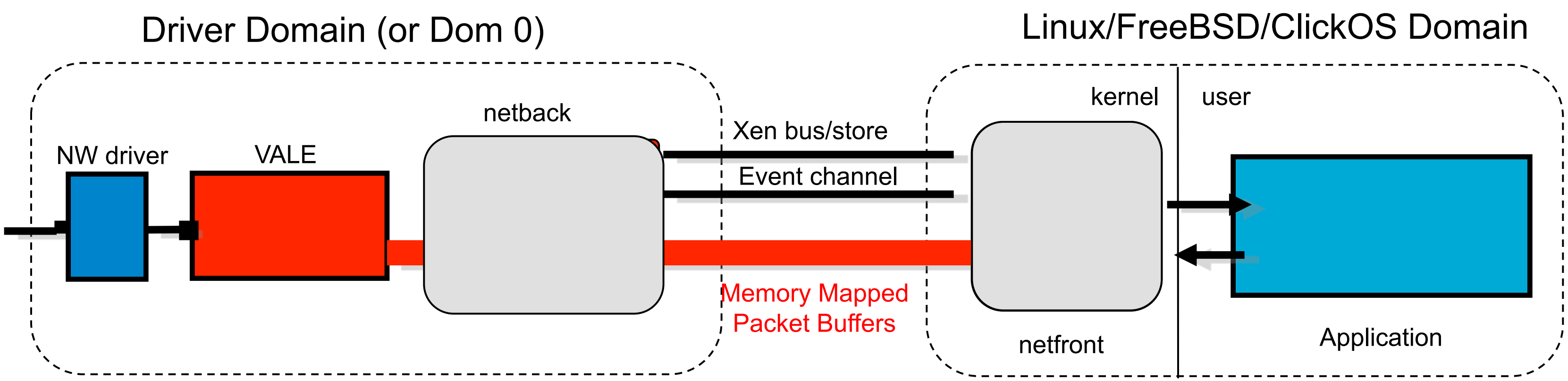
	ClickOS
Fast Instantiation	✔ < 20 msec boot times
Small footprint	✔ 5MB when running
Isolation	✔ provided by Xen
Performance	✔ 10Gb/s line rate* 45 µsec delay
Flexibility	✔ provided by Click

What's ClickOS?



- **Work consisted of:**
  - Build system to create ClickOS images
  - Emulating a Click control plane over MiniOS/Xen
  - Reducing boot times
  - Optimizations to the data plane
  - Implementation of a wide range of middleboxes

Data Plane Optimizations



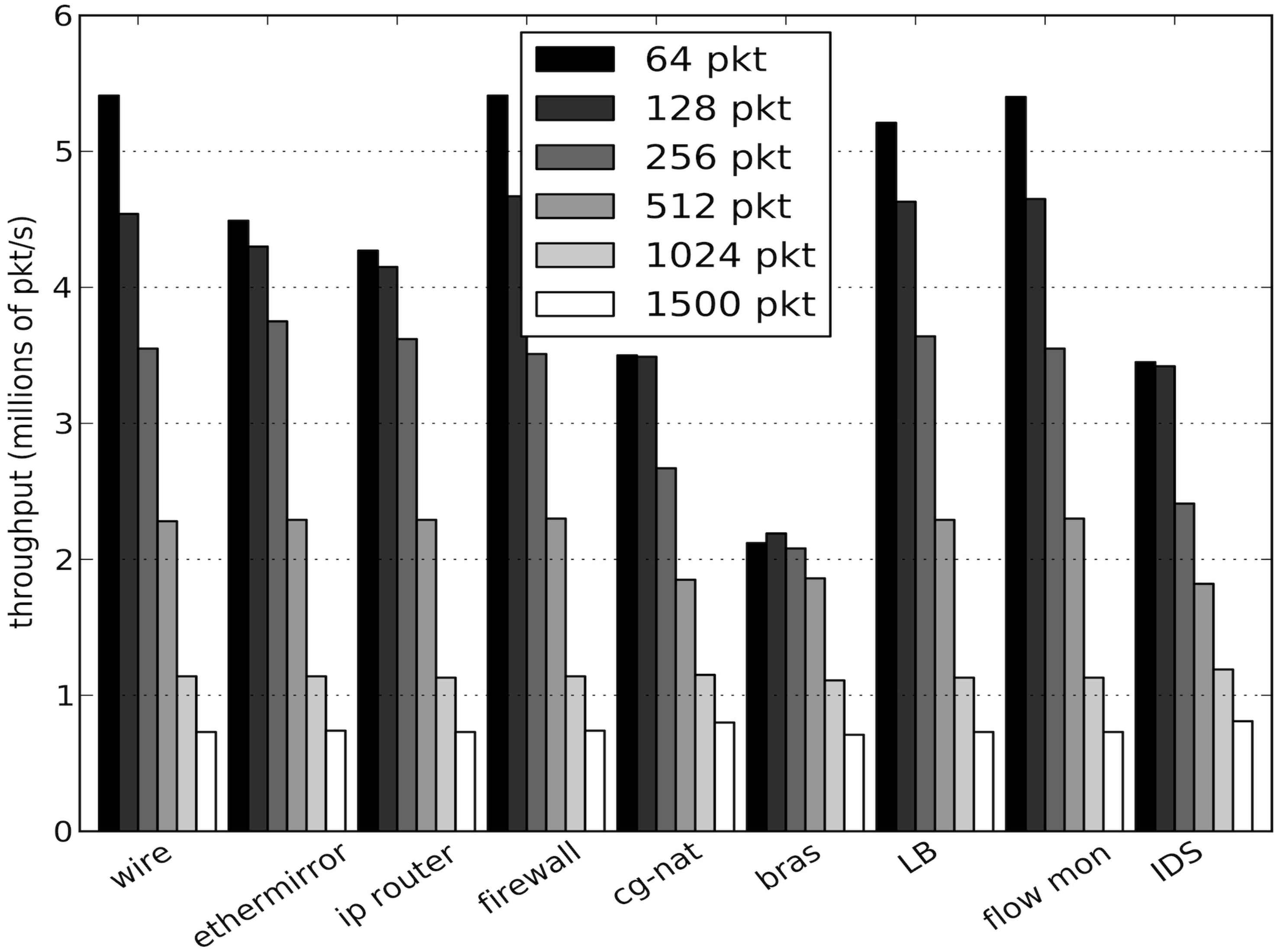
- Introduce VALE/netmap as backend switch in XEN
  - Same switch is available also for KVM/QEMU
- Permanently map grants with backend (not once per packet)
- Bypass kernel network stack for high speed packet I/O
- Larger I/O request batches
- Split interrupts for transmission and receipt

Optimizations result in 10Gb/s line rate for almost all packet sizes

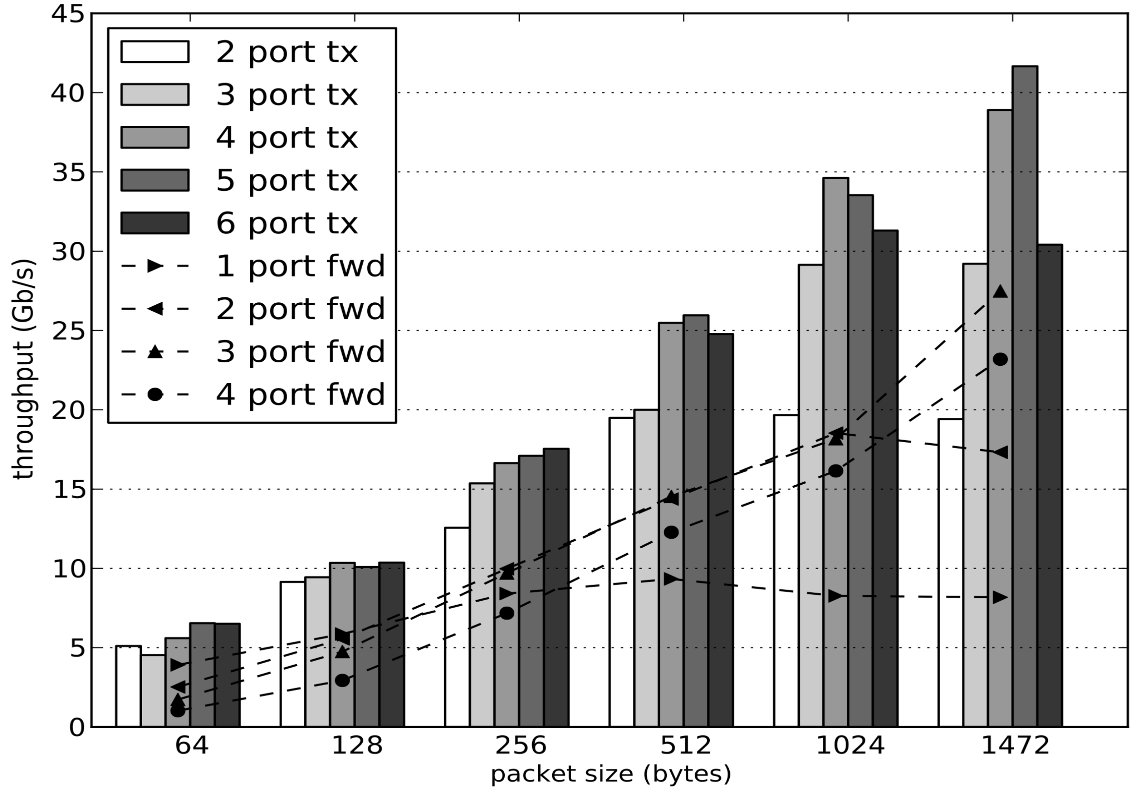
Experiment Setup

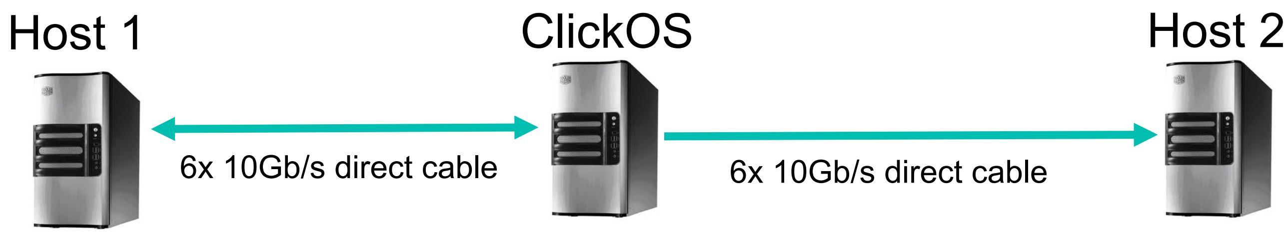
Intel Xeon E1220 4-core 3.2GHz (Sandy bridge)
16GB RAM, 2x Intel x520 10Gb/s NIC.
One CPU core assigned to Vms, 3 CPU cores Domain-0
Linux 3.6.10

Middlebox Performance (single VM)



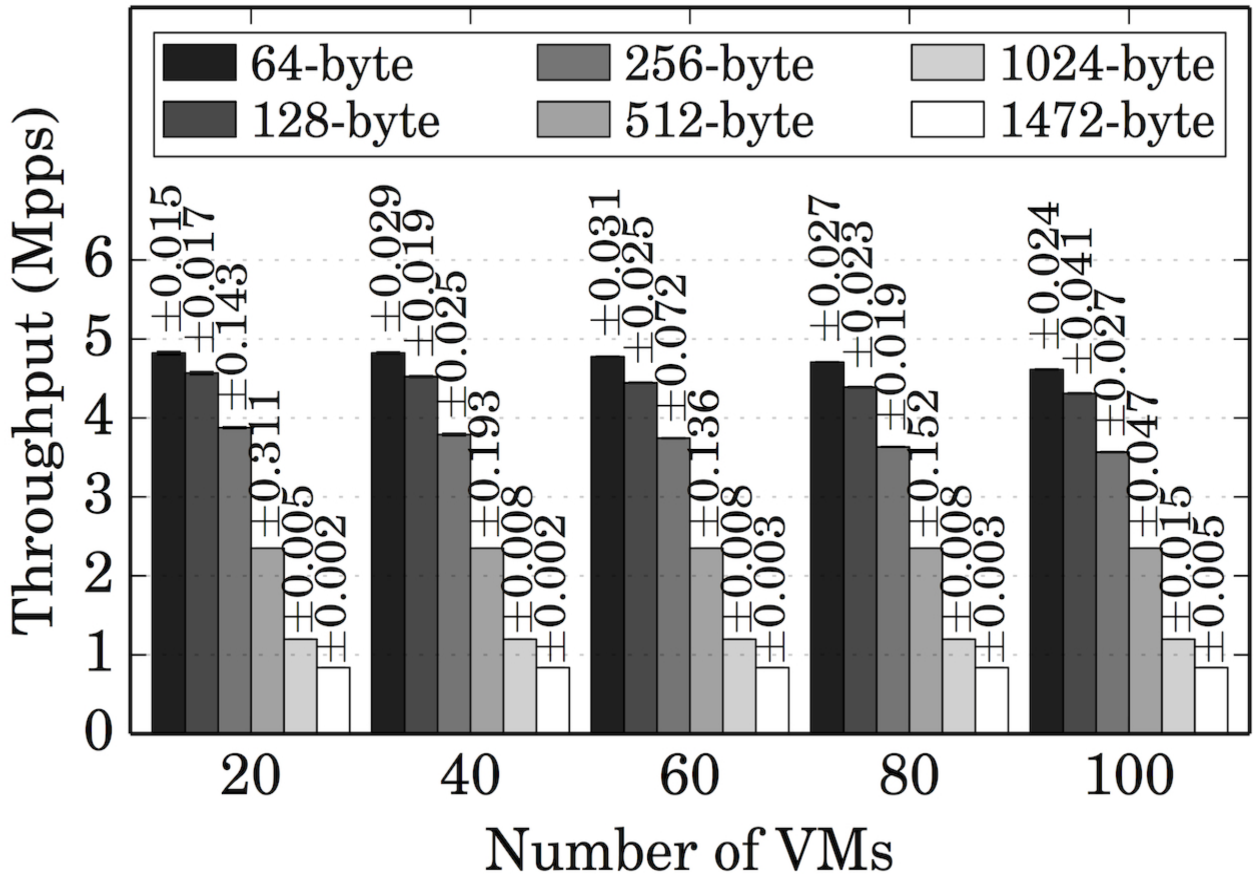
Scaling out – Multiple NICs/VMs





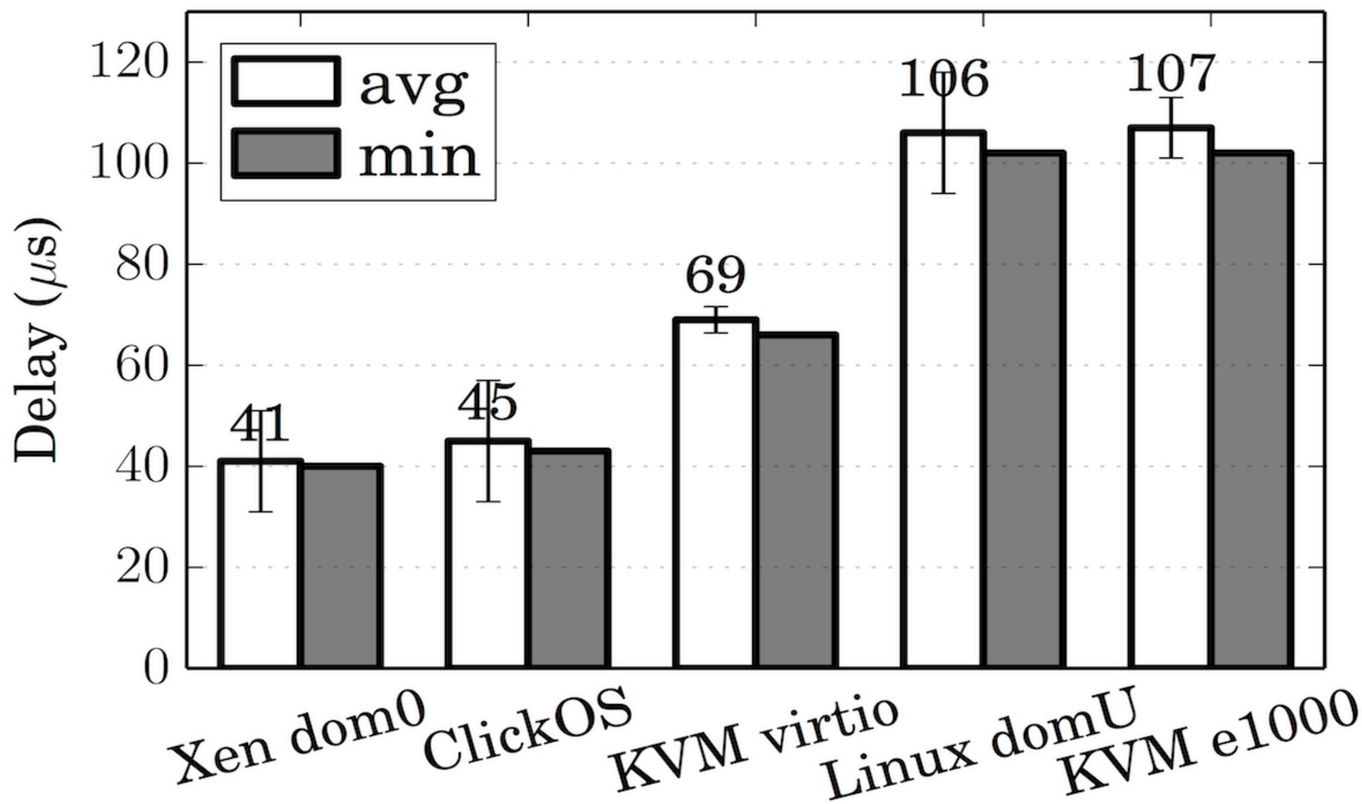
Intel Xeon E1650 6-core 3.2GHz, 16GB RAM, dual-port Intel x520 10Gb/s NIC.
3 cores assigned to VMs, 3 cores for dom0

Scaling Out – 100 VMs, Aggregate Throughput



Intel Xeon E1650 6-core 3.2GHz, 16GB RAM, dual-port Intel x520 10Gb/s NIC.
3 cores assigned to VMs, 3 cores for dom0

ClickOS Delay vs Other Systems



2. minicache: Virtualized Content Caches*

** Towards Minimalistic, Virtualized Content Caches with Minicache*
CoNEXT Hot Middlebox 2013

Overview – Virtualizing CDNs

- Current trend: Internet is becoming a “videonet”
  - 57% of Internet traffic today is video
  - 1/3 of peak traffic is the US is Netflix
  - These numbers will continue to grow

- Large majority of videos are delivered by CDNs (e.g., Akamai)
  - CDN performance is dependent on distance between content and users
    - Deploy content caches in operator networks

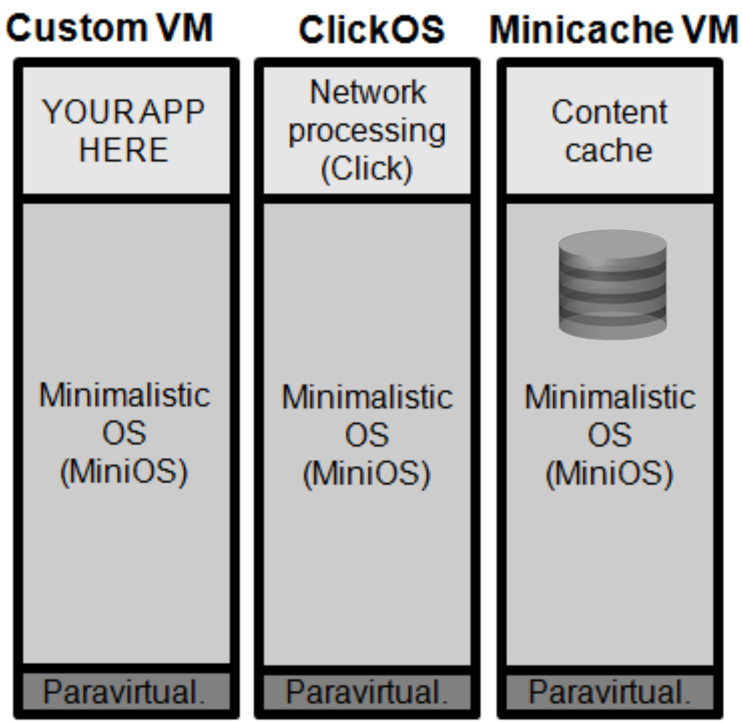
- More recently, trend towards renting infrastructure at the network’s edge
  - Micro DCs at PoPs
  - Mobile Edge Computing (e.g., next to base stations)

What's Minicache?

- Minimalistic VM for serving (video) content (CDN node)
  - Based on MiniOS
  - Uses lwIP (1.4.1) as network stack
  - Simple hash-based filesystem (SHFS)
  - Simple HTTP server
  - Interactive Shell (uSh)

- Idea: create virtual CDNs as needed, no need for upfront investments

- Added bonus: a more general VM than ClickOS, can support other types of processing



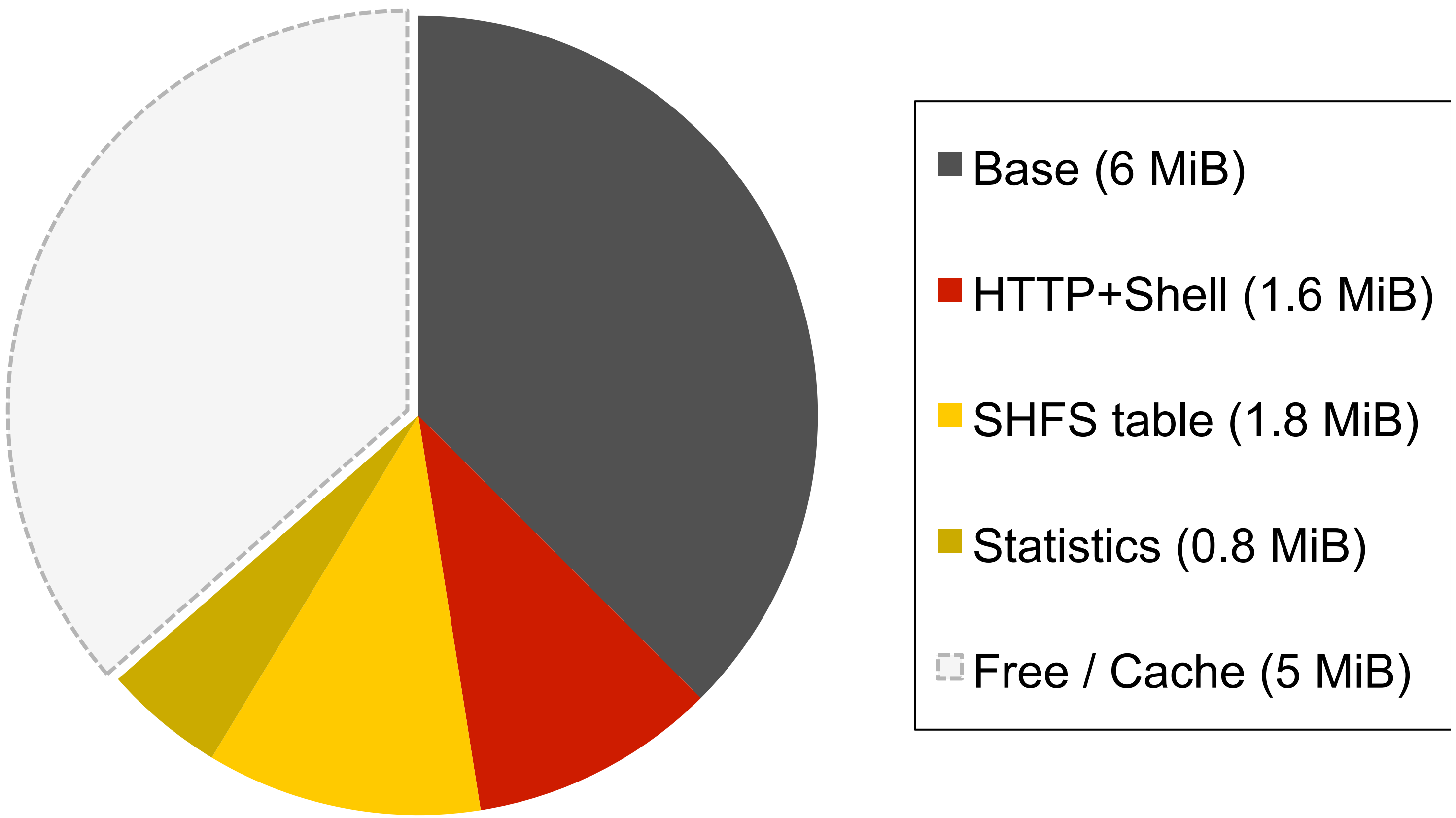
Memory Footprint

- Minimum: 8MB
  - SHFS mount adds extra memory:

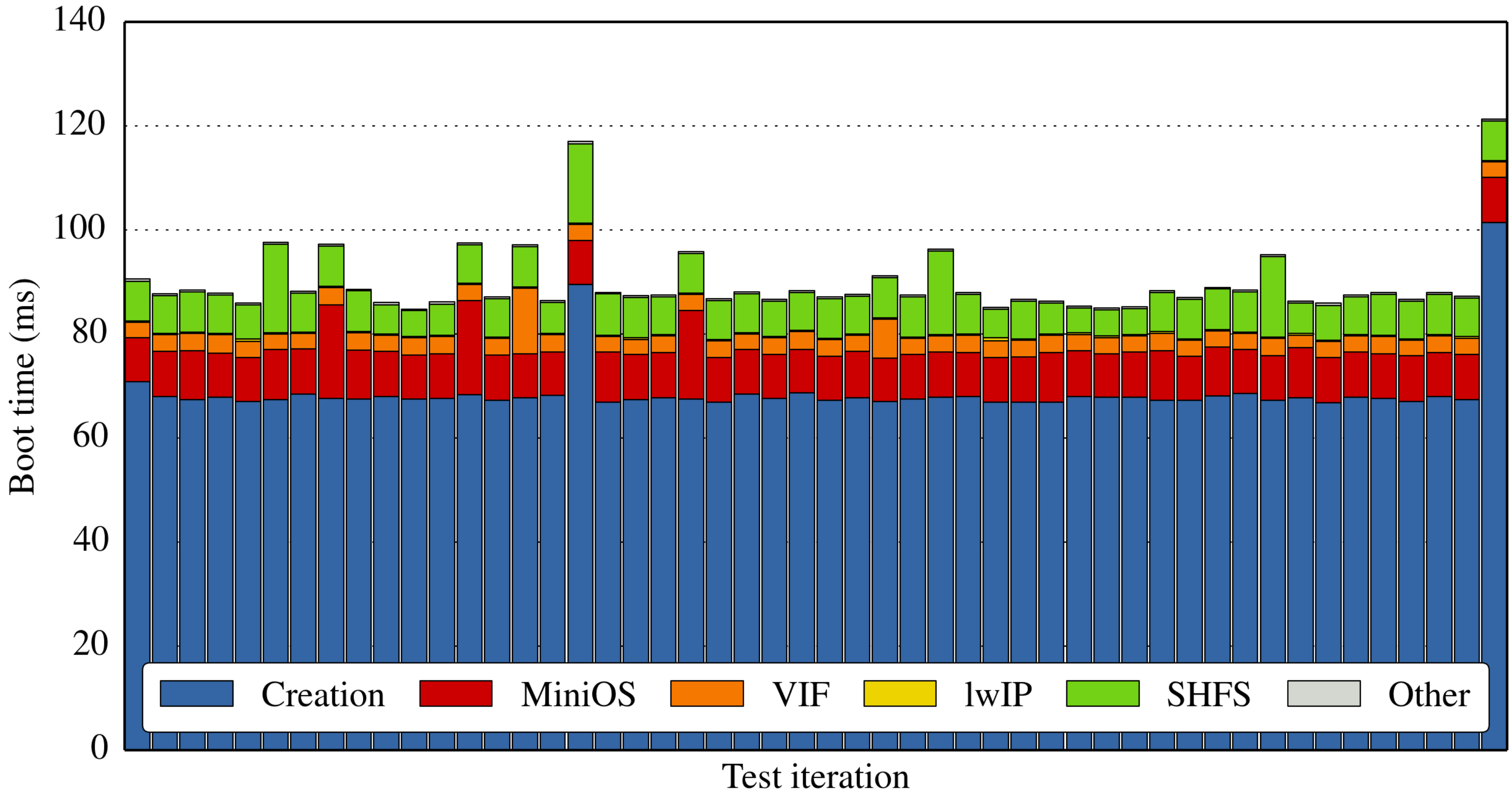
#Entries	SHFS Table size	Allocation in RAM (without stats)
512	128 KiB	230 KiB
1024	256 KiB	460 KiB
2048	512 KiB	922 KiB
4096	1 MiB	1.8 MiB
8192	2 MiB	3.6 MiB
16384	4 MiB	7.2 MiB
32768	8 MiB	14.4 MiB
65536	16 MiB	28.8 MiB

Memory Footprint - Breakdown

- 16MB Minicache VM
- SHFS mounted with 4K entries



Boot-up Times

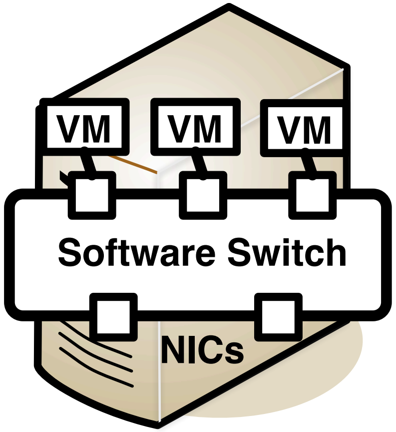


Empowered by Innovation NEC

3. VALE: a High Performance, Modular, Software Switch

Motivation

- **Software switches play an increasingly important role**
  - Interconnection between VMs and NICs
  - SDN, Network Function Virtualization (NFV)

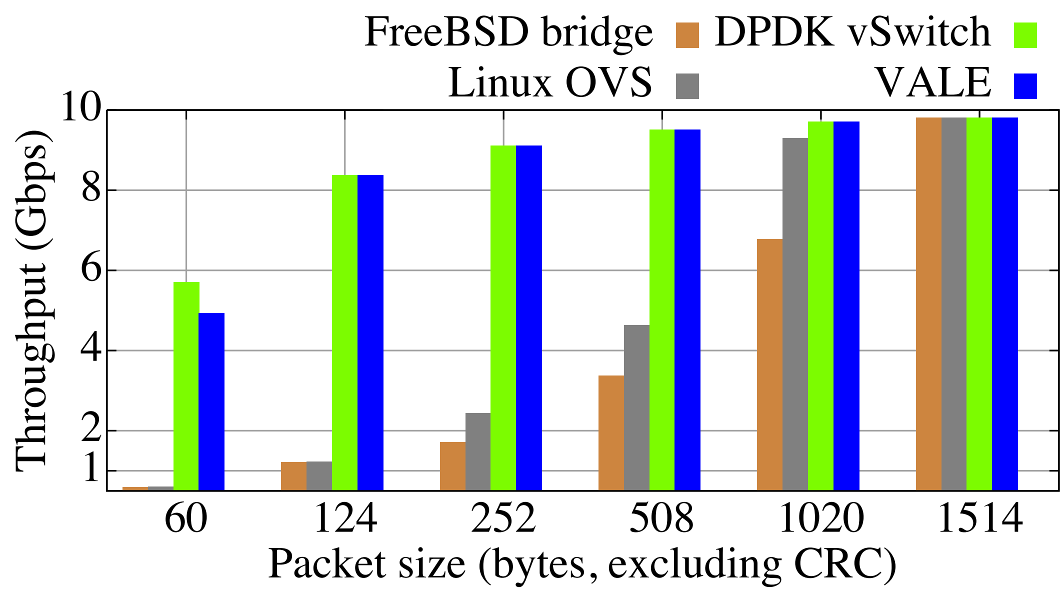


- **Requirements**
  - Throughput (e.g., 10 Gbps)
  - Scalability (e.g., 100 ports)
  - Flexibility (i.e., forwarding decision and packet processing)
  - Reasonable CPU utilization

- **Do existing software switches satisfy these requirements?**

Existing Software Switches

- OS standard switches lack high throughput
  - Small packets are common (e.g., TCP SYNs, ACKs)
- Recent switches lack scalability, flexibility and/or reasonable CPU utilization



	Throughput	CPU Usage	Density	Flexibility
FreeBSD switch	×	√	√	×
Linux switch	×	√	√	×
Open vSwitch	×	√	√	√
Hyper-Switch	×	√	×	√
DPDK vSwitch	√	×	×	√
CuckooSwitch	√	×	×	×

Our Contribution

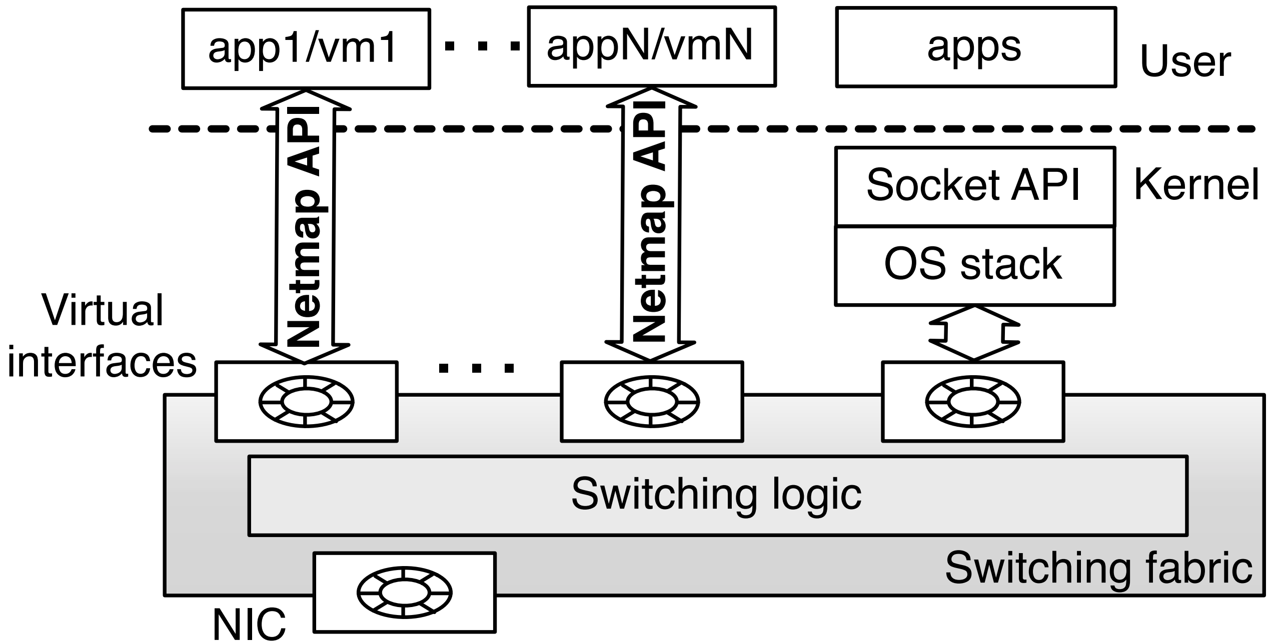
A scalable, modular software switch

- Ideal as a virtualization backend

Scalable packet forwarding algorithms

- Tens to hundreds of destination ports
- Concurrent senders to a common destination port

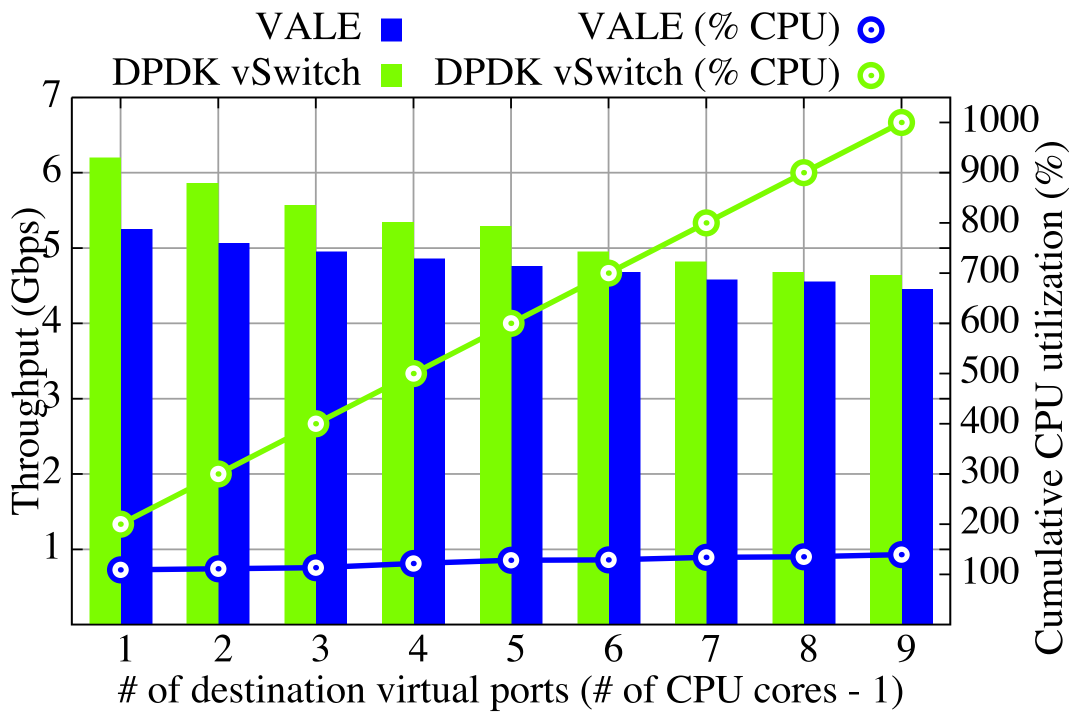
System Architecture

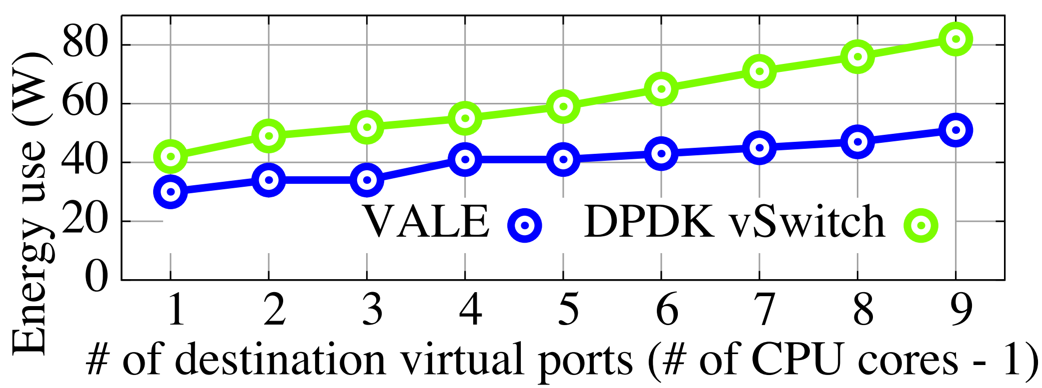


- **Switching fabric** moves packets efficiently among ports → part of the system
- **Switching logic** decides packet's destination → the user develops this

  Empowered by Innovation NEC

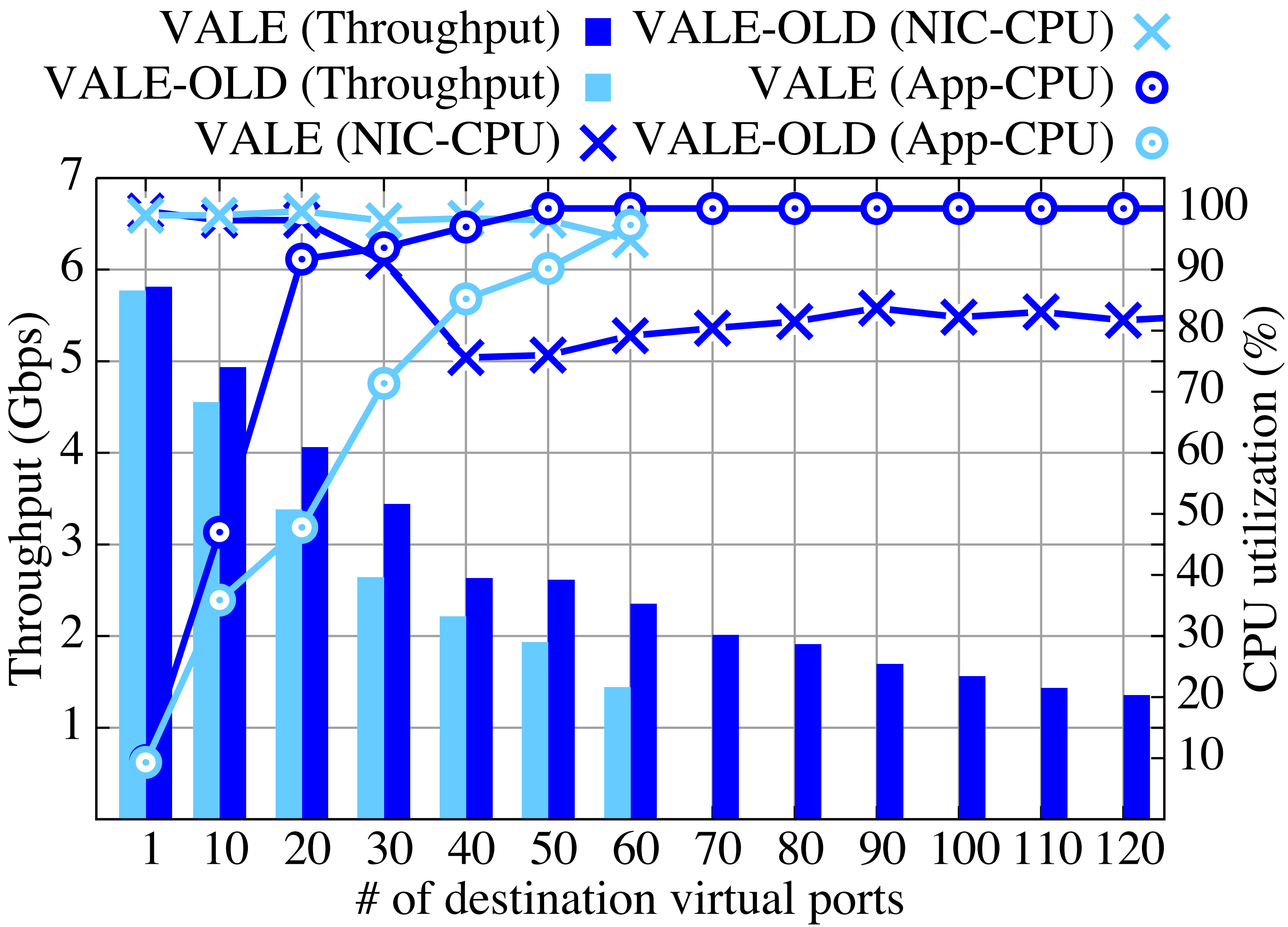
CPU utilization and Power Consumption, VALE vs OVDK





Empowered by Innovation

Port Scalability



4. Massive Consolidation*

**Towards Massive Server Consolidation*
Xen Developer Summit, 2014

Wouldn't it be Nice if…

- Thousands of guests on a single server, up to 100K
- Extremely fast domain creation, destruction and migration
  - Tens of milliseconds
  - Constant as number of guests increases

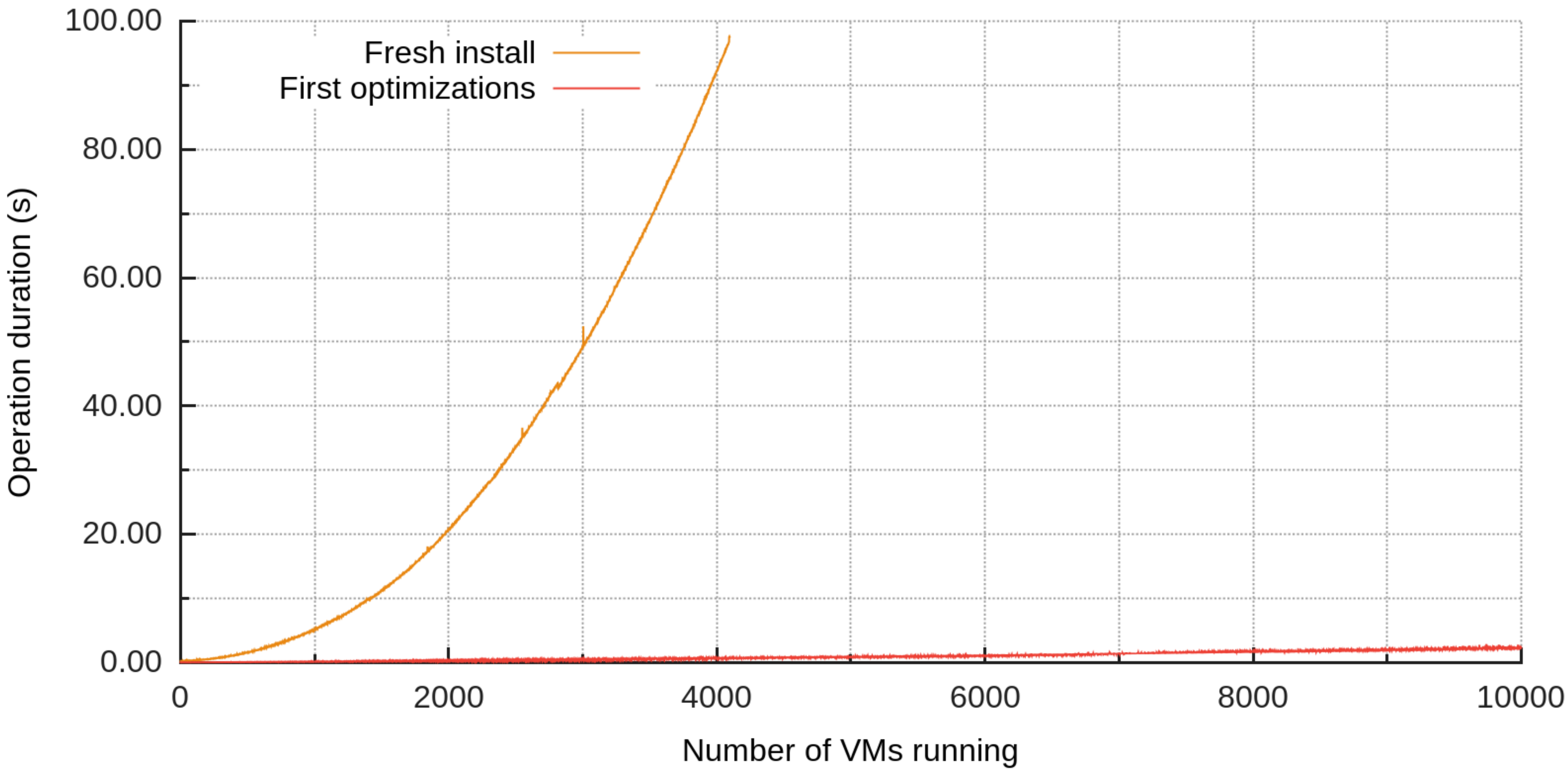
Two Types of Problems

- Hard limitations
  - Prevent guests from booting correctly
  - Only ~300 guests fully usable
- Performance limitations
  - Decreasing system performance
  - System (dom0) unusable after just a few hundred guests

First Optimizations

- Increase number of file descriptors in Linux
  - fixes console issues
- Increase number of PTYs in Linux
  - fixes console issues
- Upgrade to Xen 4.4 + Linux 3.14, kernel with NR_CPUS=4096
  - fixes # of event channels limit
- Use multiple instance of back-end switch
  - fixes # of virtual ports limitation

First Optimizations - 10K VM Boot Times



Server: 64 Cores @ 2.1GHz [4 x AMD Opteron 6376]
128GB RAM DDR3 @ 1333MHz

With Optimizations…

- Improvement: system is still usable after 10K guests
  - Although domain creation time is far from ideal

- However...
  - xenstored still CPU heavy
  - xenconsoled still CPU heavy

Current Status

- Usable system running 10K guests
  - 10K guests actually working…
  - …although idle most of the time
- Lower domain creation times
  - First domain: < 10ms
  - With 10K domains: < 100ms

- Currently working on
  - Xenconsoled: switch from poll to epoll: CPU util down to 10% max
  - Improved XenStore (lixs, **Li**ghtweight **X**en**S**tore)
  - Simplified control toolstack (xcl: **X**en**C**trl **L**ight)

Will it Work up to 100K VMs? Remaining Issues

- Improve lixs and Xenstore protocol
- Have guests doing useful work
- Scheduling
  - Number of guests much bigger than number of cores
  - With that many guests we'll have scheduling issues
- Reducing Memory Usage
  - Smaller image sizes
  - Share memory between guests booting same image

Wrap-Up

Conclusions

Introduced a number of technologies and technologies in support of a more “fluid” network cloud

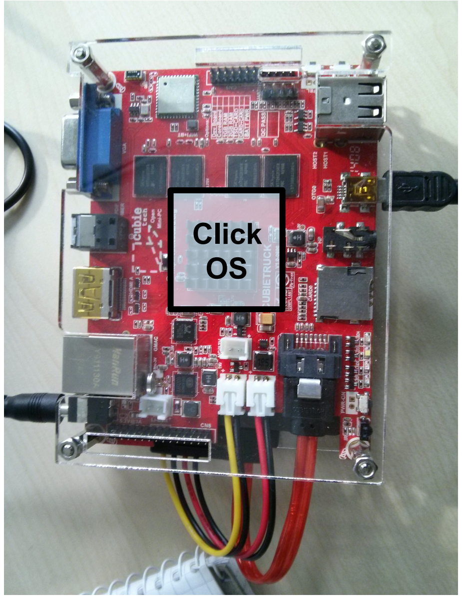
- Massive consolidation
- On-the-fly service instantiation (in milliseconds)
- Fast migration (hundreds of milliseconds)
- High throughput (10-40+ Gb/s)

Tailor-made operating system, supports

- Network processing functions (e.g., firewall, tunnel endpoint, etc.)
- Content caching (MiniCache)
- **Your application!**

Ongoing and Future Work

- Integration with OpenStack/Neutron
- Started porting to KVM (OSv & MiniOS)
- Support for ARM platforms
  - Cubietruck already working
  - ARM64 when available



CubieTruck

- We’re looking for operators for PoCs/trials…

Questions?

Cloud Networking Performance Lab

http://cnp.neclab.eu

felipe.huici@neclab.eu

\Orchestrating a brighter world

NEC brings together and integrates technology and expertise to create
the ICT-enabled society of tomorrow.
We collaborate closely with partners and customers around the world,
orchestrating each project to ensure all its parts are fine-tuned to local needs.

Every day, our innovative solutions for society contribute to
greater safety, security, efficiency and equality, and enable people to live brighter lives.

Empowered by Innovation

NEC