# DON'T SHARE, DON'T LOCK: LARGE-SCALE SOFTWARE CONNECTION TRACKING WITH KRONONAT

NICOLAS LE SCOUARNEC
JOINT WORK WITH FABIEN ANDRÉ,
STÉPHANE GOUACHE, ANTOINE MONSIFROT

# Typical internet access network


technicolor
FEEL THE WONDER

# vCPE Rationale

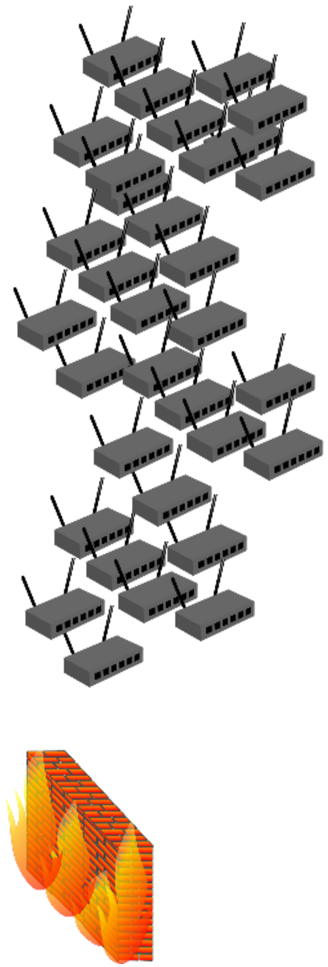
## Simplify software running on millions of embedded devices

- Easier upgrades
- Better integration

## Provide visibility into home network

- Secure IoT
- Remote troubleshooting

# Building middleboxes for residential networks




Middlebox
(e.g., NAT and Firewall)

Internet

Customer Premise Equipment
(Residential Gateway)



technicolor
FEEL THE WONDER

# What (not) to use ?

## NFV approach (virtualized appliances)

- One VM/container per customer
- Running existing software (e.g., OpenWRT or Linux)
- As done for example in R-CORD

## Virtual Switches for traffic dispatching to VM

**Does not scale to millions of VMs/containers**

**Not cost effective**

technicolor
FEEL THE WONDER

# Which equipment to use ?

| | Vendor B (HW Appliance) | Vendor C (HW Appliance) | 2x Xeon E5 v4 (40 cores) | Vendor A (VM) | StatelessNF (NSDI'17) |
|---|---|---|---|---|---|
| ***L4 Throughput (Simple IMIX)*** | 58 Mpps<br>130 Gbps | 63 Mpps<br>140 Gbps | *180 Mpps*<br>*400 Gbps* | 4,5 Mpps<br>10 Gbps | 4 Mpps<br>10 Gbps |
| ***Cost*** | 65 K$ (HW+SW) | 200 K$ (HW+SW) | *30 K$ (HW)* | 21 K$ (SW) | NA |
| ***Redundancy model*** | 1+1 | 1+1 | *N+1* | 1+1 | N+1 |

**Available SW for running on COTS server**

Objective:
180 Mpps / server
4.5 Mpps / core

technicolor
FEEL THE WONDER

# The residential vCPE challenge

**Build a middlebox (firewall, NAT, …)**
**for residential networks**
**from COTS hardware**

**<u>Efficient</u>, <u>Reliable</u>, <u>Scalable</u>**
**L4 connection tracking**
**For millions of users**

# Best practices for high-performance networking software

## Avoid context switches

- Use kernel-bypass systems (e.g., DPDK)

## Don’t lock, don’t share

- Cross-core sharing is expensive even without explicit locking

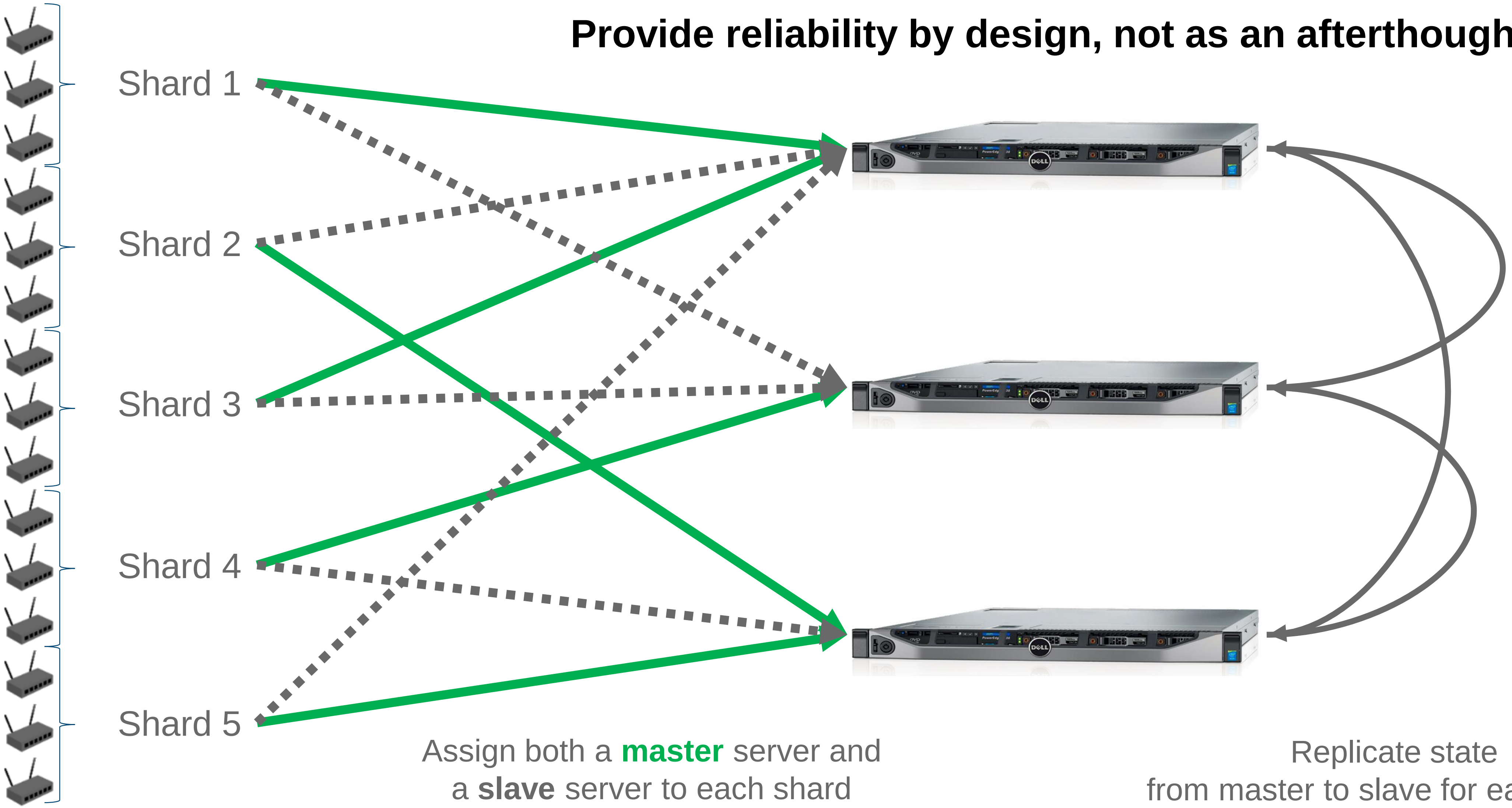
## Run-to-completion model

- Receive, process, transmit, without buffering nor blocking

## Applying all these principles everywhere is non-trivial

# Reliable - sharding and replication

**Provide reliability by design, not as an afterthought**




Assign both a **master** server and
a **slave** server to each shard

Replicate state
from master to slave for each shard

technicolor
FEEL THE WONDER

# Replication - Availability rather than Consistency

## No external DB

- Faster insertion and lookup rate (450M lookups/second on 18 cores)
- Non-blocking (no remote memory access)

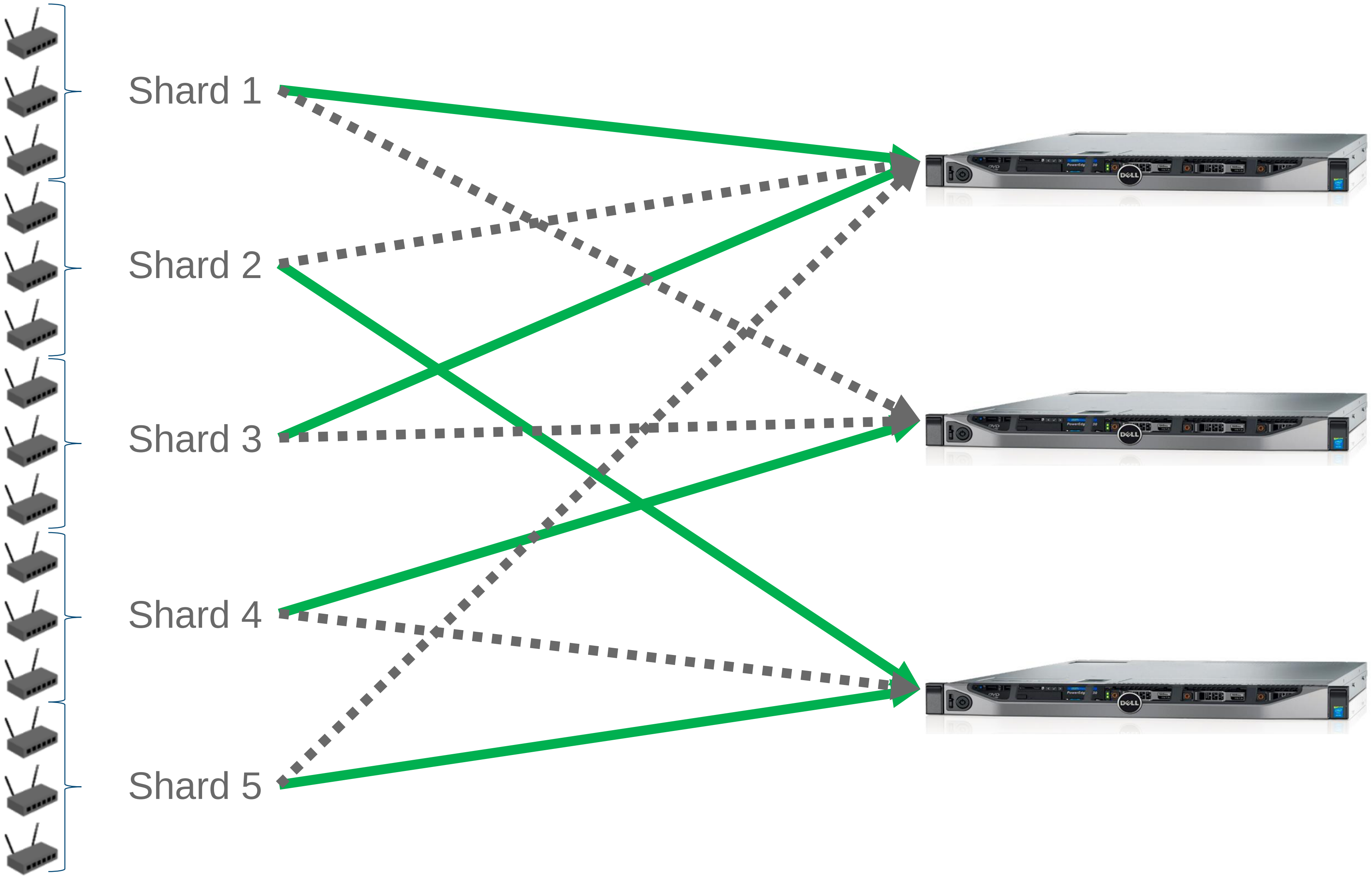
## Availability rather than consistency

- Networks are unreliable, applications will recover
- Yet, even short unavailabilities are noticed by user
- Master does not wait for acknowledgment from slave

## Efficient lock-less replication

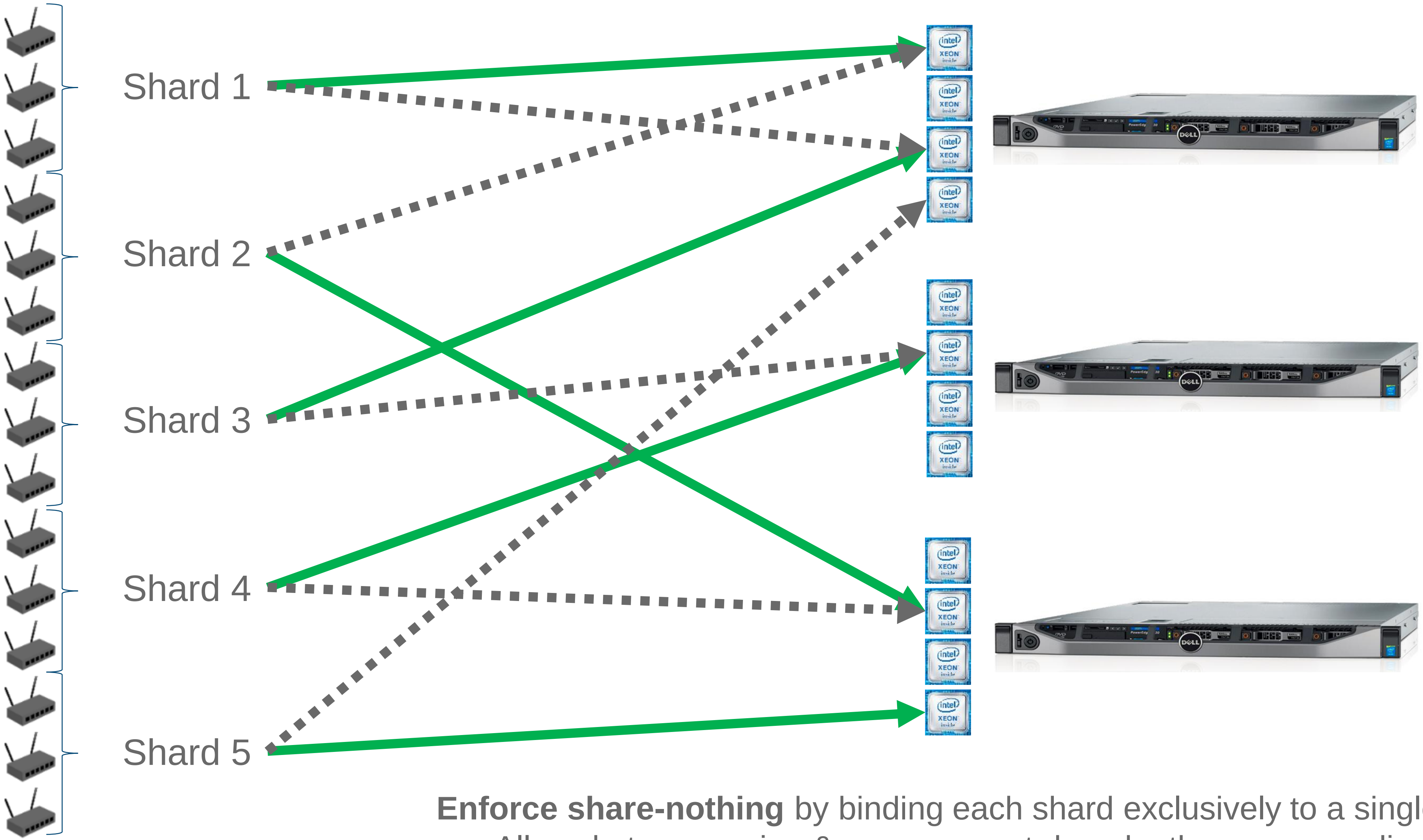
- Batching for improved performance
- **Same thread for packet processing and replication**
- Traffic not interrupted during slave initialization, using support from hash table

# Efficient (I) – Sharding to the core



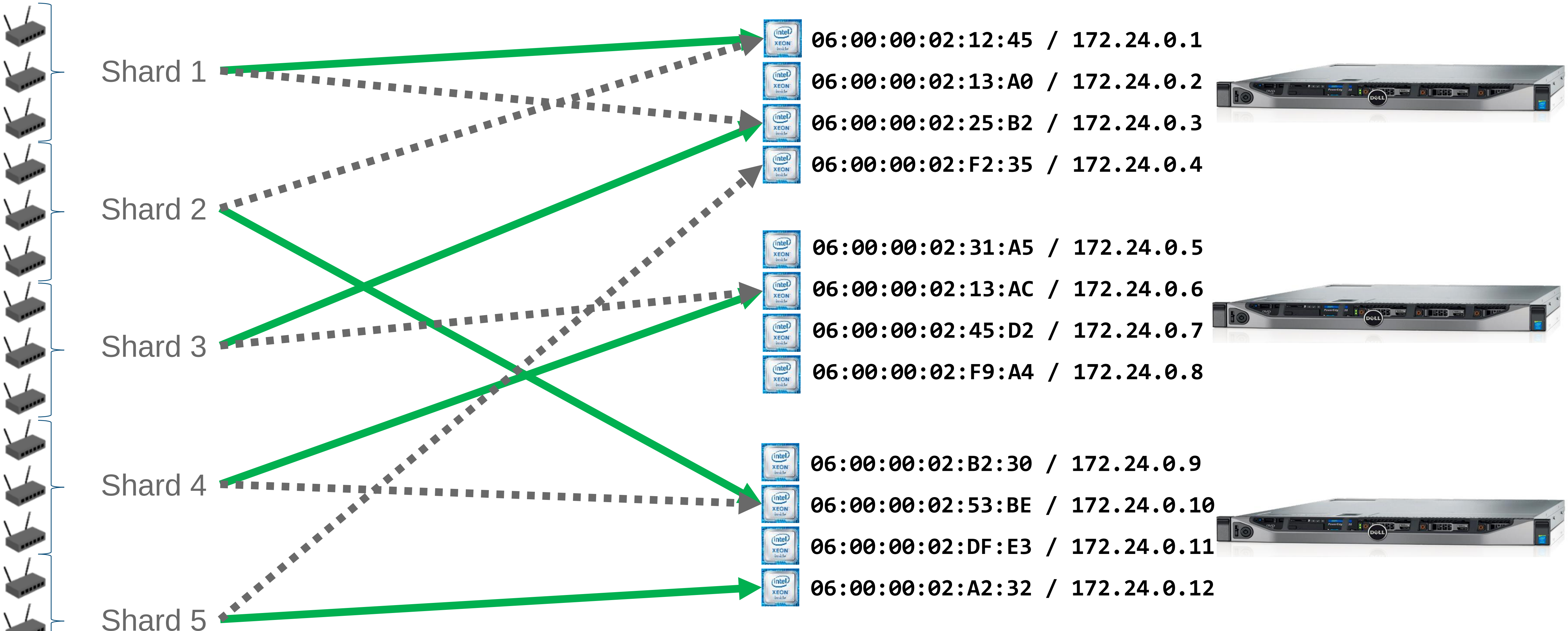

technicolor
FEEL THE WONDER

# Efficient (I) - Sharding to the core




**Enforce share-nothing** by binding each shard exclusively to a single CPU core
All packet processing & management done by the corresponding thread

# Efficient (II) - Expose each core to the network



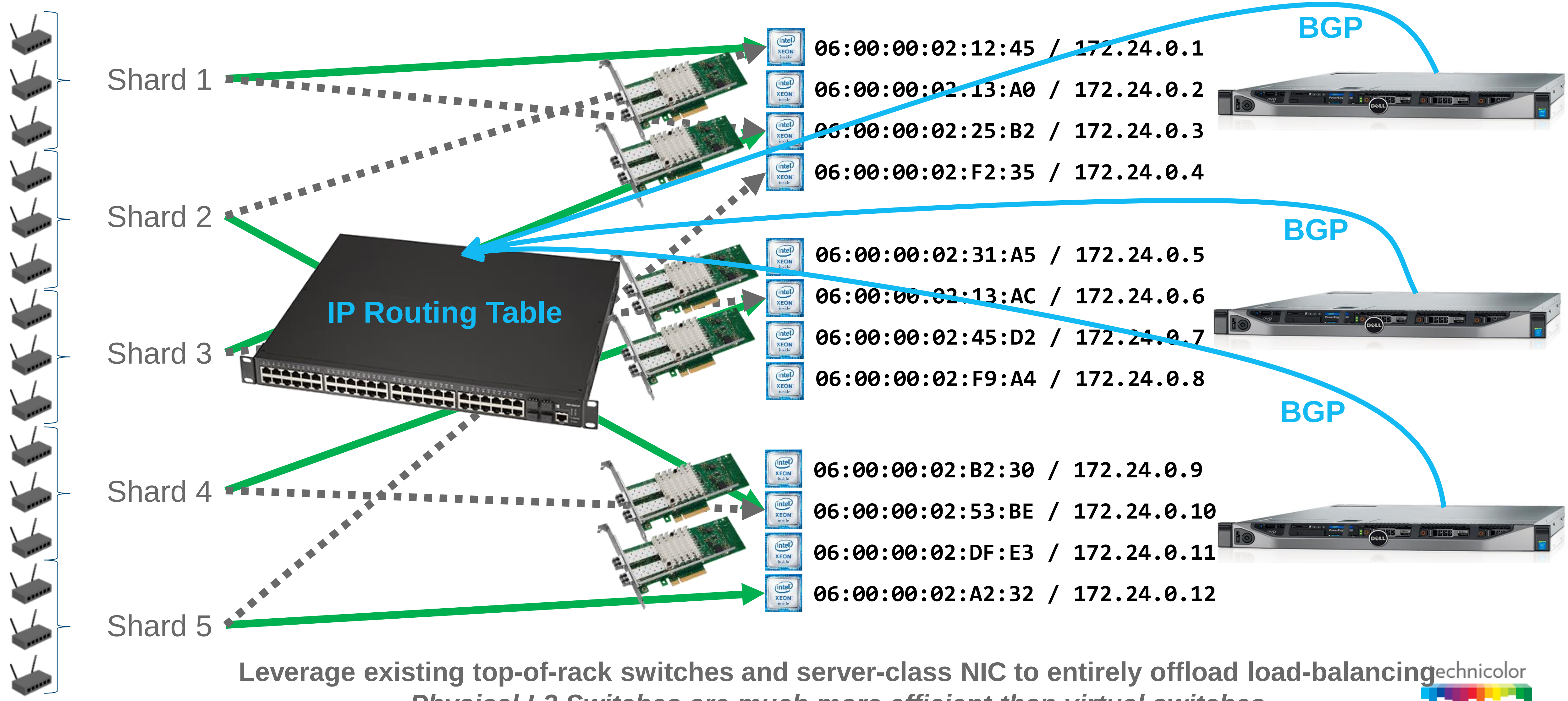

**Expose an independant identity for each core ~~(not server nor NIC)~~ on the network**

***One single mechanism to address between and within servers***

***Each core appears in the system as a independent router***

technicolor
FEEL THE WONDER

# Efficient (II) - Scalable load-balancing by NICs and Switches

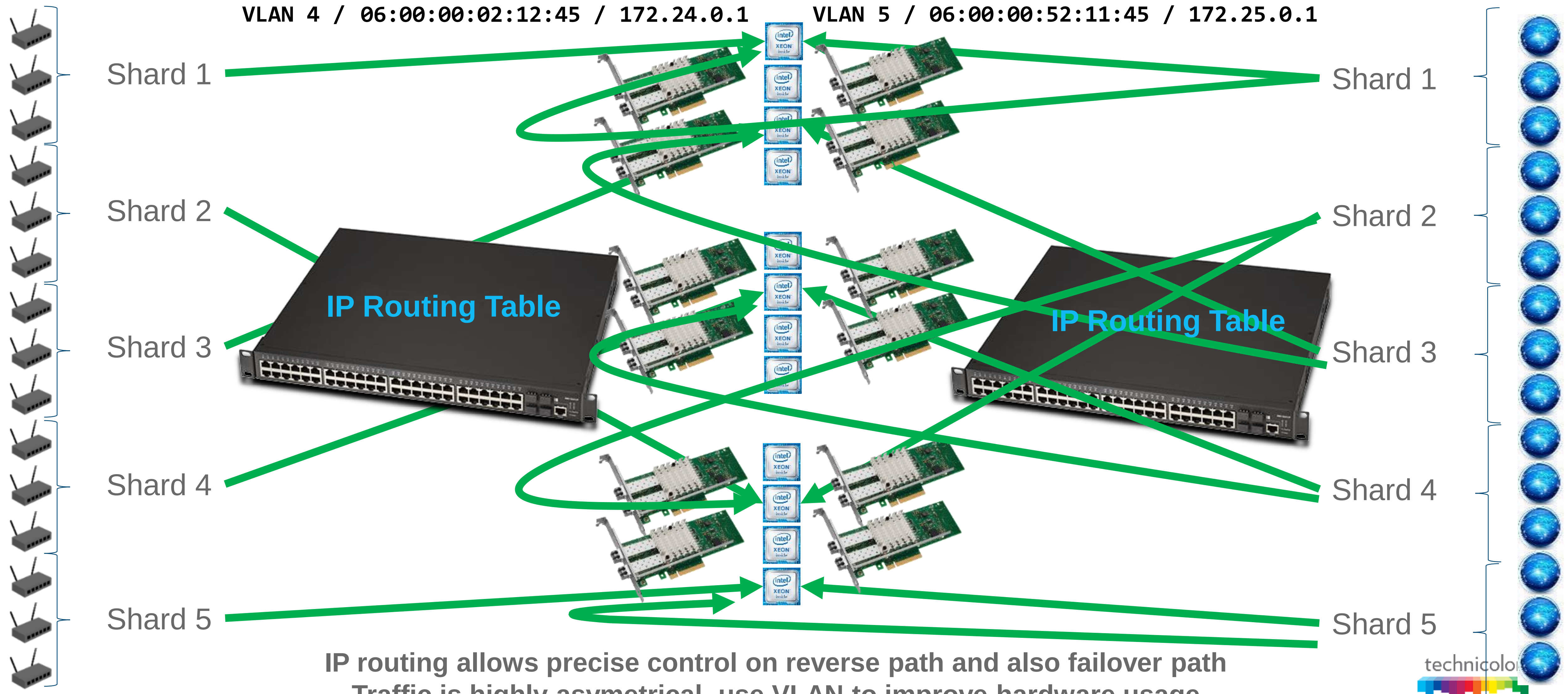



**Leverage existing top-of-rack switches and server-class NIC to entirely offload load-balancing**

***Physical L3 Switches are much more efficient than virtual switches***

technicolor
FEEL THE WONDER

# Efficient (III) – Handle reverse traffic efficiently




**IP routing allows precise control on reverse path and also failover path**
**Traffic is highly asymetrical, use VLAN to improve hardware usage**
**IP routing allows more control than RSS or ECMP based distribution**

technicolor FEEL THE WONDER

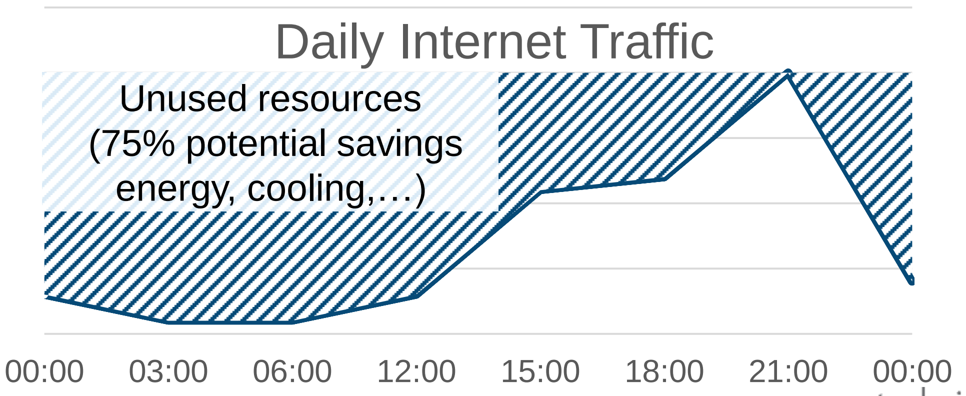
# Our design: benefits

## Distribution across servers and across cores identical

- Simplified implementation
- Performance scale linearly across cores and across servers

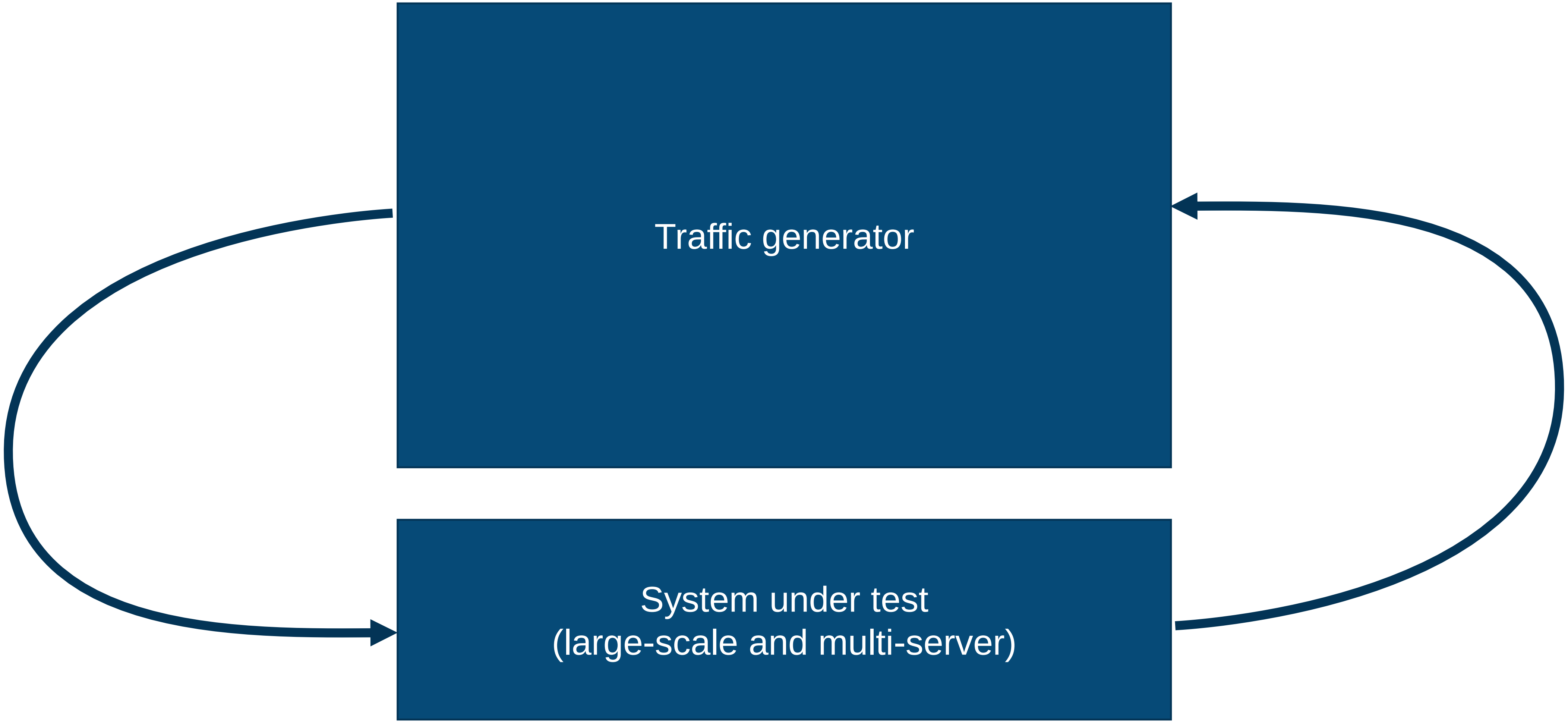
## Dynamic load-balancing included (dynamic routing + replication)

- Re-balance the load between servers
- Scale-out and in as demand evolve :  elasticity



# Benchmarking

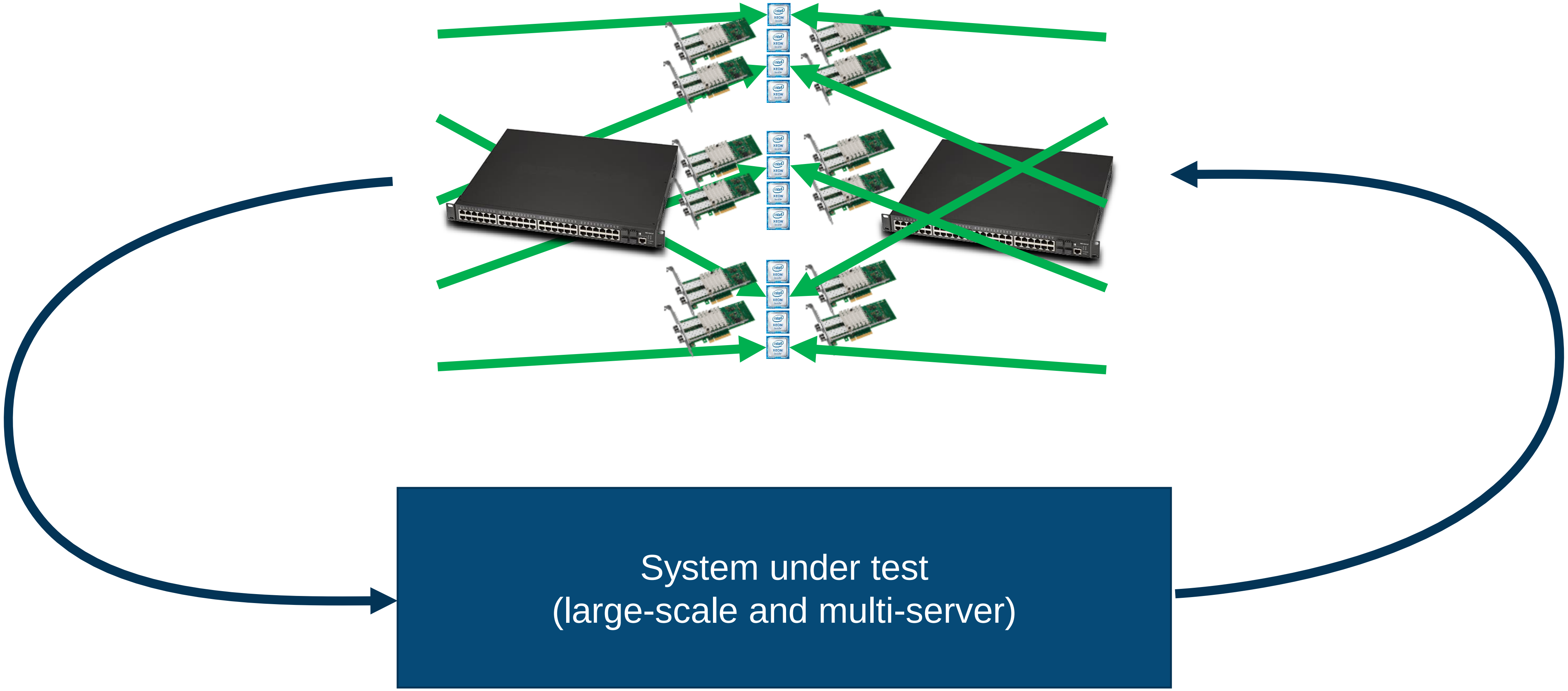
## Multi-core, multi-server benchmarking tool following the same principles




technicolor
FEEL THE WONDER

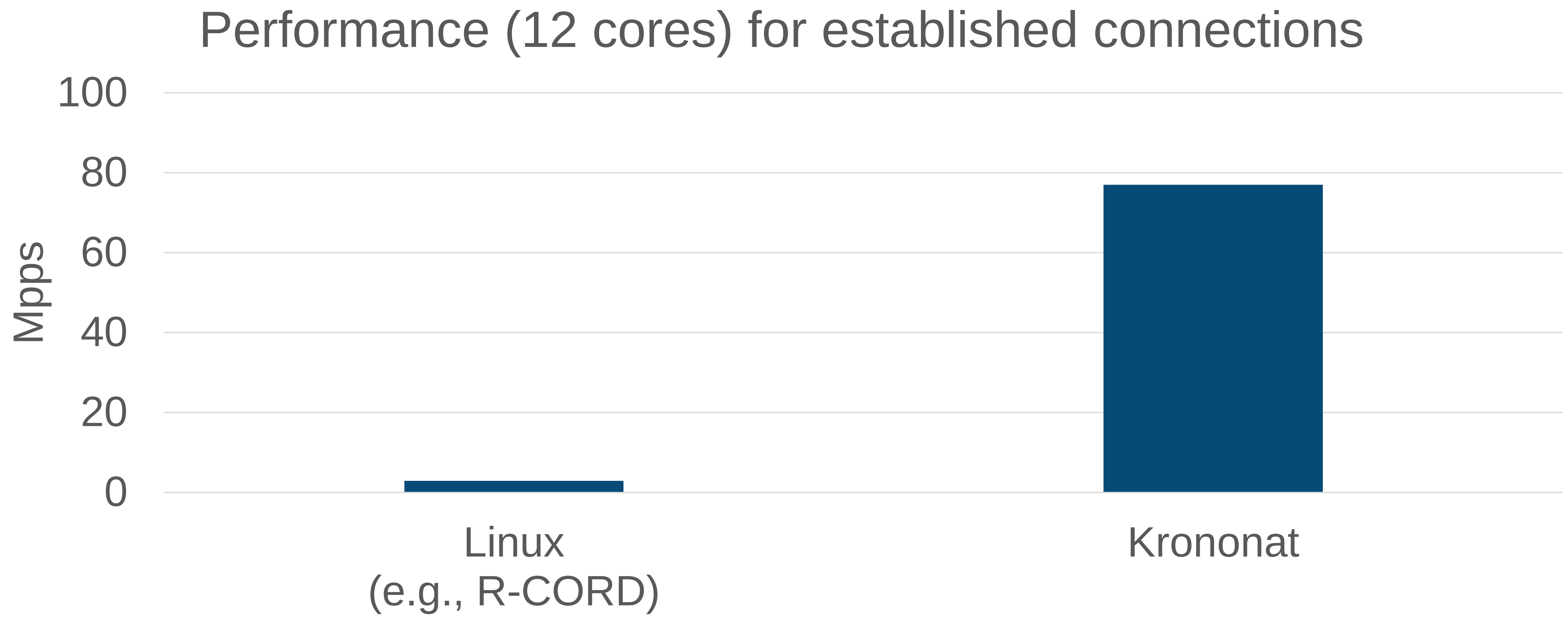
# Benchmarking

## Multi-core, multi-server benchmarking tool following the same principles

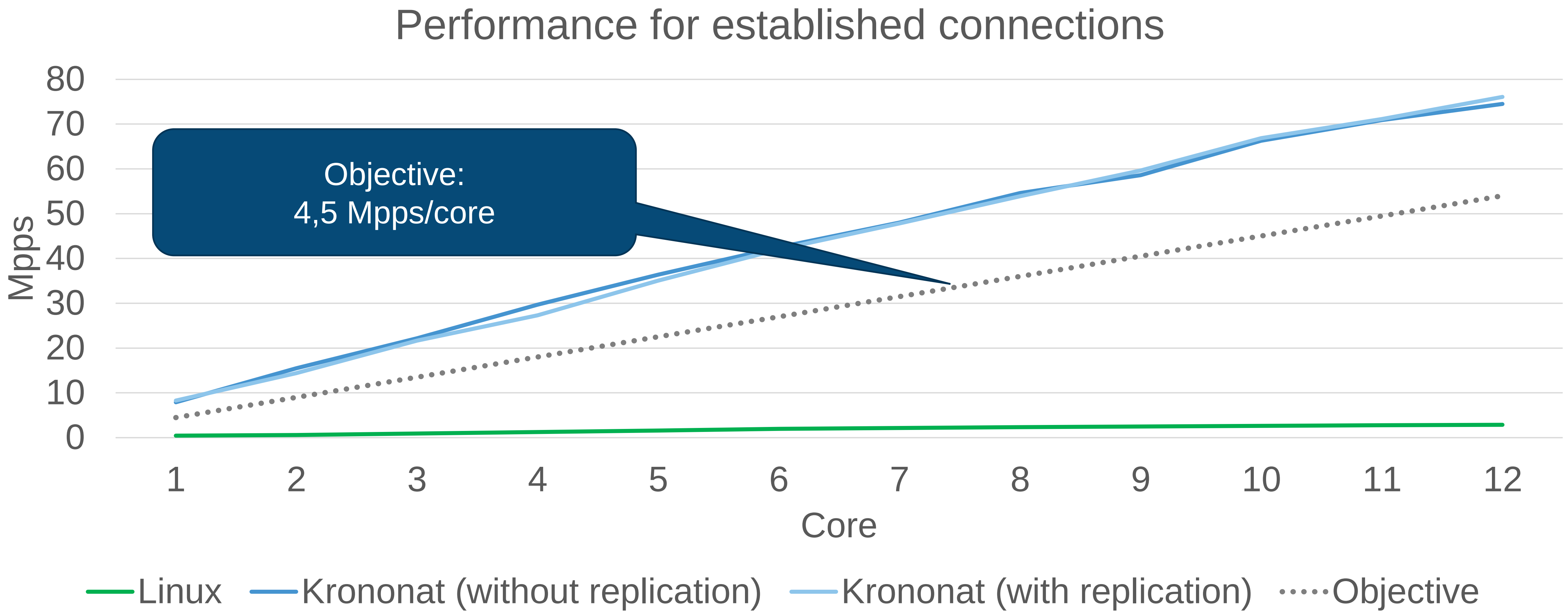



technicolor
FEEL THE WONDER

# Performance

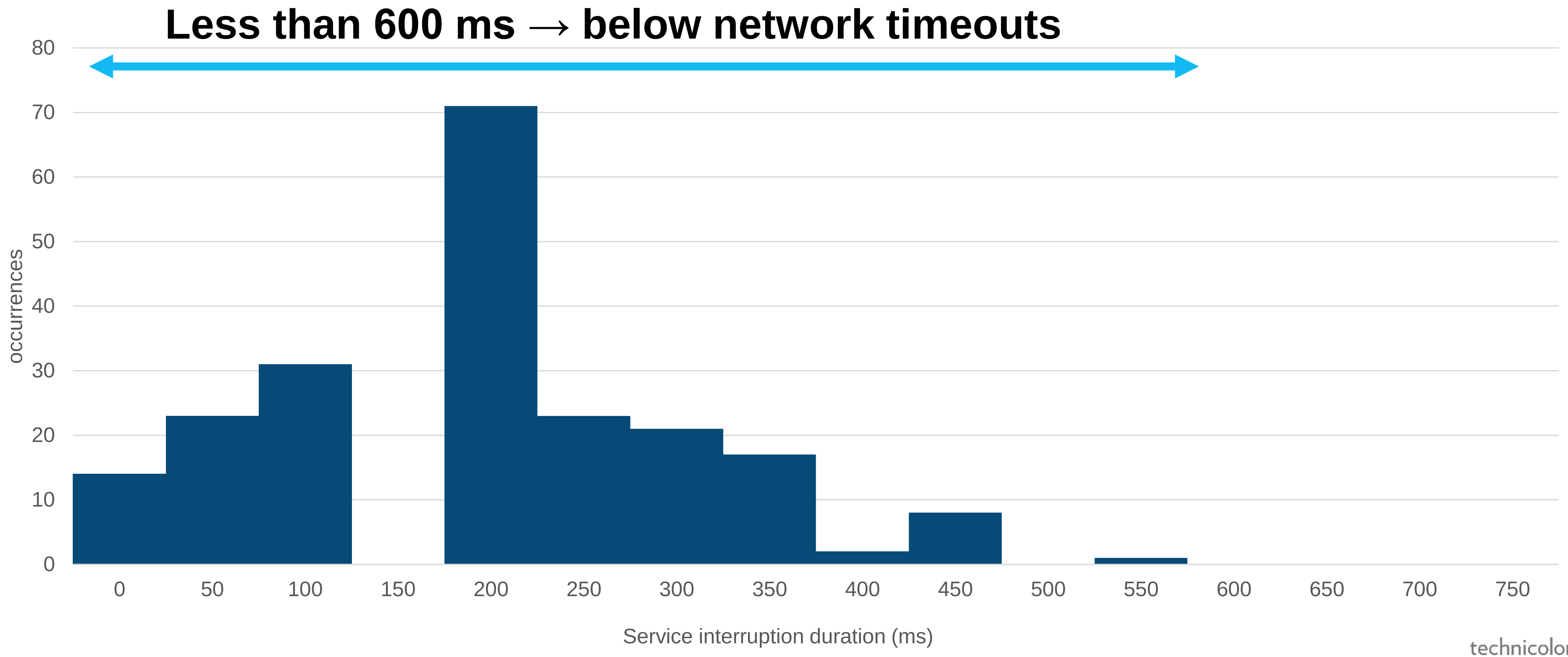


# Scalability




technicolor
FEEL THE WONDER

# Availability - Server departure





technicolor
FEEL THE WONDER

# Conclusions

## Resilient distributed middlebox using COTS hardware

- **77 million packets per second on only 12 cores**
  - 6,4 Mpps/core above objective (4,5 Mpps/core)
- Recover from failures automatically without users noticing
- **Cost-effective N+1 redundancy**
- Redundancy and dynamic load-balancing allow **elasticity**

## Re-usable design

- Expose each core as a distinct entity to the network
- Push per-core traffic steering to the networking equipments (NIC, switches)
- Applied to multi-server multi-core benchmarking tool


technicolor
FEEL THE WONDER

# References


**Don’t share, Don’t lock: Large-scale Software Connection Tracking with Krononat**

***Fabien André, Stéphane Gouache, Nicolas Le Scouarnec and Antoine Monsifrot***

**USENIX ATC’18**

**Cuckoo++ Hash Tables: High-Performance Hash Tables for Networking Applications**

***Nicolas Le Scouarnec***

**ACM/IEEE ANCS’18**

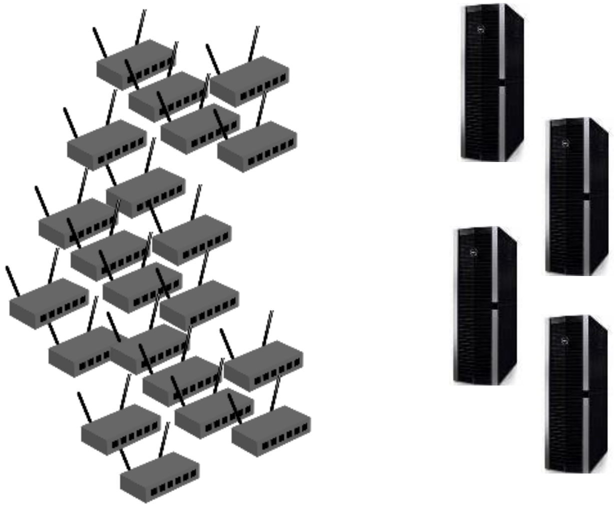

technicolor

FEEL THE WONDER

# APPENDIX

# Building a distributed software CG-NAT/FW/…

## Access network



## L4 Load-balancers

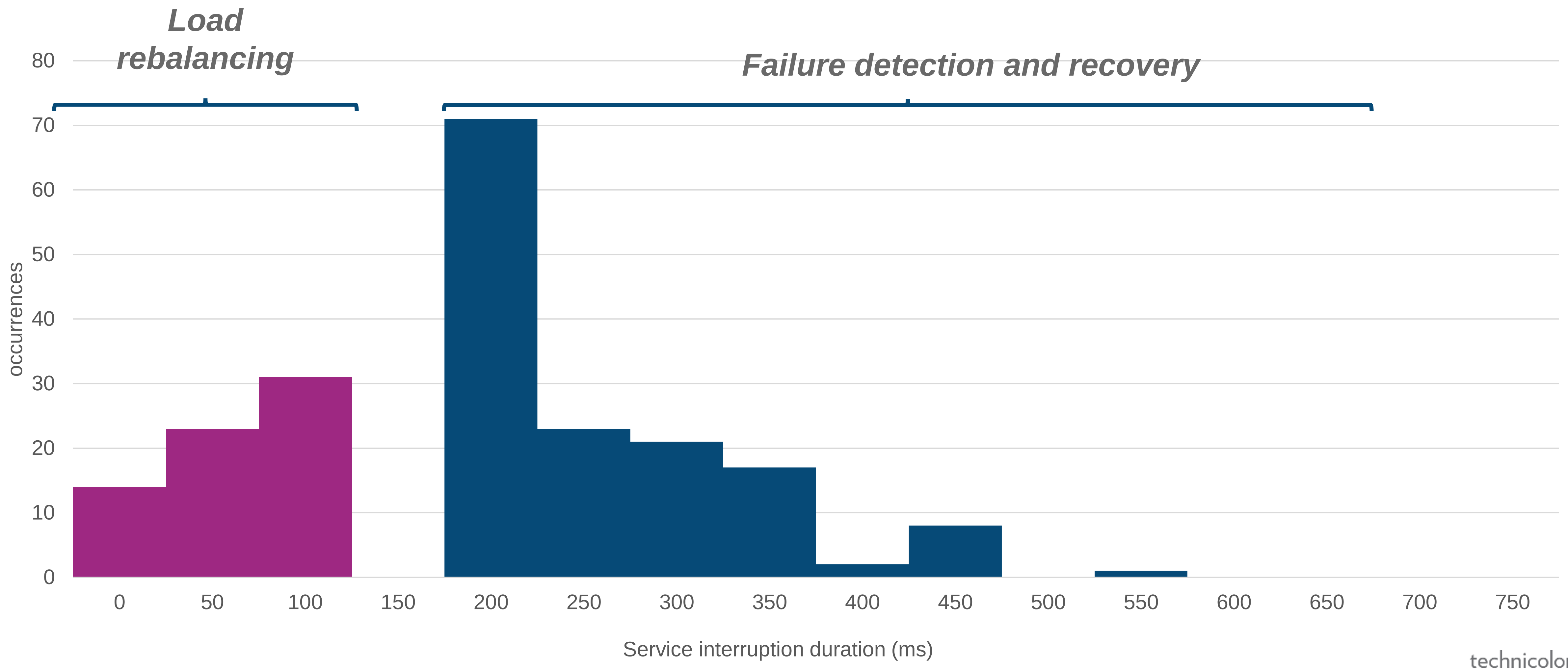
- Maglev
- Ananta
- Fastly@NSDI
- SilkRoad
- …

- Bi-directional traffic
- Must filter unknown connections

- No-reverse path traffic (DSR)
- Leverage deterministic hashing

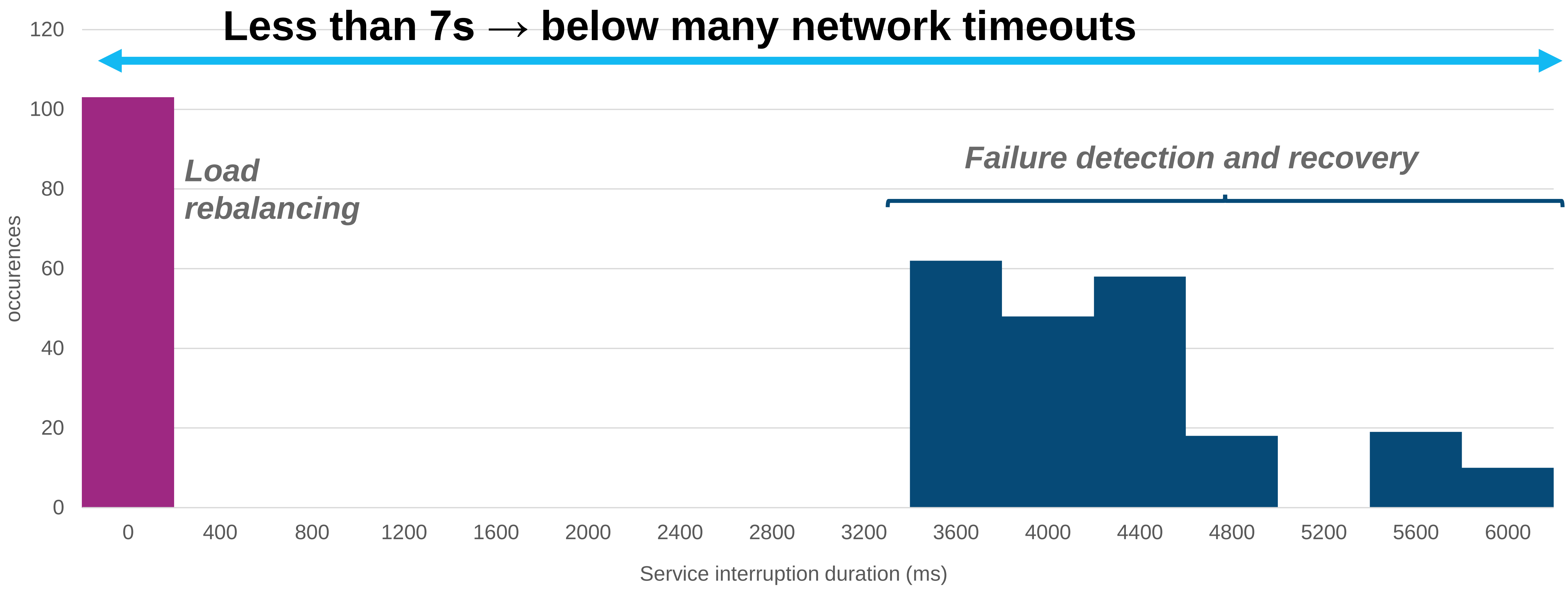
technicolor
FEEL THE WONDER

# Availability : Graceful departure




technicolor
FEEL THE WONDER

# Availability : Hard Failure




technicolor
FEEL THE WONDER