OVH.com

Innovation is Freedom

Efficient routing on multi-socket x86 machines

Jakub Słociński

Agenda

- About me
- About servers
- Network connection
- vRouter made in OVH
- Tuning
- Summary

About me

R&D network engineer

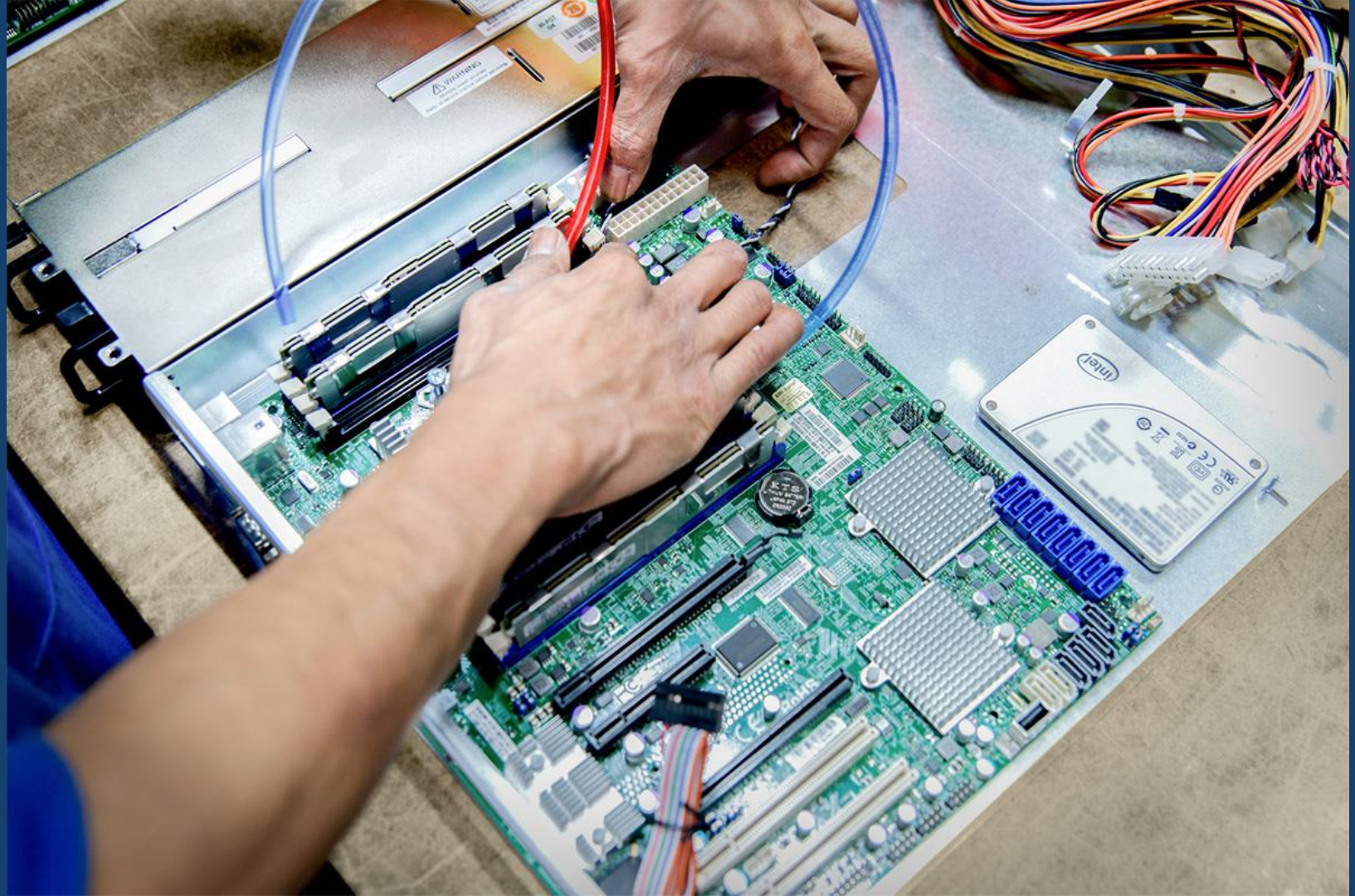
vRouter tech leader

OVH group
Innovation is Freedom

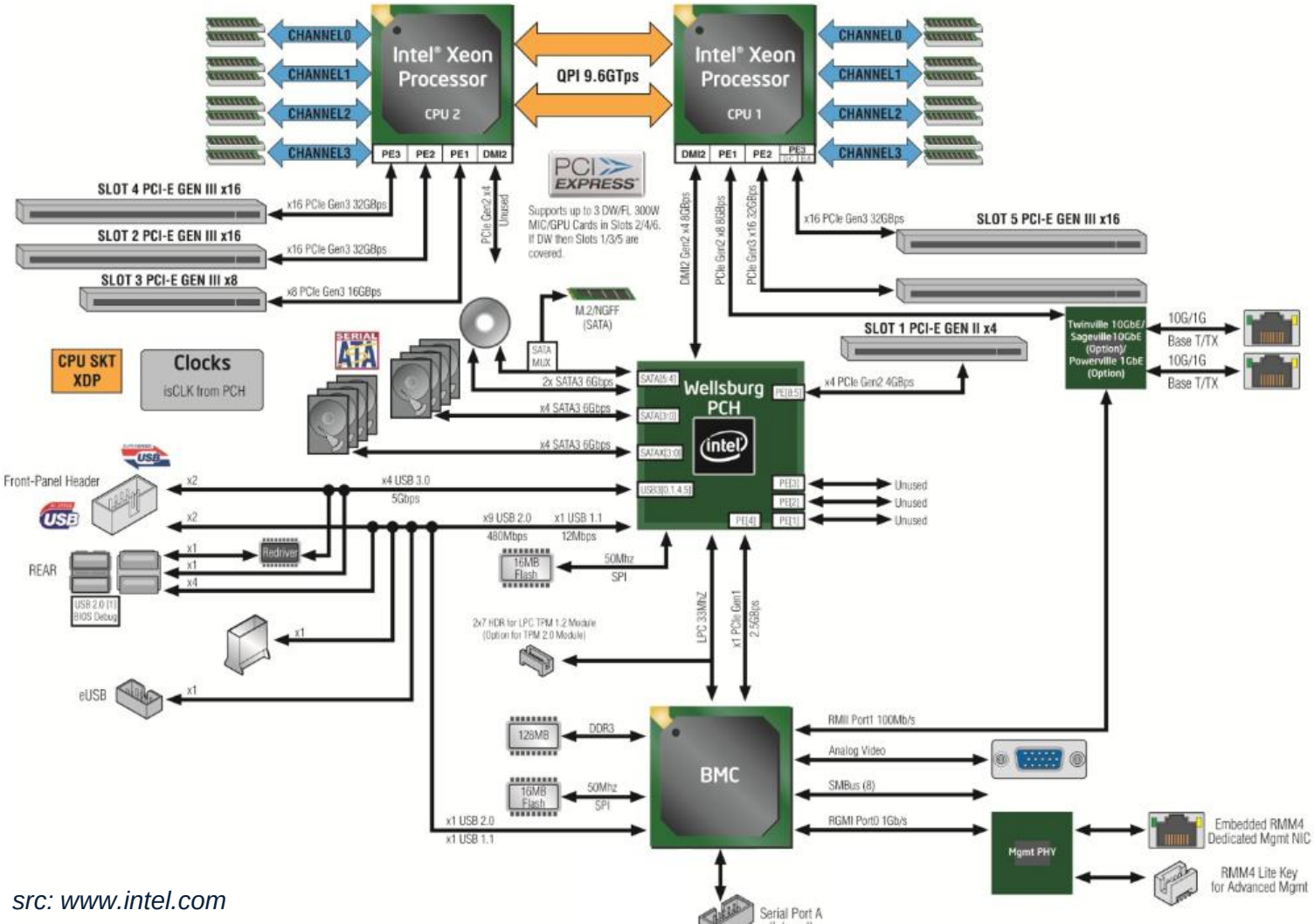
NET::IP
Meetup

Server solutions

Architecture and technology



OVH.com



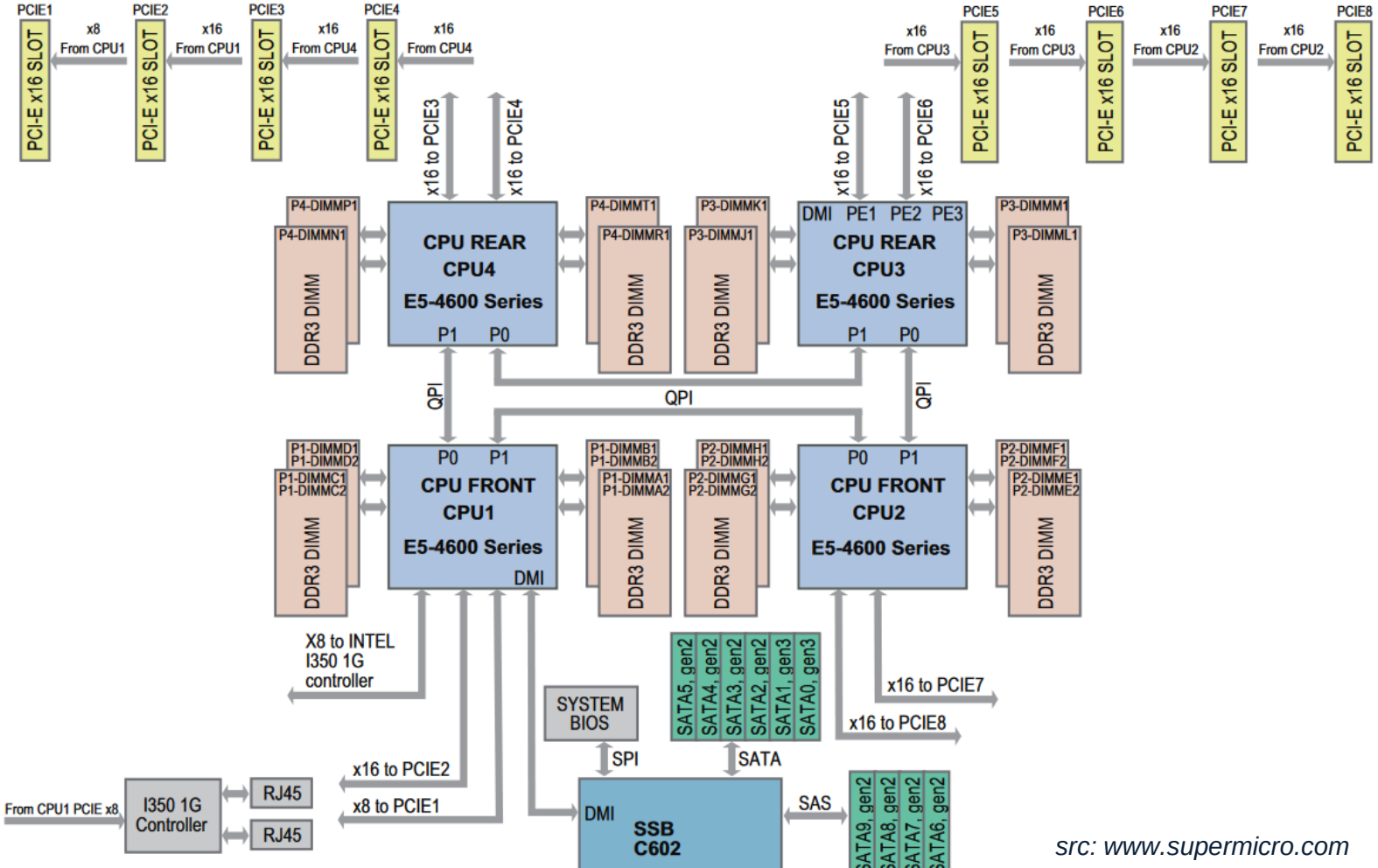
src: www.intel.com

NUMA

- NUMA (Non-Uniform Memory Access)
  - Single memory space for all CPUs
  - Fast access to local memory
  - Slower access to remote memory and I/O ports
  - Better scalability

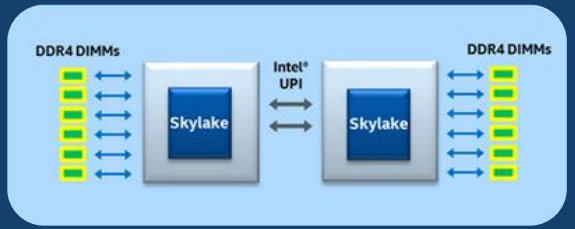
- Platform architecture in terms of networking?

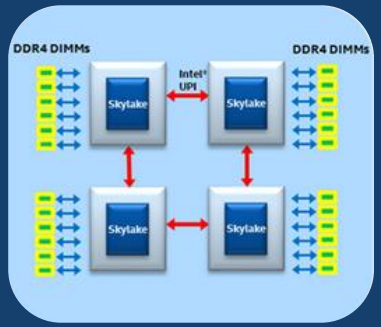
X9QRi-F Block Diagram

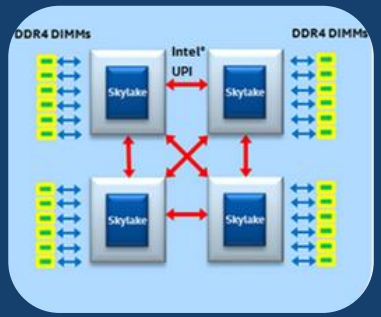


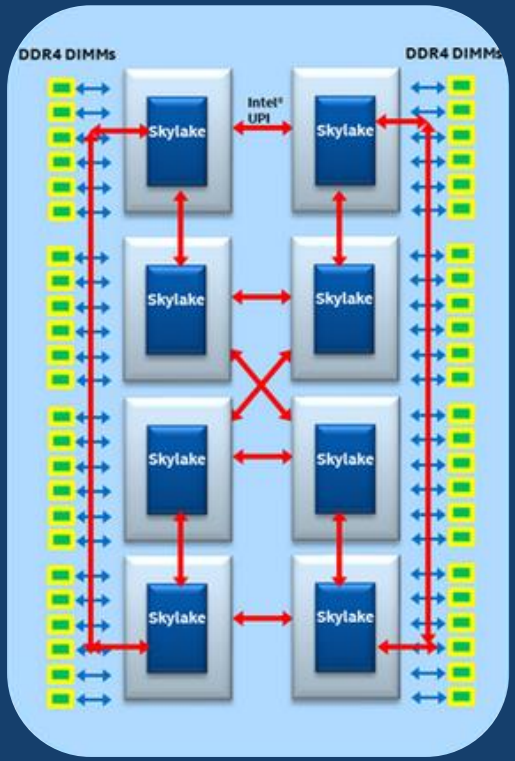
src: www.supermicro.com

NUMA









src: www.intel.com

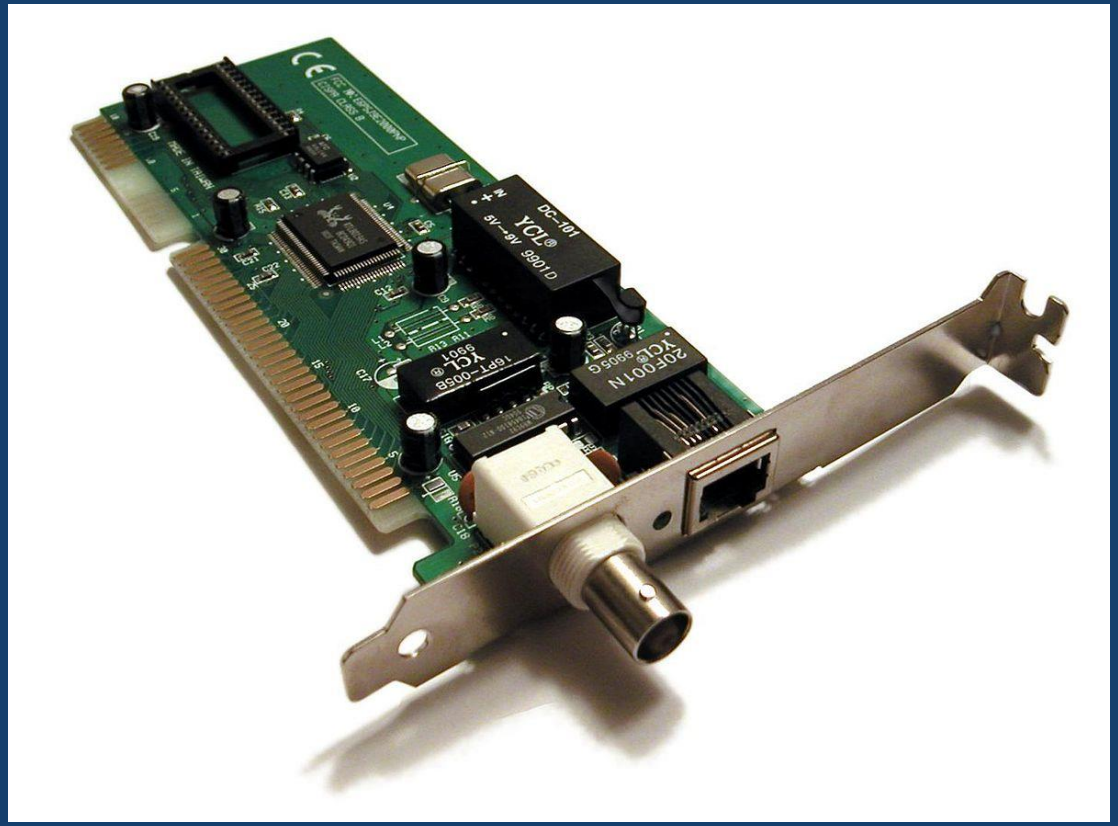
OVH.com

Performance

- PCIe 3.0
  - 8 GT/s = 7,877 Gbps (per signal-line)
  - 40 PCIe lines: 315,08 Gbps per CPU
- Memory [1]
  - 90 GB/s = 720 Gbps (DDR4, Haswell E5-2670v3)
  - *Latest CPUs: up to ~2.7 Tbps (EPYC) or ~3.3 Tbps (MCDRAM, Knights Landing) !*
- QPI/UPI/HT3 [2]
  - QPI: 9,6 GT/s * 16 bit = 153,6 Gbps (19,2 GB/s in both directions)
    - 2 QPI links in Skylake E5 and 3 links for E7
  - UPI: 10,4 GT/s (3 links in Skylake-SP Platinum and some Gold)
  - HT3.1: 6,4 GT/s * 16 bit = 102,4 Gbps (12,8 Gbps in both directions)

[1] src: https://www.karlrupp.net/2016/07/knights-landing-vs-knights-corner-haswell-ivy-bridge-and-sandy-bridge-stream-benchmark-results/ , www.amd.com
[2] src: www.hardwaresecrets.com, www.intel.com

Network Interface Cards (NICs)



src: www.wikipedia.org

OVH.com

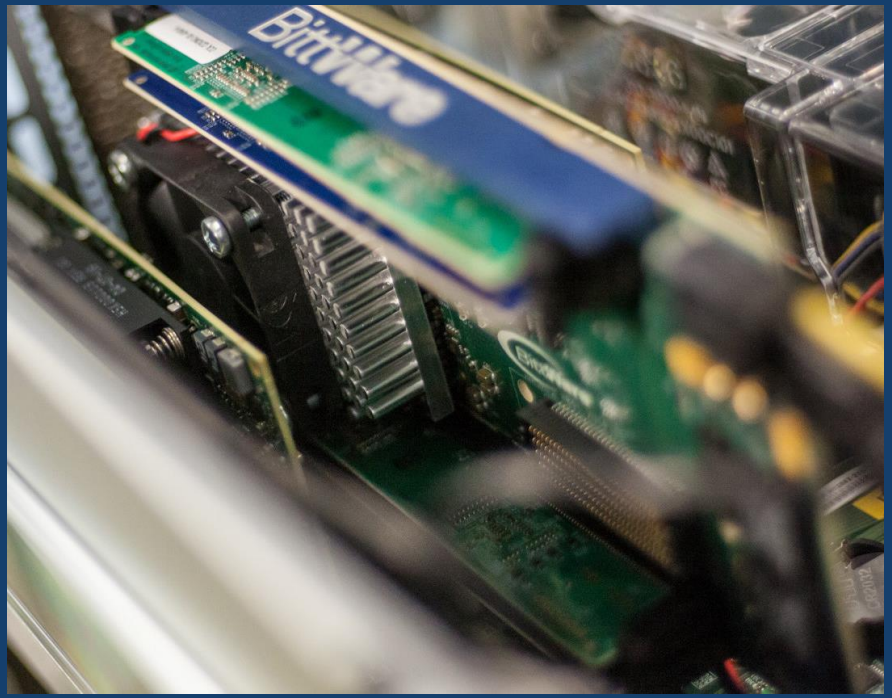
NICs

Ports:

- 1x, 2x, 4x (verify number of PCIe lines)
- 1, 10, 25, 40, 50, 100, 200 GbE

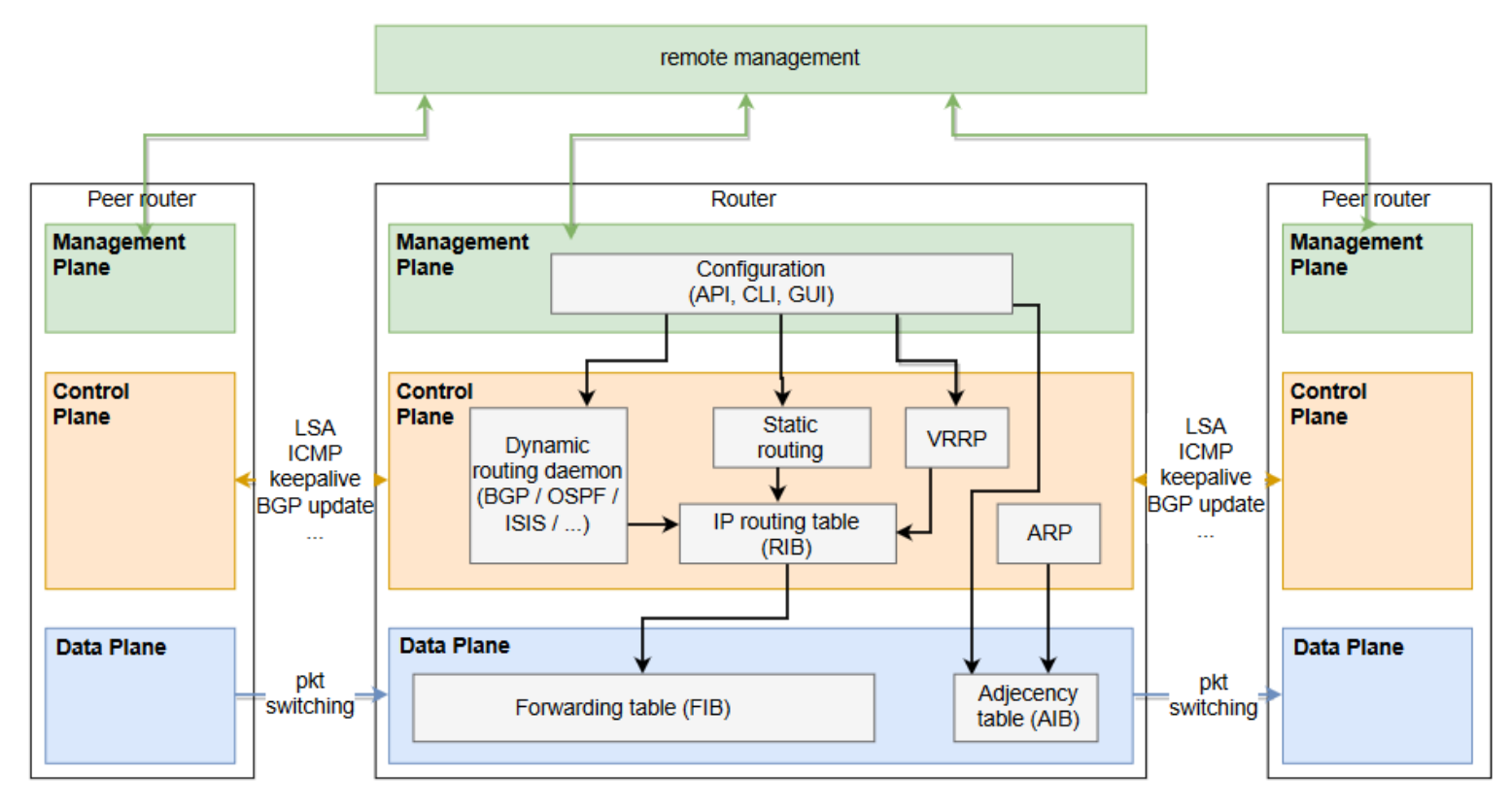
Types:

- NIC: up to 200 Mpps! [1]
- FPGA
- SmartNIC (FPGA or SoC) [2]
    - 50 Gbps @1000Bpp
    - 100 Gbps @500Bpp



[1] src: www.mellanox.com, Mellanox Connect-X6 dual 200GbE
[2] src: www.netronome.com, Agilio LX dual 40GbE and single 100GbE

Router – architecture



Data Plane

Linux kernel *

- Tx: 14,8 Mpps
- Rx: 12 Mpps (experimental patches) (eBPF drop: 9 Mpps in lab)
- IPv4 fw: 1 - 2 Mpps (on a single core) ☹

Kernel bypass

- Intel DPDK, netmap, PF_RING ZC, snabb, PFQ, …

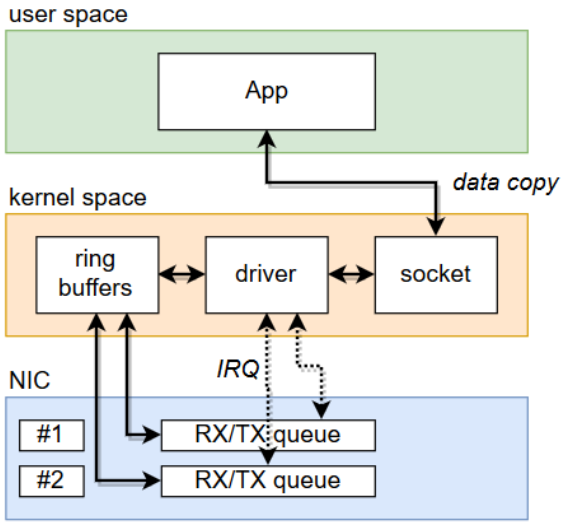
Delegate network functions – SmartNIC

- Offloads:
  tx, rx, checksum, lso, tso, QinQ, vxlan, …
- More:
  OpenVSwitch, eBPF, tunneling, LB, …

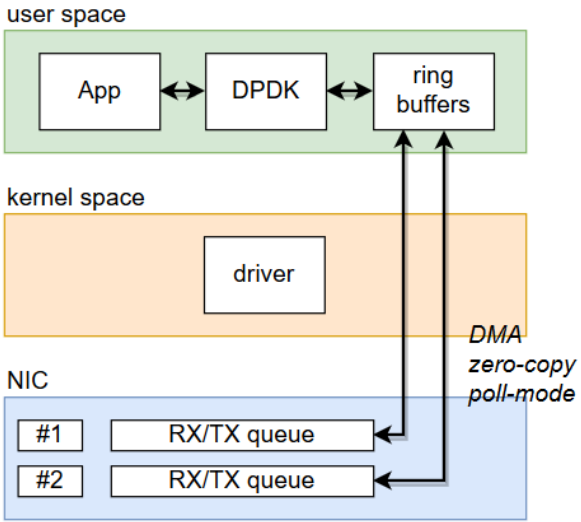
wire-speed		
	84 B	1500 B
10 Gbps	14,88 Mpps	0,83 Mpps
40 Gbps	59,5 Mpps	3,33 Mpps
200 Gbps	297,6 Mpps	16,66 Mpps

* src: Jesper Brouer @netdev1.1

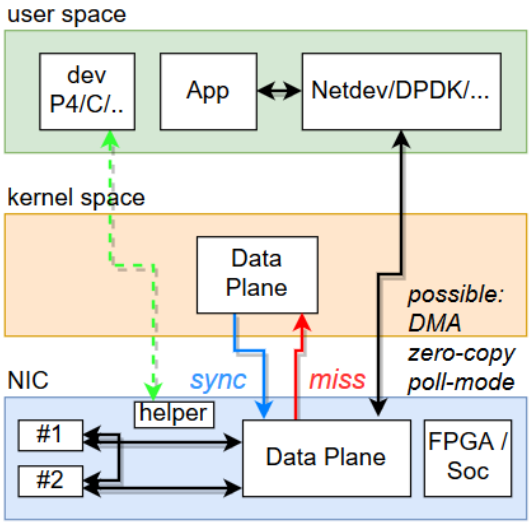
Linux kernel networking



Kernel bypass (DPDK)



SmartNIC



OVH vRouter

OVH.com

Functions and software

Router function definition

- BGP, VRF, DHCP relay, VRF
- Symmetric traffic (inbound/outbound)
- AntiDDoS modules
- Redundancy: VRRP, BGP
- Extra: IPSec z VRF

Software

- CP: Bird (multi-NUMA!), keepalived, ISC DHCP relay (patched), strongswan
- DP: 6Wind’s DPDK (more than 2 years of cooperation)

More: delivery, monitoring, management, ...

Hardware

- Intel Server Board S2600 (dualsocket, 4x PCIe x16, 2133 MHz ECC DDR4)
- CPU (Xeon E5)
  - 2667v3 (8c @3,6GHz)
  - 2687Wv4 (12c @3,5GHz)
  - 2650v4 (12c @2,9GHz)
  - …
- Mellanox Connect-X4 (40/100GbE)
- Intel X520 (10GbE)
- Dysk SSD (config, OS, logs buffer)

src: www.carztune.com

Basic tuning – don’t miss it!

OS / DPDK [1]

- Use all DRAM slots with at least 4GB, fastest supported by MB
- Memory size depends on supported protocols, interfaces, size of tables (routing, neighbors)
- Use supported NIC, Intel or Mellanox preferred
- Disable CPU power-saving and TurboBoost in BIOS
- Define the max CPU frequency instead of auto setting
- Disable virtualisation in BIOS if not needed
- Update the BIOS/NIC firmware!

[1] src: http://dpdk.org

CPU

- Core-Port allocation in DPDK/FastPath

```
: ${FP_MASK:=1-5,17-21}
: ${FP_PORTS:='0000:03:00.0 0000:03:00.1 0000:05:00.0 0000:05:00.1'}
: ${CORE_PORT_MAPPING:=c1=0/c2=0/c3=2/c4=2/c17=0/c18=0/c19=2/c20=2}
```

```
# numactl --hardware
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7 16 17 18 19 20 21 22 2
node 0 size: 64340 MB
node 0 free: 42428 MB
node 1 cpus: 8 9 10 11 12 13 14 15 24 25 26 27 28 2
node 1 size: 64495 MB
node 1 free: 61449 MB
node distances:
node   0   1
  0:  10  21
  1:  21  10
```

node	0	1	2	3	4	5	6	7
0:	10	12	17	17	19	19	19	19
1:	12	10	17	17	19	19	19	19
2:	17	17	10	12	19	19	19	19
3:	17	17	12	10	19	19	19	19
4:	19	19	19	19	10	12	17	17
5:	19	19	19	19	12	10	17	17
6:	19	19	19	19	17	17	10	12
7:	19	19	19	19	17	17	12	10

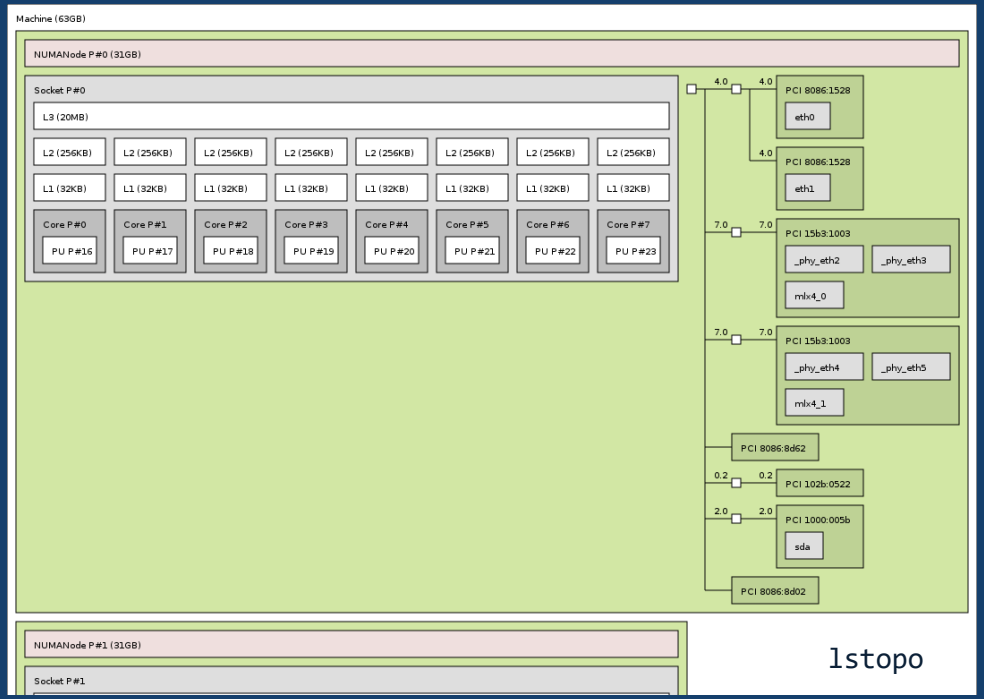
- Core allocation for apps (Control Plane per numa)

NUMA: tools

```
lstopo
dmidecode
lspci -tv
lscpu
hwloc-info -v
numactl

/proc/cpuinfo
/sys/bus/pci/devices/*/numa_node
/sys/devices/system/node/node*

mlc (intel memory latency checker)
```

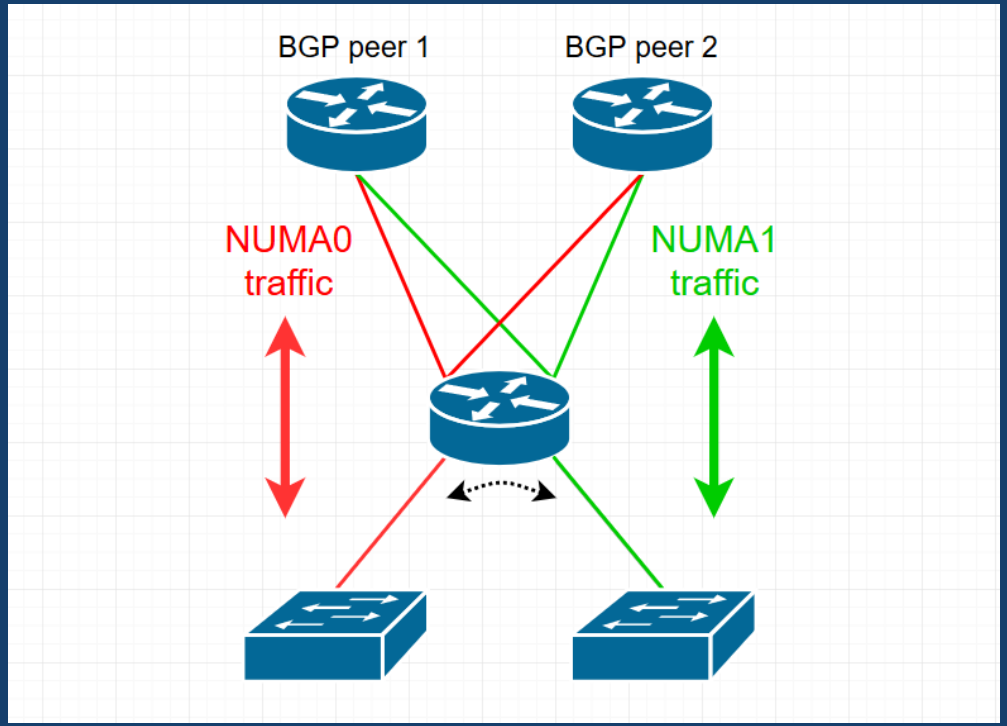


More:

https://www.open-mpi.org/projects/hwloc/tutorials/20140422-ComPAS-hwloc-tutorial.pdf

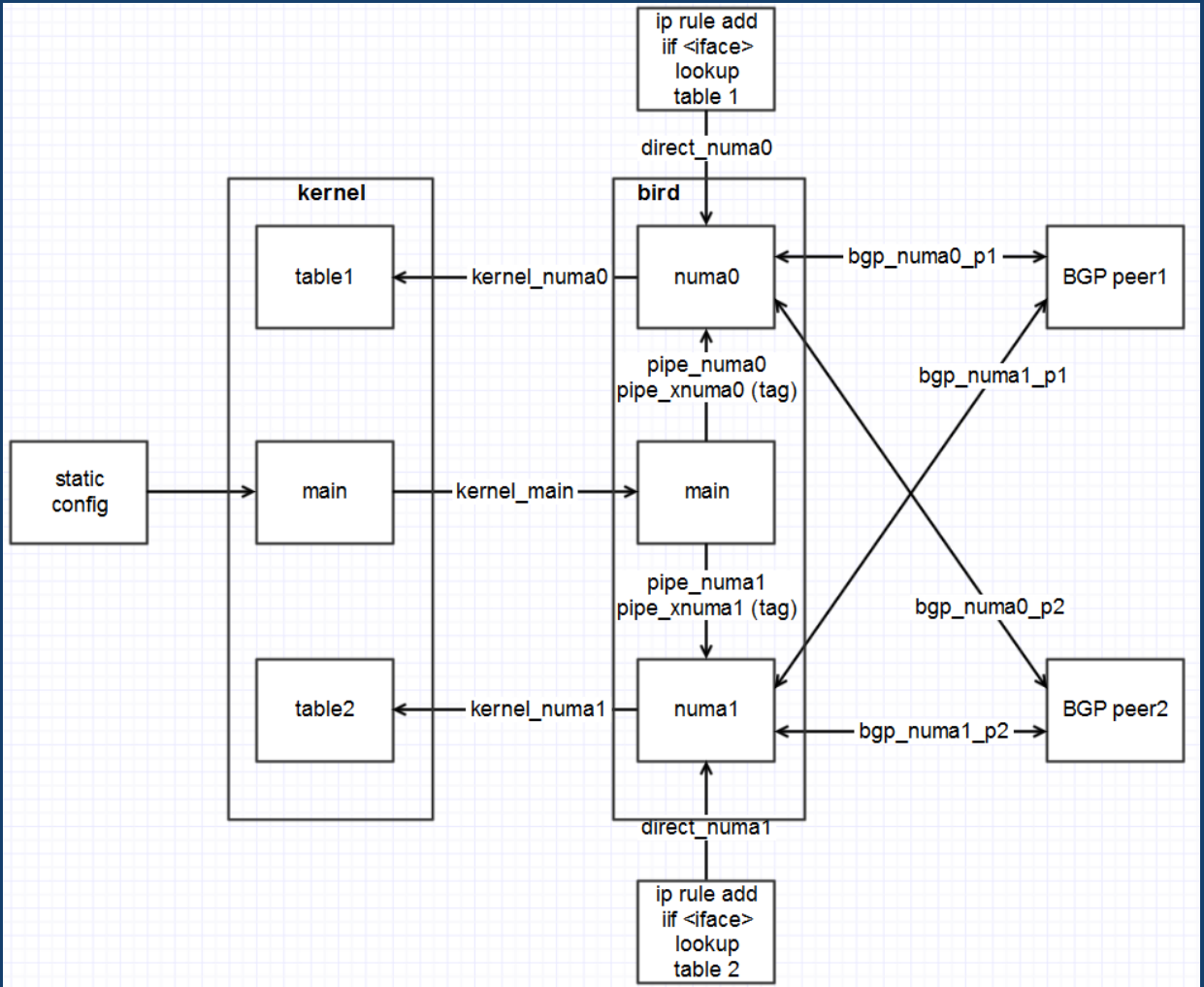
https://www.dcl.hpi.uni-potsdam.de/teaching/numasem/slides/NUMASem_Topology_discovery.pdf

NUMA



PBR for NUMA

- BIRD routing daemon
- Routing tables: `main / numa0 / numa1`
- Functions: `if_numa0() / if_numa1()`
- Control protocols:
    - direct_numa0/1: paths for direct-connected links (per-numa)
    - kernel_numa0/1/main: table sync Bird <> kernel
    - pipe_numa0/1: copy from main => numaX if_numaX()
    - pipe_xnuma0/1: copy from main => numa1/0 (cross numa) + tag (no BGP)
    - bgp_numa0/1_peer1/2: BGP sessions from numa0/1 for peer 1 and 2


ip rule add
iif <iface>
lookup
table 1
direct_numa0
kernel
bird
table1
kernel_numa0
numa0
bgp_numa0_p1
BGP peer1
pipe_numa0
pipe_xnuma0 (tag)
bgp_numa1_p1
static
config
main
kernel_main
main
pipe_numa1
pipe_xnuma1 (tag)
bgp_numa0_p2
table2
kernel_numa1
numa1
bgp_numa1_p2
BGP peer2
direct_numa1
ip rule add
iif <iface>
lookup
table 2

NUMA: CP – DP

Numa0: Control Plane

Numa1: Data Plane

Pros:

- Separate CPU for CP
- All CPU cores allocated for Data Plane
- Unequal-CPU?

Cons:

- All control traffic (incl. ICMP!) is a cross-NUMA traffic
- Unequal-CPU doesn’t (always) work (at least for S2600CW) ☹

NUMA: other ideas

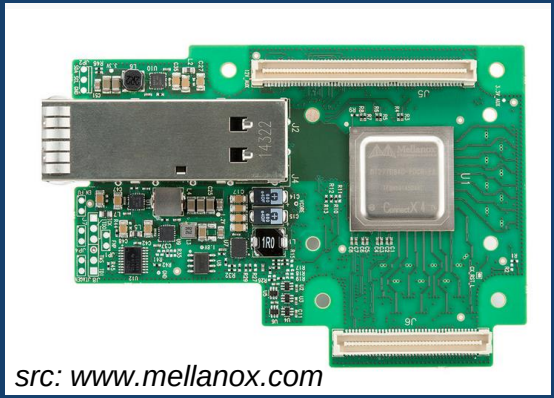
- Multi-host interfaces
- Breakout cables 40g->4x10G (compatibility!)
- NumaSCALE etc.
- Multi-CPU inside single NUMA (w00t?):
  AMD EPYC and Infinite Fabric

  AMD Epyc 7601:
  32c/64t @2.2-3.2 GHz,

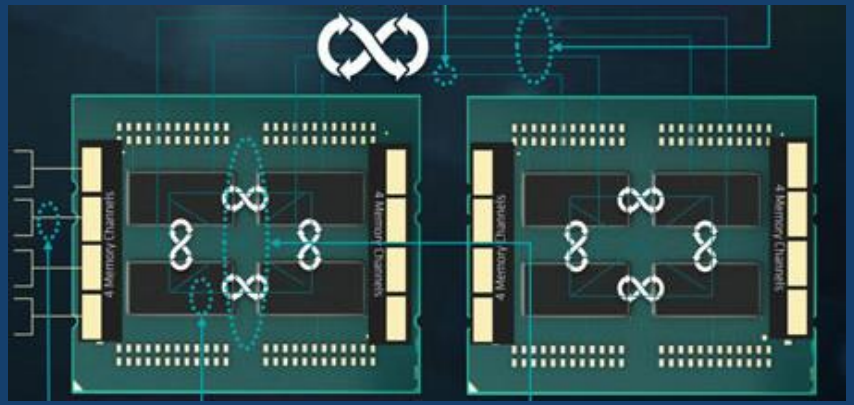
  64MB L3 cache, 2666MHz RAM

  ~4*300Gbps Infinite fabric

  128x PCIe lines



src: www.mellanox.com



src: www.amd.com

LAB tests:	8500 bpp		685bpp		85bpp	
ports	Gbps	Mpps	Gbps	Mpps	Gbps	Mpps
2x 2x10GbE Intel 82599	**30,0**	0,4	**28,0**	5,0	10,0	**15,0**
2x 40GbE MCX4	**80,0**	1,3	**78,0**	14,0	14,5	**22,0**
2x 100GbE MCX4	**172,0**	2,5	**138,0**	25,0	30,0	**40,0**

Summary

- Wisely choose hardware for the needs
- Sometimes 2x40 != 80 [Gbps]
- Proper software and it’s config may have strong impact
- NUMA allows you to scale better but don’t bet on auto-setup!
- Lack of CPU/core/ht numbering standard doesn’t help
- Ask your vendor for details
- Don’t trust single tool
- Be master of your hardware ☺

Thanks!

Questions?

@KubaAtOvh

meetup.com/Wroclaw-Net-IP-Meetup / netip.me

ovh.pl/jobs

OVH.com