10GE síťový prvek

switch & router

Radim Roška & Moris Bangoura

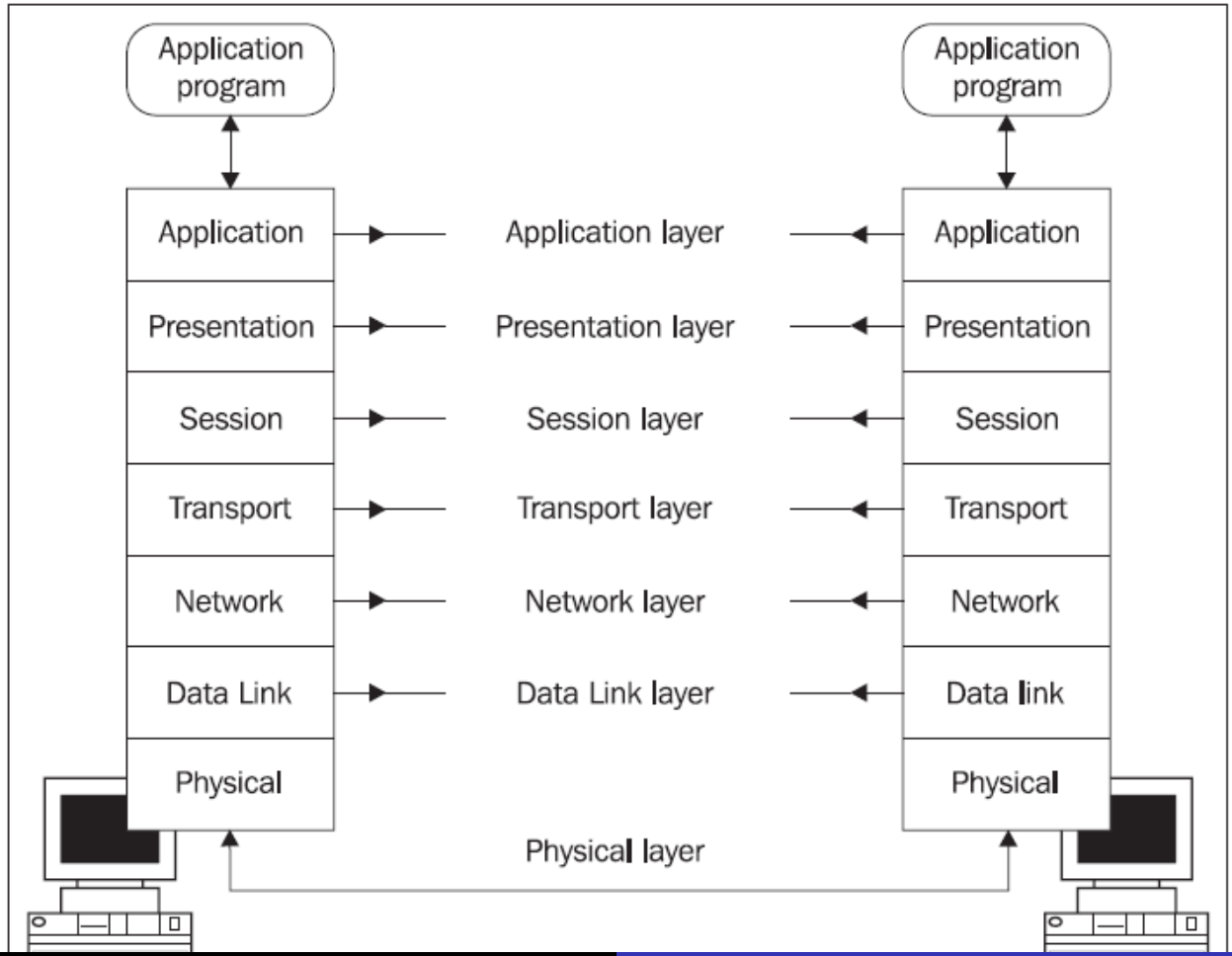
Installfest 2012
Silicon Hill

3.3. 2012

Obsah

Outline

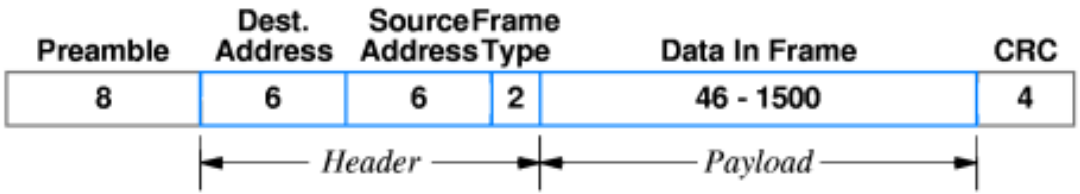
Povinný začátek

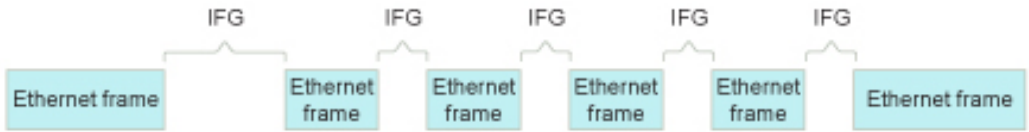


Layer 2

- L2 - MAC, error control, flow control, fragmentation
- Ethernet - rodina standardů pro LAN z cca 1980
- data unit - frame (rámec)
- MAC address (00:1d:7d 04:26:ca)

Frame





- IFG = 96 bit times (pro 10GE $96/10^{10} = 9.6ns$)
- overhead = 8B + 12B = 20B
- účinnost - podle velikosti rámce (64B: 76.1%, 1518B: 98.7%)

Komplikace při 10GE

- počet rámců za sekundu (teoretická propustnost) - *per packet delay* - počet přerušení
  - 64B rámce: 14880952 pps
  - 1518B rámce: 812723 pps
- objem dat - *per byte delay* - rychlost sběrnic,pamětí...
- např. interrupt live lock - zahlcení přerušeními

Outline

Co je to?

1. agregující síťový prvek, spojuje síťové segmenty
2. přepíná v rámce v rámci jednoho subnetu - 1 hop
3. CAM tabulka (content-addressable memory - asociativní paměť)

CAM tabulka

CAM Table

Station	Port1	Port2	Port3	Port4
00-00-3D-1F-11-01			X	
00-00-3D-1F-11-02				X
00-00-3D-1F-11-03	X			

Received Frame

Destination	Source	Data	CRC
00-00-3D-1F-11-05	00-00-3D-1F-11-01		

modprobe bridge

- flexibilní, funkční switch na linuxu
- STP
- L2 filtrování and shapování

Switch v Linuxu - konfigurace

```
linux-sw@root # brctl addbr br0
linux-sw@root # brctl addif br0 eth1
linux-sw@root # brctl addif br0 eth2
linux-sw@root # ip link set br0 up
```

SW řešení

- přerušení
- polling
- NAPI - adaptive interrupt coalescing (přerušení - nízká latence, polling - propustnost)
- softirq - rozdělení přerušení na kritickou a přerušitelnou část

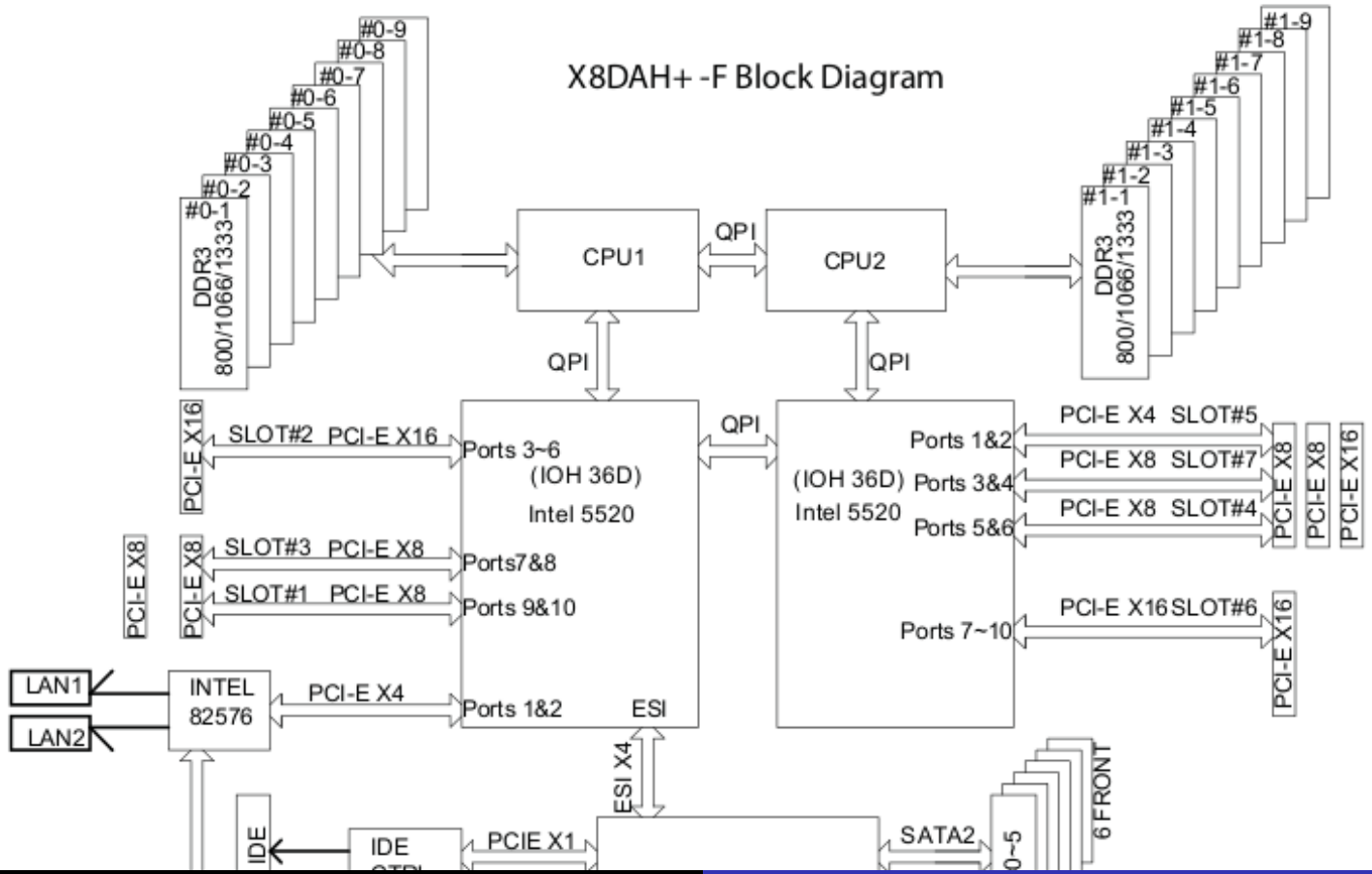
HW řešení

- více jader CPU
- více RX/TX front na síťových kartách a MSI-X (SMP affinity)
- NUMA
- PCI-Express (v2): 16x lanes = 16 * 500MB * 2 = 16GB/s (fullduplex)
- propustnost operačních pamětí: tripple channel DDR3-1333 = 3 * 10666MB/s = 32 GB/s
- QPI sběrnice: 25GB/s

HW řešení

- více jader CPU
- více RX/TX front na síťových kartách a MSI-X (SMP affinity)
- NUMA
- PCI-Express (v2): 16x lanes = 16 * 500MB * 2 = 16GB/s (fullduplex)
- propustnost operačních pamětí: tripple channel DDR3-1333 = 3 * 10666MB/s = 32 GB/s
- QPI sběrnice: 25GB/s

HW řešení


X8DAH+ -F Block Diagram
#0-9
#0-8
#0-7
#0-6
#0-5
#0-4
#0-3
#0-2
#0-1
DDR3 800/1066/1333
CPU1
QPI
CPU2
#1-9
#1-8
#1-7
#1-6
#1-5
#1-4
#1-3
#1-2
#1-1
DDR3 800/1066/1333
QPI
QPI
PCI-E X16
SLOT#2 PCI-E X16
Ports 3~6
(IOH 36D)
Intel 5520
QPI
(IOH 36D)
Intel 5520
Ports 1&2
Ports 3&4
Ports 5&6
PCI-E X4 SLOT#5
PCI-E X8 SLOT#7
PCI-E X8 SLOT#4
PCI-E X8
PCI-E X8
PCI-E X16
PCI-E X8
SLOT#3 PCI-E X8
Ports7&8
PCI-E X8
SLOT#1 PCI-E X8
Ports 9&10
Ports 7~10
PCI-E X16 SLOT#6
PCI-E X16
LAN1
LAN2
INTEL 82576
PCI-E X4
Ports 1&2
ESI
ESI X4
IDE
IDE
PCIE X1
SATA2
0~5
6 FRONT

10GE Síťová karta

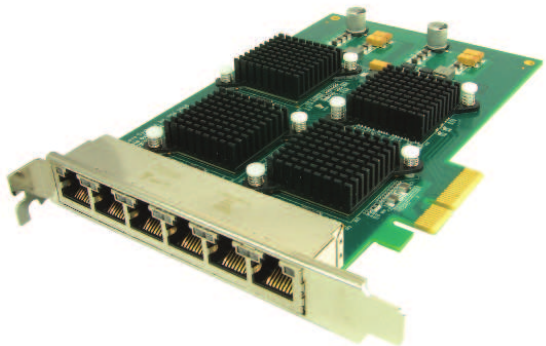
dual port 10GbE Intel controller 82599

- kvalitní chipset Intel 82599 s open source ovladačema
- MSI-X podpora (definuje až 2048 přerušení na zařízení)
- RSS (Received Side Scaling)
- offloading . . .
- PCI-Express 8x - dost na 2x10GE
- 128 TX i RX front

1GE Síťová karta

1GbE NIC s 6 portama a chipsetem Intel 82576

- 6 portů
- PCIe 4x



Jak měřit?

- RFC 2544
- 10GE měřáky
- GNU/Linuxové nástroje:
  - pktgen
  - netperf - jednoduchý nástroj na měření propustnosti
  - sar
  - mpstat
  - ethtool - detailní nastavení parametrů ovladače síťové karty
  - modprobe - různé parametry NIC ovladačů
  - modinfo - zobrazí nastavení ovladaču

pktgen

- kernel-space nástroj
- modul v kernelu
- velmi užitečný nástroj - dá se nastavit téměř vše (i generování s pomocí více jader na více front)
- ždímá výpočetní sílu CPU

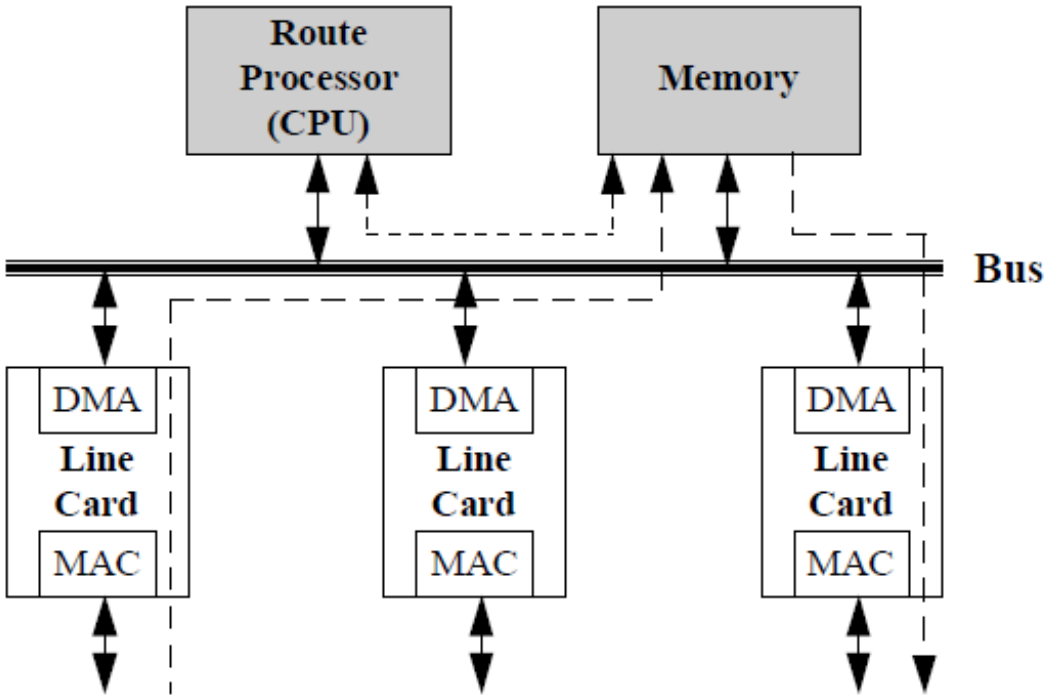
Outline

Co to je?

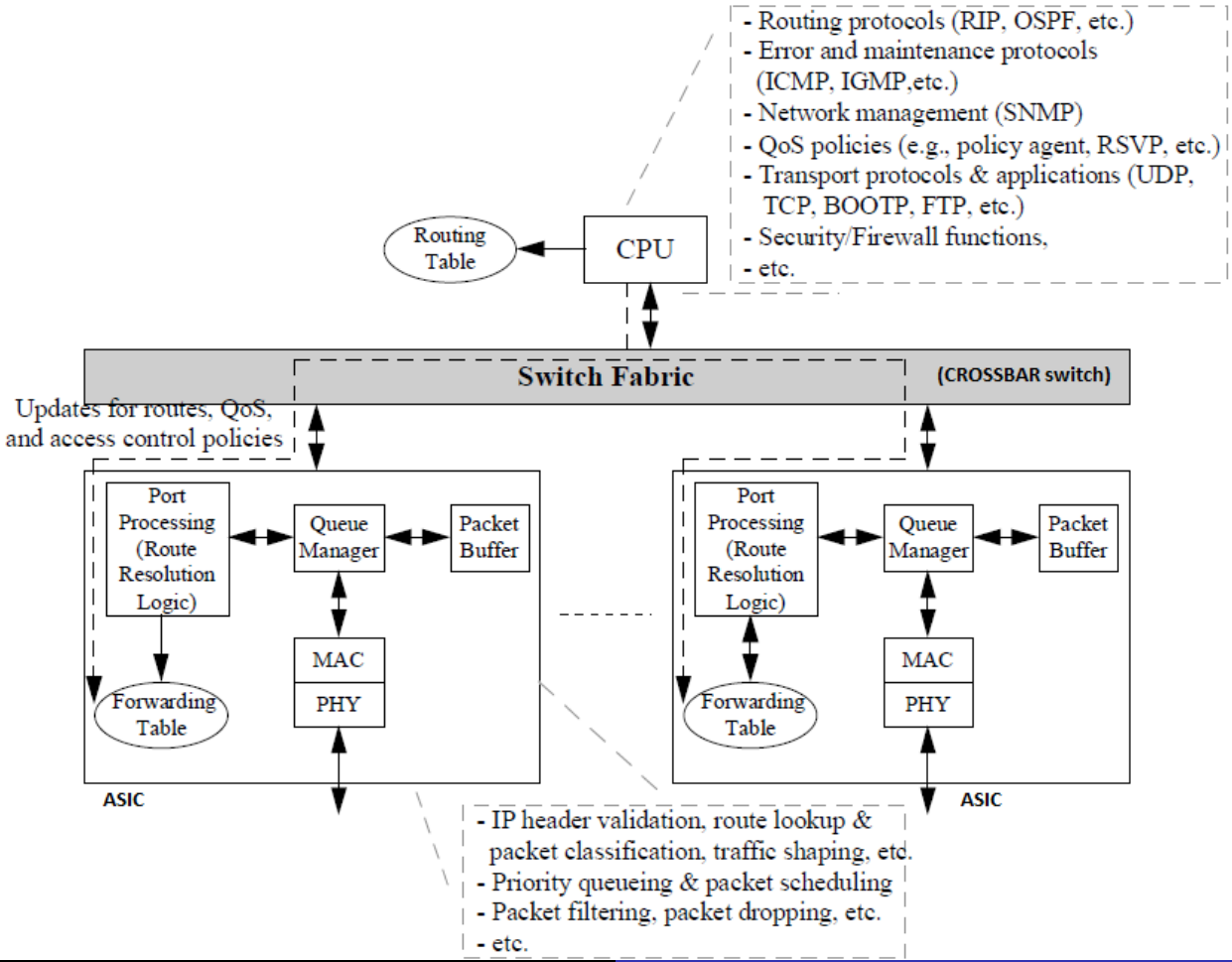
- aktivní síťová jednotka pracující s IP pakety
- směruje příchozí pakety k cíli (směrovací tabulka)
- rozhodování podle cílové IP adresy (Layer 3 OSI)

Outline

SW: NIC, Bus, CPU, RAM



HW: ASIC, TCAM, Crossbar Switch



SW?

Klady:

- linux :) Můžeme si hrát...
- jednoduše rozšířitelný (změnou programu)
- velká RAM (routovací tabulka... BGP)
- relativně levný

Zápory:

- vše zpracovává CPU (GPU)... no Tbps wirespeeds
- elektrická spotřeba?

Outline

Historie

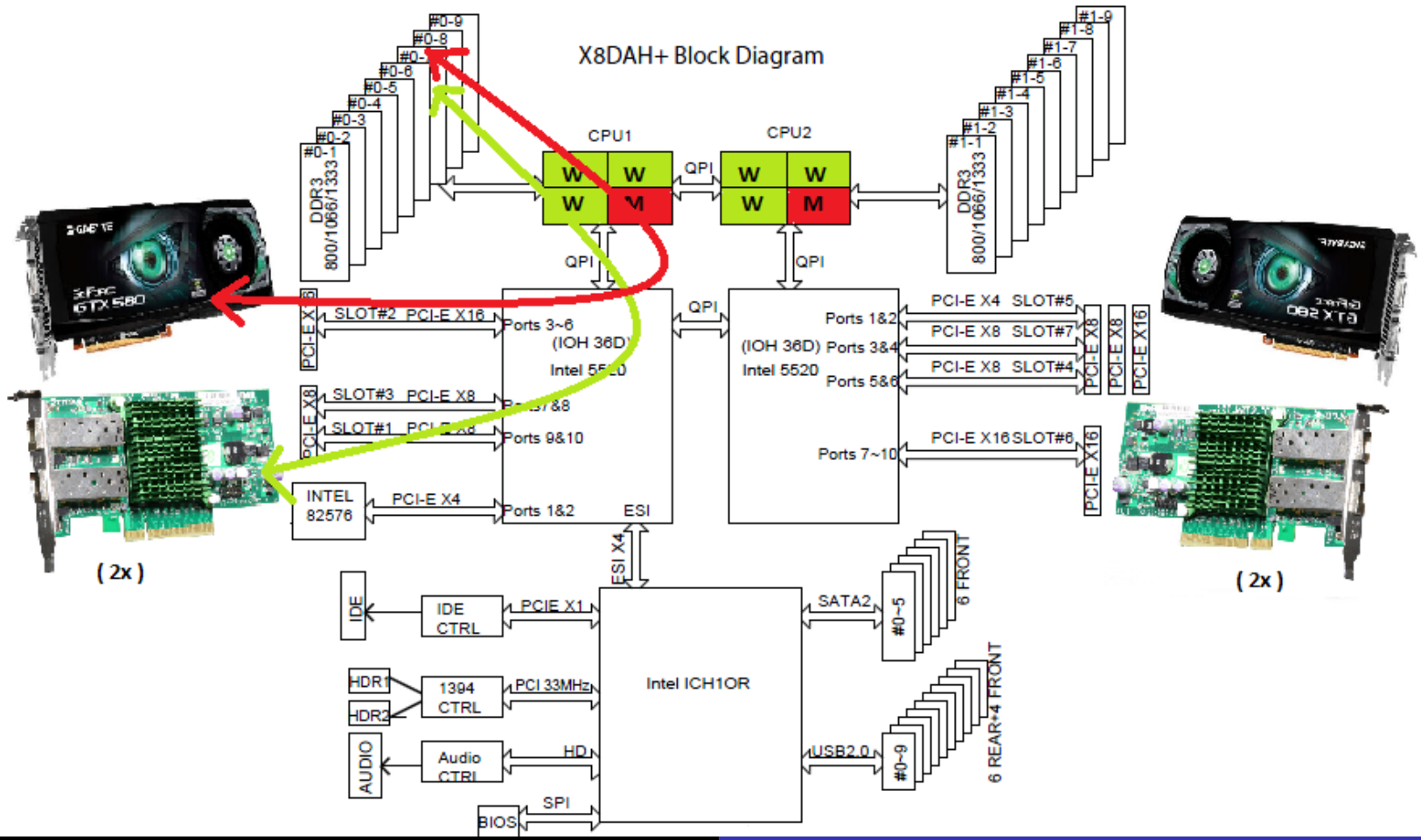
AGP/PCI arch.:

- FastEthernet (133/266MBps sdílená PCI sběrnice)

PCIe arch., vícejádrové CPU, QPI/Hypertransport:

- SMP PCIe: GigabitEthernet (1 PCIe HUB, <32 PCIe linek)
- NUMA PCIe: 10GigabitEthernet (2 PCIe HUBy, 64 PCIe linek)

Blokové schéma PCIe arch. - Intel 5520 NUMA



10GbE SW router projekty

10GbE SW Routing:

- Bifrost (tuned Linux Distro): http://bifrost.slu.se/
- Click – Modular Router (kernelspace/userspace): http://read.cs.ucla.edu/click/click
- Packet Shader Framework (GPU Accelerated): http://shader.kaist.edu/packetshader/

Outline

Routing na CPU - omezení?

Počet CPU cyklů na 1 packet:

- cca. 1200 CPU cyklů pro forw. paketu z jednoho interface na druhý):
- to je hodně i pro multicore CPU (chceme 14.88 Mpps)
- nezbývá vypočetní výkon na routing a další funcke SW routeru (GPU?)

Šířka pásma při zpracování paketu (PCIe 2,0):

- PCIe: OK (8 linek na 2x10GbE NIC = 4GBps)
- QPI: OK (25,6GBps)
- Tripple Channel DDR: OK (32 GBps)

Akcelerace pomocí GPU

Proč GPU?

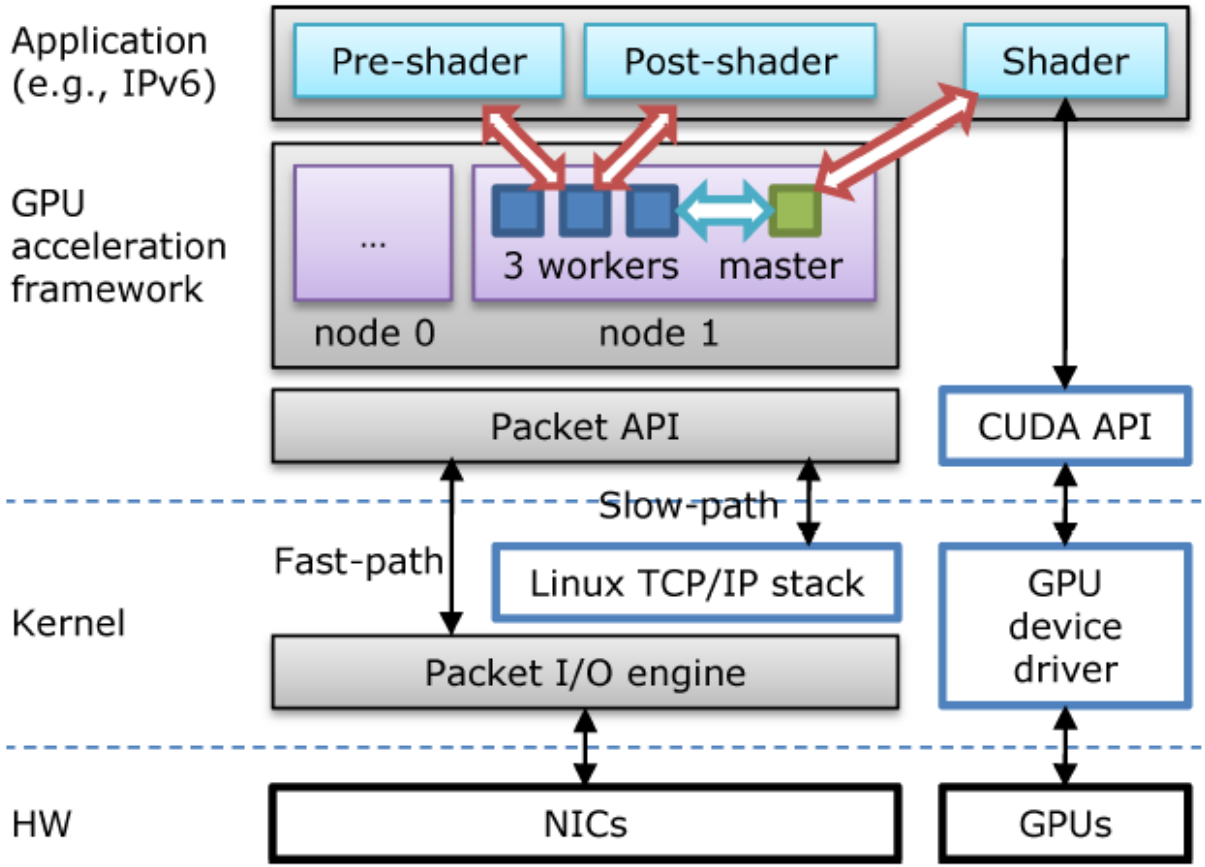
- vysoká šířka pásma (177GBps GTX 580 vs. 32GBps Xeon 5550)
- nízká latence
- paralelních zpracování SIMT (512 vláken)
- dostupnost, nízká cena
- userspace. . . CUDA / OpenCL :-/

Packet Shader Framework

GPU akcelerovaný SW Router, 40Gbps při 64B paketech a full BGP routing table.
Součásti:

- Packet Shader I/O modul
- Packet Shader Userspace aplikace

PS Framework - schéma

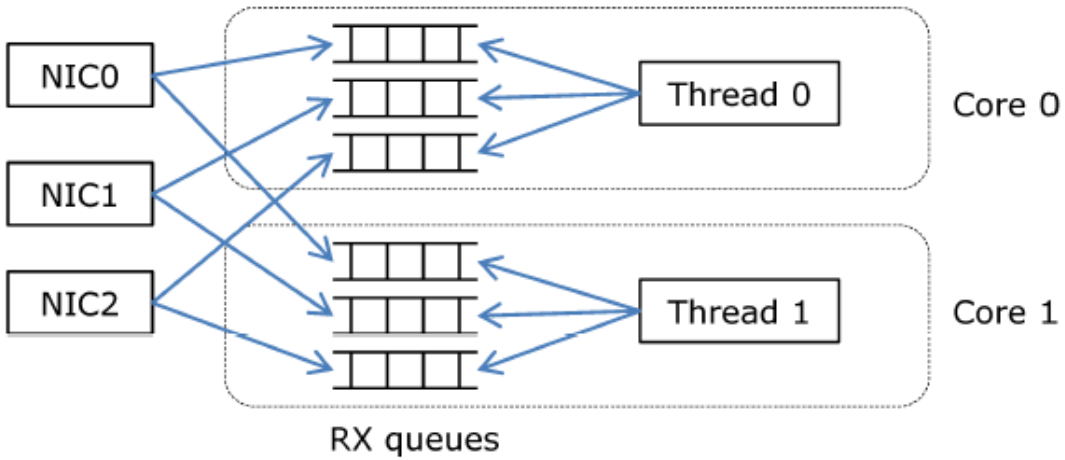


Packet Shader: I/O

Přepsaný optimalizovaný Intel 82599 ixgbe driver (1200 -> 200 CPU cyklů):

- propojení mezi kernelem a uživatelskou aplikací
- huge packet buff: nealokuje linux skb buffer pro každý paket, jsou 2 velké STATICKÉ kruhové buffery (metadata and data)
- BATCH processing: dávkové zpracování paketů (NIC, PCIe, userspace) – 32, 64 chunks najednou
- CPU AFFINITY: RX a TX fronty spjaty s každým worker jádrem, per queue counters, 64B zarovnání paměti
- vyhýbáme se NUMA “node corssing” ! (60% výkon dolů)
- automatic hybrid interrupt/polled packet RX

Per fronta CPU Affinity, Receive Side Scalling (RSS)



- RSS: src/dst IP/port + proto -> zachovává pořadí per flow

Packet Shader: userspace

Multivláknová userspace aplikace:

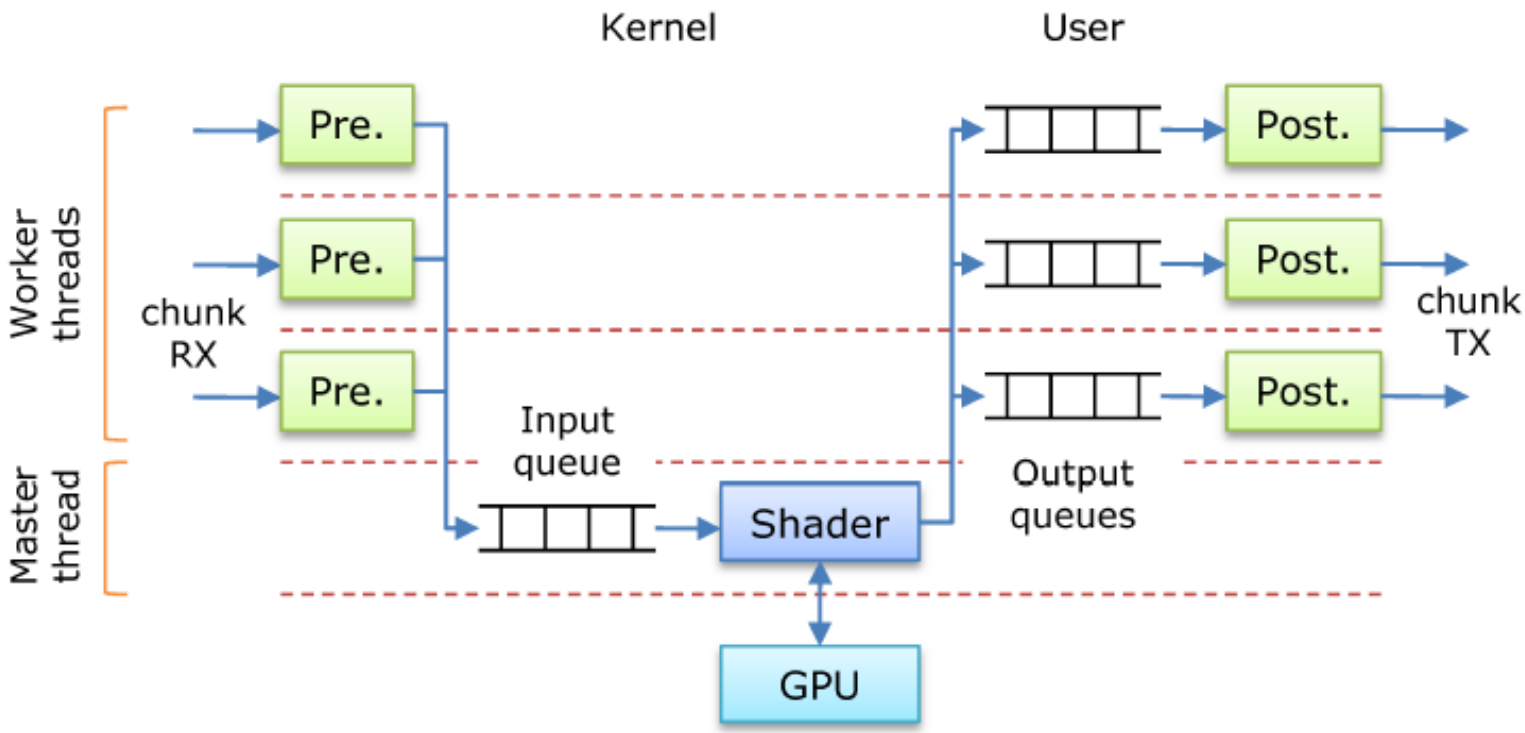
- worker vlákno: sběr paketů z I/O
- master thread: komunikace s GPU
- komunikace mezi vlákny via fronty (mutexy)
- optimalizace: zero-copy, pipelining (worker), gather-scatter + concurrent copy-execution (master)

Packet Shader: fáze

3 fáze:

- pre-shade (worker RX... meta -> master inp.q.)
- shade (GPU route lookup... output meta -> worker outp.q.)
- post-shade (worker TX)

Cesta paketu PS, fáze



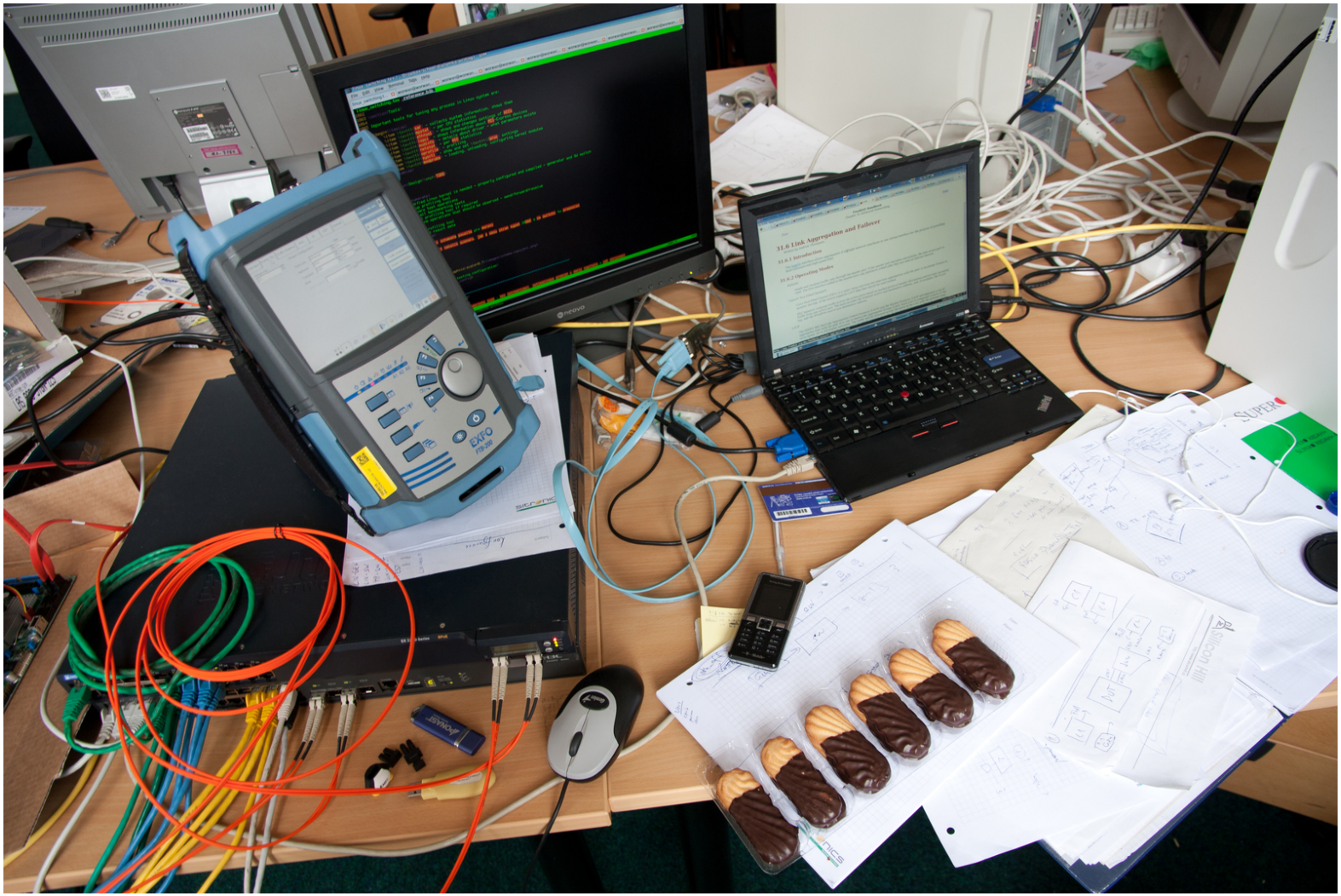
Outline

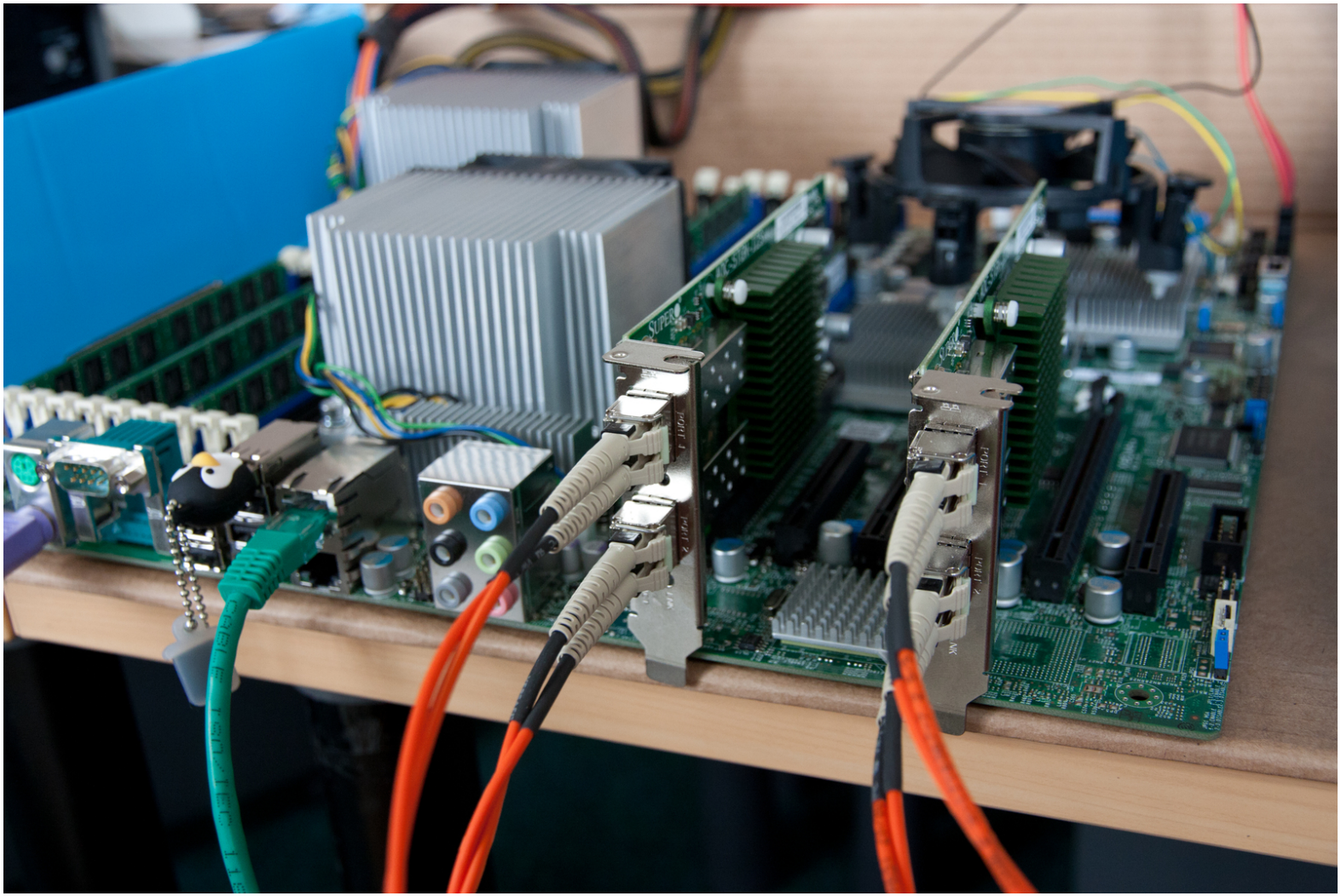
10GbE na DSN

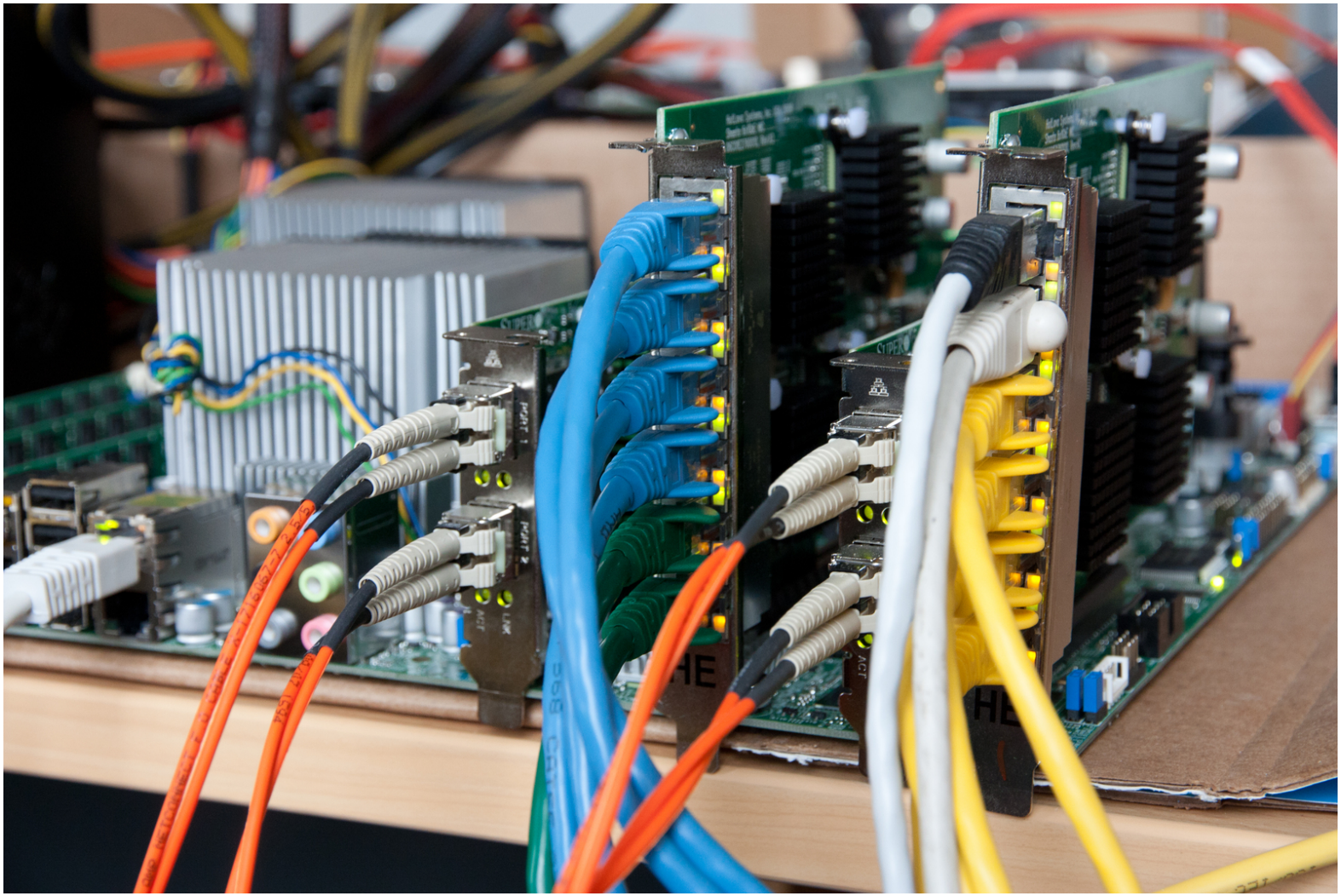
Projekt ČVUT FEL, CESNETu.

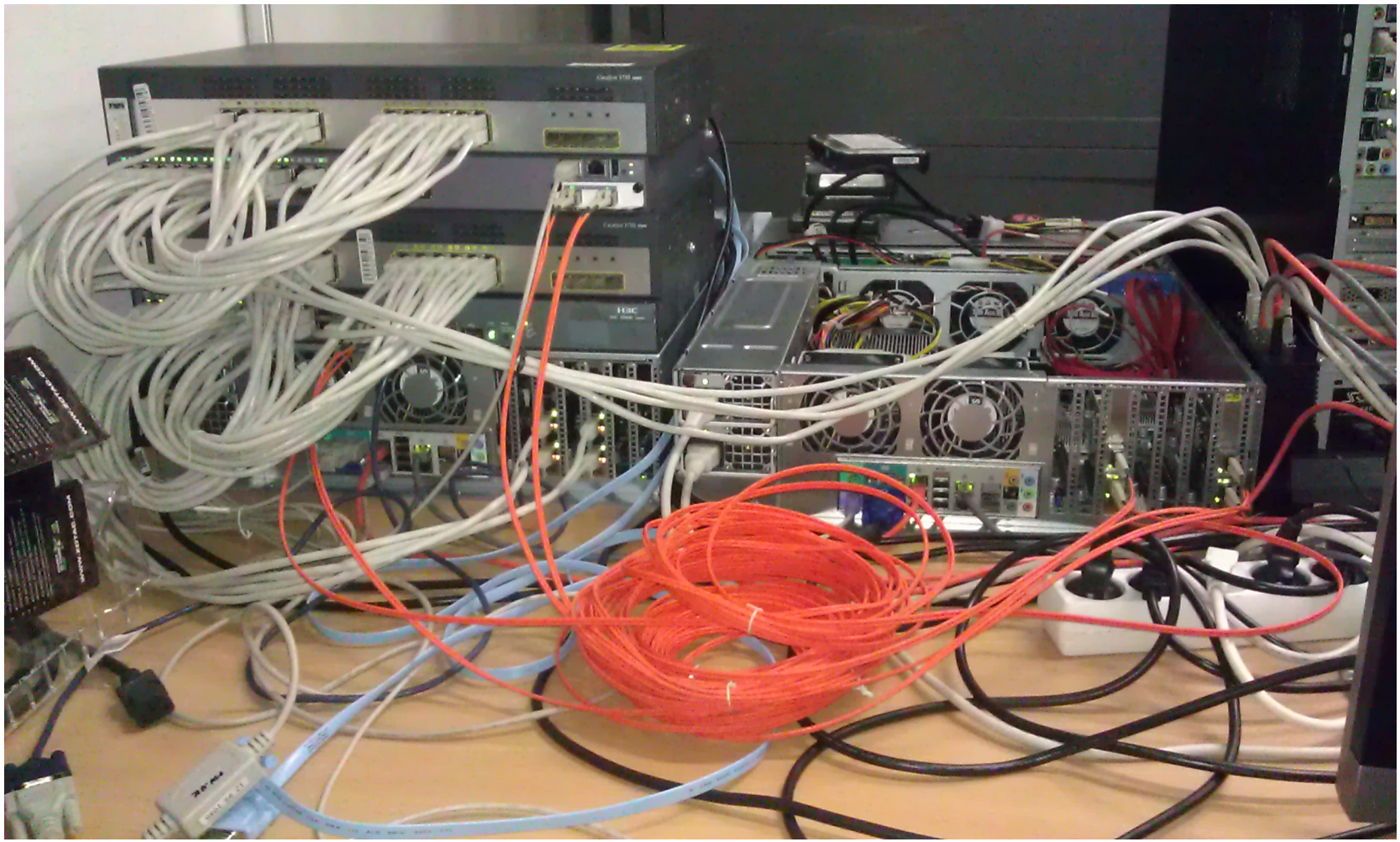
- nejsou sources pro PS userspace část: píšeme znova
- propojení z dynamickým routovacím daemonem: XORP (via click socket)
- GPU firewalling (ACL like)
- . . . a další :)
- https://dsn.felk.cvut.cz/wiki/projekty/10gbeth

Outline









Q/A

?