Status from Martin Josefsson
Hardware level performance of 82599

Bifrost Workshop 2010
d.27/1-2010

Testlab and Data provide
by Martin Josefsson
Procera Networks

by
Jesper Dangaard Brouer <jdb@comx.dk>
Master of Computer Science
ComX Networks A/S

Background

- Martin Josefsson (aka Gandalf)
  - Could not attend Bifrost Workshop 2010
    - Thus, I'm giving a resume of his findings
- Tried to determine
  - the hardware performance limits of the 82599 chip
  - Found some chipset limitation
- Hardware shows (82599 + Nehalem)
  - Very close to wirespeed forwarding capabilities
    - But partial cacheline PCIe transfers drops perf

Test setup

- Dell R710:
- CPU 1 x E5520
  - 4 cores 2.26GHz, QPI 5.86GT/s, 1066MHz DDR3
  - Hyperthreading enabled improve by ~20%.
- NIC: 2 x Intel X520-DA: 2 port
  - single 82599ES, using pci-e x8 gen2
- Traffic generator:
  - Smartbits with two 10GE modules.

Operating System: PLOS

- PLOS – is Procera Networks owns OS
  - 1 thread runs Linux, 7 threads run PLOS
- small standalone kernel
  - designed for packet processing *only* (no userspace)
- uses a channel concept
  - each channel, assigned two interfaces
  - packets are forwarded between the interfaces
  - giving a 1:1 mapping of RX and TX interfaces
  - (details in bonus slides)

Dumb forwarding: HW level

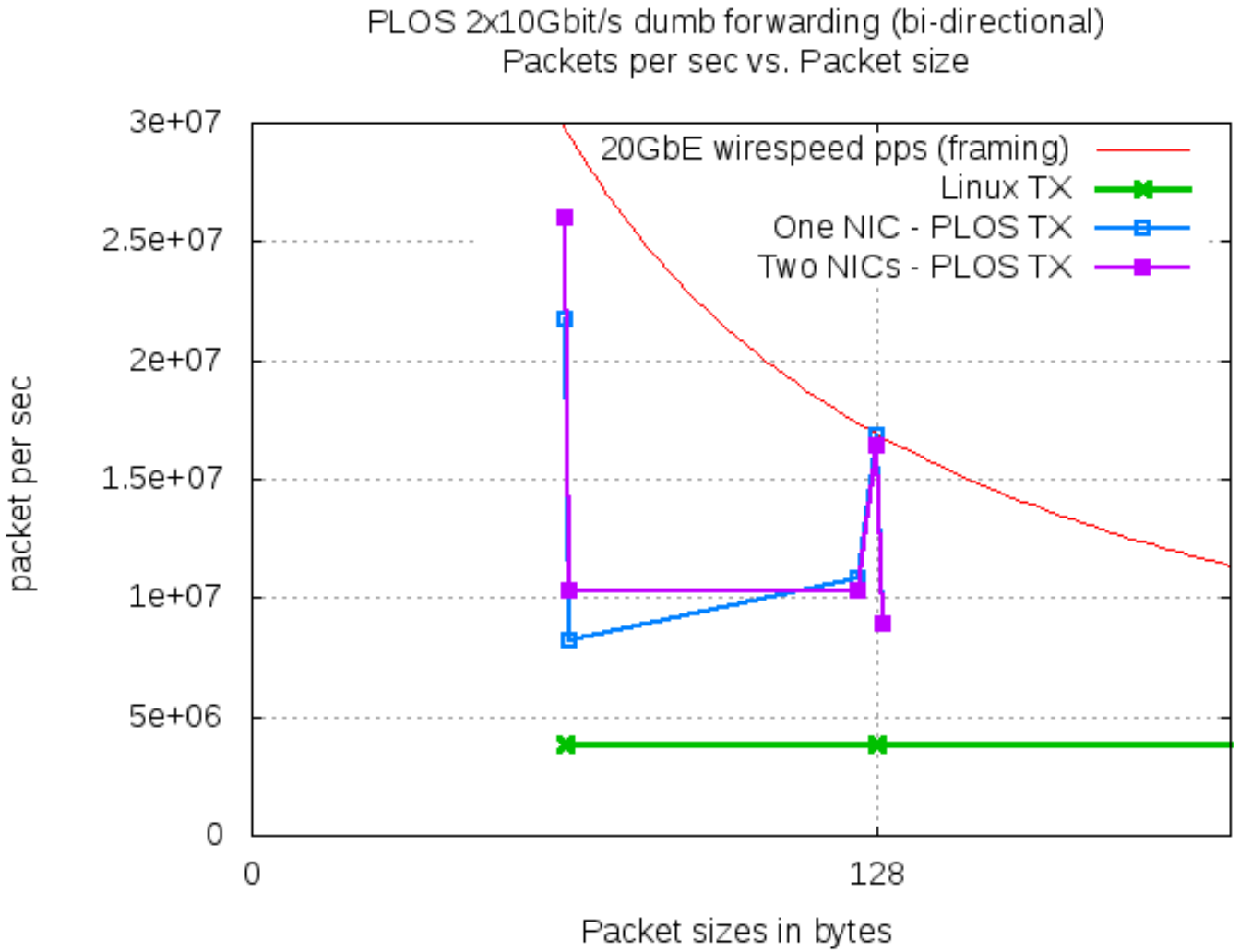
- Test results/setup
  - Very dumb forwarding of packets
  - Purpose: **find the hardware limits**
  - The packet data is not even touched by the CPUs
- Tests are **bidirectional**
- Results counts
  - Forwarded TX packets per sec
  - From **both** interfaces

Test results

- Two tests
  - One NIC: Using two ports on one dual port NIC
  - Two NICS: Using two NICs on port on each

	Million Packets Per Second		
Packet size bytes	*One NIC*	*Two NICs*	**2xWirespeed**
64	21.7 Mpps	26 Mpps	29.76
65	8.2 Mpps	10.3 Mpps	29.41
124	10.8 Mpps	10.3 Mpps	17.36
128	16.1 Mpps	16.4 Mpps	16.89
129	8.9 Mpps	8.9 Mpps	16.78

Test results: Graph



Issue: Partial cacheline PCIe

- The cause of the performance hit:
- According to Intel there is an
  - issue with partial cacheline pci-e
  - transfers in the Tylersburg (X58, 5500, 5520) chipset.
- Each partial cacheline transfer
  - uses one outstanding pci-e transfer
  - and one outstanding QPI transfer.

Workaround?

- No solution at the moment to this issue :(
- A possible workaround
  - get the 82599 to pad pci-e transfers
  - to a multiple of the cacheline size
  - but unfortunately it doesn't have this capability

Close to Wirespeed

- A single port on an 82599
  - Can't RX more than around 13.8 Mpps at 64 bytes
    - Intel verified this
  - Ethernet wirespeed is 14.8 Mpps at 64 bytes
- Which I find quite impressive
  - Also compared to other 10GbE NICs
  - And compared to peek 5.7 Mpps by Linux

Conclusion

- The Nehalem architecture
  - a massive improvement over the old Intel arch
  - still issues that impacts performance greatly
- The 82599 NIC
  - Comes very close to wirespeed
    - Parial cacheline PCI-e hurts badly!
  - Numbers are still impressive
    - compared to Linux
      - room for optimizations!

Bonus slides

- After this slide is some bonus slides
  - With extra details...

Bonus#1: locking of RX queue

- Each interface
  - a single RX-queue and 7 TX-queues in tests
- Each PLOS thread locks the RX-queue
  - grabs a bunch of packets,
  - unlocks the RX-queue lock
  - forwards the packets
    - to one of the TX-queues of the destination interface

Bonus#2: locking of RX queue

- RX is serialized by a spin-lock (or trylock)
  - threads continue to try to lock the next interface
    - RX-queue if the current one is already locked
- This is clearly not optimal
  - RSS and DCA will be investigated in the future.
- TX is not serialized
  - but cases where serializing TX actually helps

Strange PCIe x4 experience

- Seems, get higher performance
  - with none cacheline sized packets
  - when the 82599 is running at pci-e 4x gen2.
- Channel with 2 ports on the same 82599ES:
  - pci-e x8 and 65 byte packets: 8.2 Mpps
  - pci-e x4 and 65 byte packets: 11.2 Mpps
  - I can't explain this, feels like this could have something
  - to do with either pci-e latency
  - or with how many outstanding pci-e transfers
    - BIOS has configured for the pci-e port