Report from Netconf 2009

Bifrost Workshop 2010
d.27/1-2010

by

Jesper Dangaard Brouer <hawk@comx.dk>
Master of Computer Science
Linux Kernel Developer
ComX Networks A/S

ComX Networks A/S

What is netconf

- Summit of the Linux network developers
  - invitation-only
    - main maintainers and
    - developers of the Linux networking subsystem
- In 2009
  - Held in: USA, Oregon, Troutdale
  - Dates: 18/9 to 20/9-2009

How to get invited

- Basics:
- Know: David Stephen Miller
  - Top1 comitter
  - Top3 Sign-off'er
  - Maintainer of
    - netdev
    - sparc
    - IDE
  - Postmaster
  - Don't piss him off
  - NHL: San Jose Sharks
    - Don't send patches if they lost ;-)

Homepage

- http://vger.kernel.org/netconf2009.html
  - first day, played golf (Linus didn't show up)
  - Paul E. McKenney (RCU inventor)
    - made really good summaries
    - http://vger.kernel.org/netconf2009_notes_1.html
    - http://vger.kernel.org/netconf2009_day2.html

Its all Roberts fault!

- Paul E. McKenney
  - presentation on RCU
  - Funny slide: "its all Robert Olssons fault"
    - why we have RCU_bh()
    - (show his slide)

Intel and 10GbE

- Intel,three optimizations, all needed, for perf boost.
  - NUMA aware buffer allocation,
  - buffer alignment (to cache lines) and
  - removing of all references to shared vars in driver
- Performance peak of 5.7 Mpps (packets per sec)
  - Nehalem CPUs 2x, 2.53 Ghz, DDR3-1066MHz
  - 1x Dual Port Niantic/82599, 8 queue pairs per port
- Stock NUMA system (no optimization): peak at 1.5 Mpps
  - NUMA hurt performance, without NUMA aware allocs
- My 10GbE test: peak of 3.8 Mpps
  - single CPU Core i7-920 (DDR3-1666Mhz)

Multiqueue

- Linux Network stack scales with number of CPUs
  - Only for NIC with multi-hardware queues
    - Almost all 10G NIC
    - For 1GbE recommend Intel 82576
  - SW: Avoid locking and cache misses across CPUs
  - TX path: DaveM's multiqueue TX qdisc hack
    - now the default queue
- Each hardware queue (both RX and TX) has their own IRQ.
  - multiqueue NIC, lot of IRQs.
  - Try looking in /proc/interrupts

Traffic Control vs. Multiqueue

- Problem: Advanced Traffic Control
  - Kills multiqueue scalability
  - TX path will stop to scale
    - and returns to single CPU scaling
- Possible solution by: Paul E. McKenny
  - simply having "pools" per CPU (of e.g. tokens),
  - only contact other CPUs when the local pool empty
  - (idea taken from how NUMA mem manager works)

The End

- Better stop here
  - Have not covered everything
    - Look at the website:
      - http://vger.kernel.org/netconf2009.html

- Thanks for listening
  - even though I used too much time...