40 Gigabit Ethernet and Beyond: Challenges of End-to-End Communication

Cosmin Dumitru
c.dumitru@uva.nl

University of Amsterdam
System and Network Engineering Research Group

31 March 2010
GNARP 2011

Motivation: Beyond 10Gbit/s

- Infiniband Quad Data Rate already available (DAS4 has it!)
  - latency and throughput issues 'fixed' by bypassing the TCP/IP stack - zero copy
  - libsdp and IPoIB allow transparent transport of IP apps over Infiniband
  - Range limited to the cluster/data center (10G gateways/extenders)
- Multiple bonded 10Gigabit Ethernet links
  - 'commodity' interconnect
  - Bonding is done at MAC Layer
  - 10Gbit/s max per IP flow
  - CPU usage can be a problem

40Gigabit Ethernet

- Multiple transport media - fiber & copper (1m-10KM)
- 802.3ba – 40/100GE standard released in June 2010
- Keeps the same Ethernet frame format
- Client cards announced in September 2009
- Switches announced in Q2 2010 (Force10, Extreme Networks, Broadcom, etc.)
- 'multi-core for networks'
  - Multi Lane Distribution - 4 x 10Gbit
  - Synchronizing done at the PHY Level

40GE Gear

Mellanox ConnectX2 40GE NIC
PCI-E 2.0 x8

12 pair fiber cable w/ QSFP+ optics

Expected performance

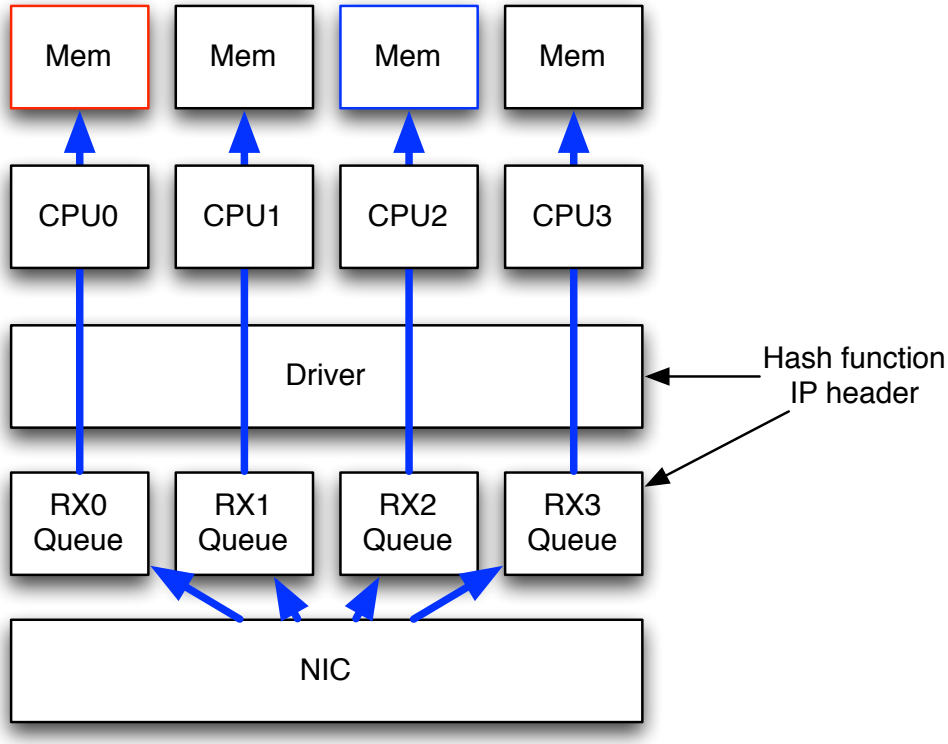
- PCI-E x8 = 40Gbit/s signaling rate but 32Gbit/s data rate (8/10 bit encoding)
- PCI-E protocol adds extra overhead
- 'Traditional' IP/Ethernet overhead
- 27Gbit/s is a realistic maximum

Early Experience with 40GE

- UvA, SurfNET, VU, Ciena, Mellanox
- GLIF2010 demo - the world's first 40GE transmission over WAN (1650KM)
- Sustained throughput of 25Gbit/s with *enough* IP flows (measured at application level)
- Single IP flow unable to saturate the NIC

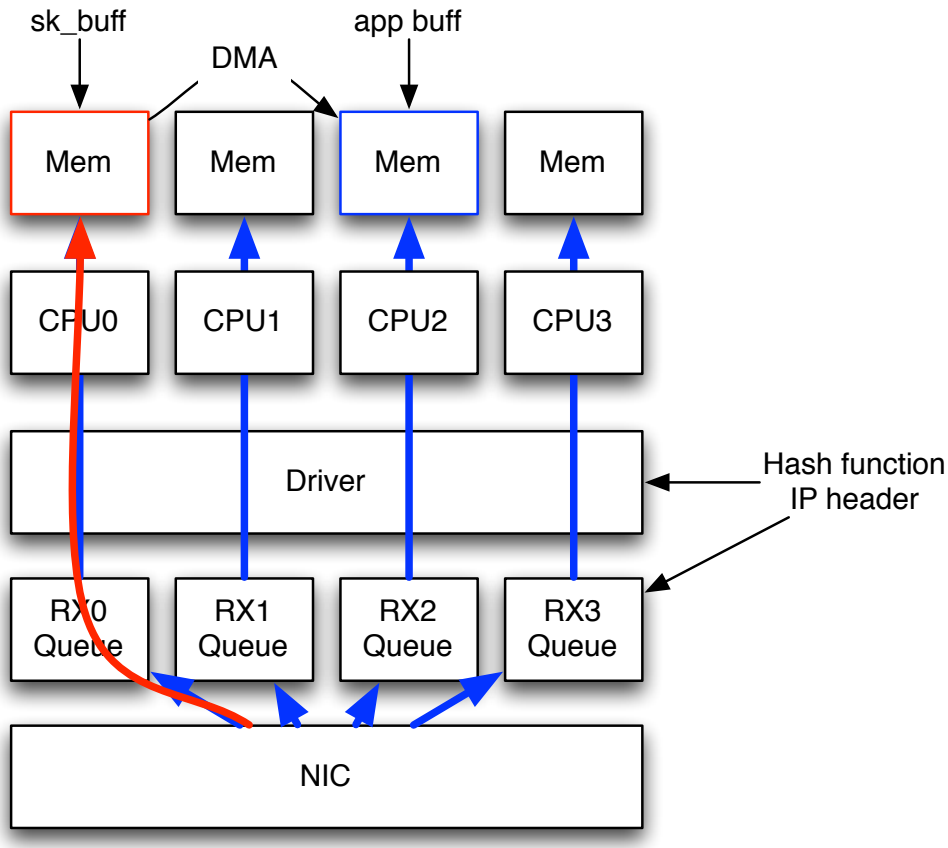
Multi Queue Network Cards

- Scale network performance horizontally - Multicore



Typical Linux Networking Stack - Receive Side

Multi Queue Network Cards (2)



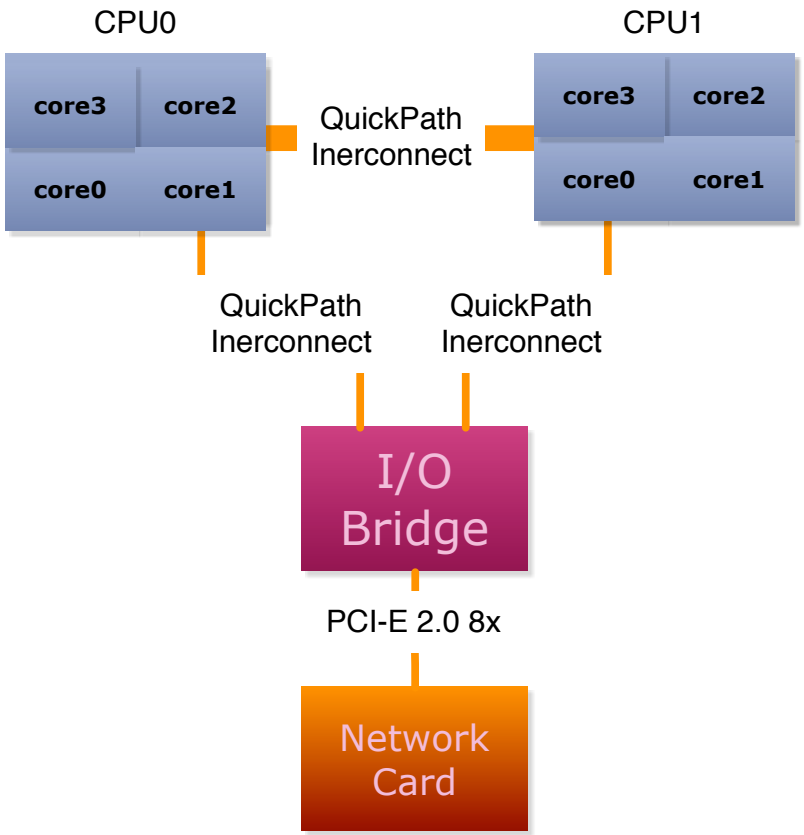
Typical Linux Networking Stack - Receive Side - Unfavorable mapping

Experimental setup

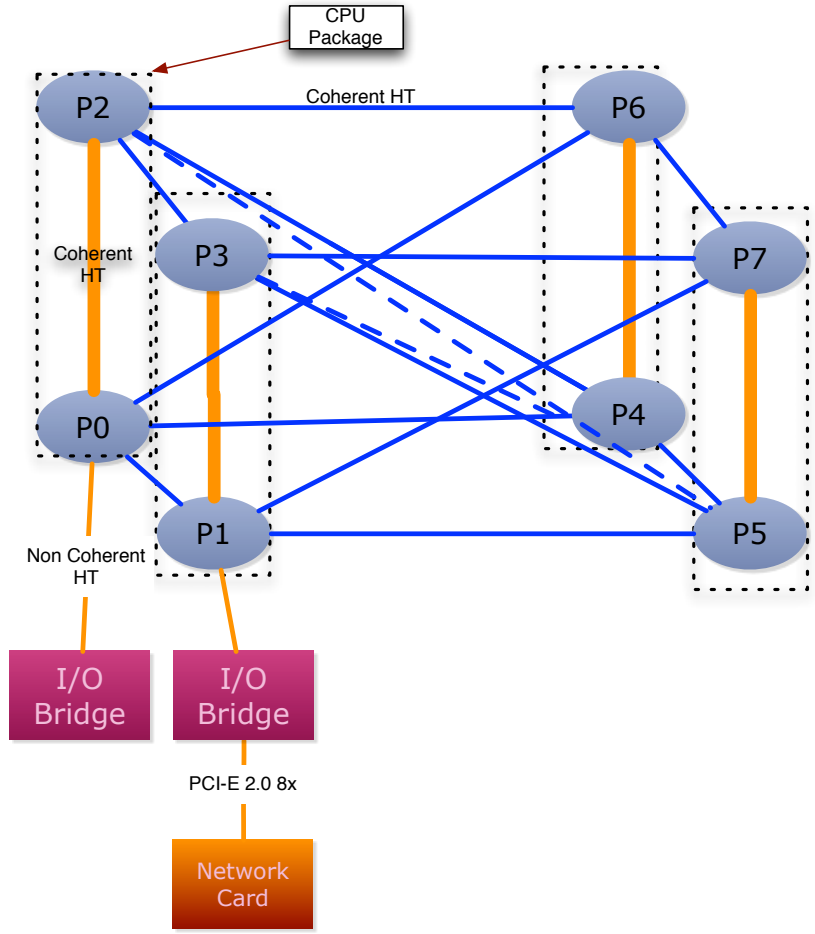
- 2 servers connected back to back w/ 40GE
  - Intel Nehalem (dual socket) - 8 cores
  - AMD Magny Cours (quad socket) - 48 cores
- How does NUMA affect network throughput?
- How can systems be tuned to achieve maximum throughput?

Server Model	Supermicro H8DTT-HIBQF	Dell R815
CPU Model	Intel Xeon E5620 2.4GHz	AMD Opteron6172 2.1GHz
Core Count	2 x 4 cores	4 x 12 cores
RAM	24GB	128 GB

NUMA Layout



Intel Nehalem (2P) I/O Topolgy

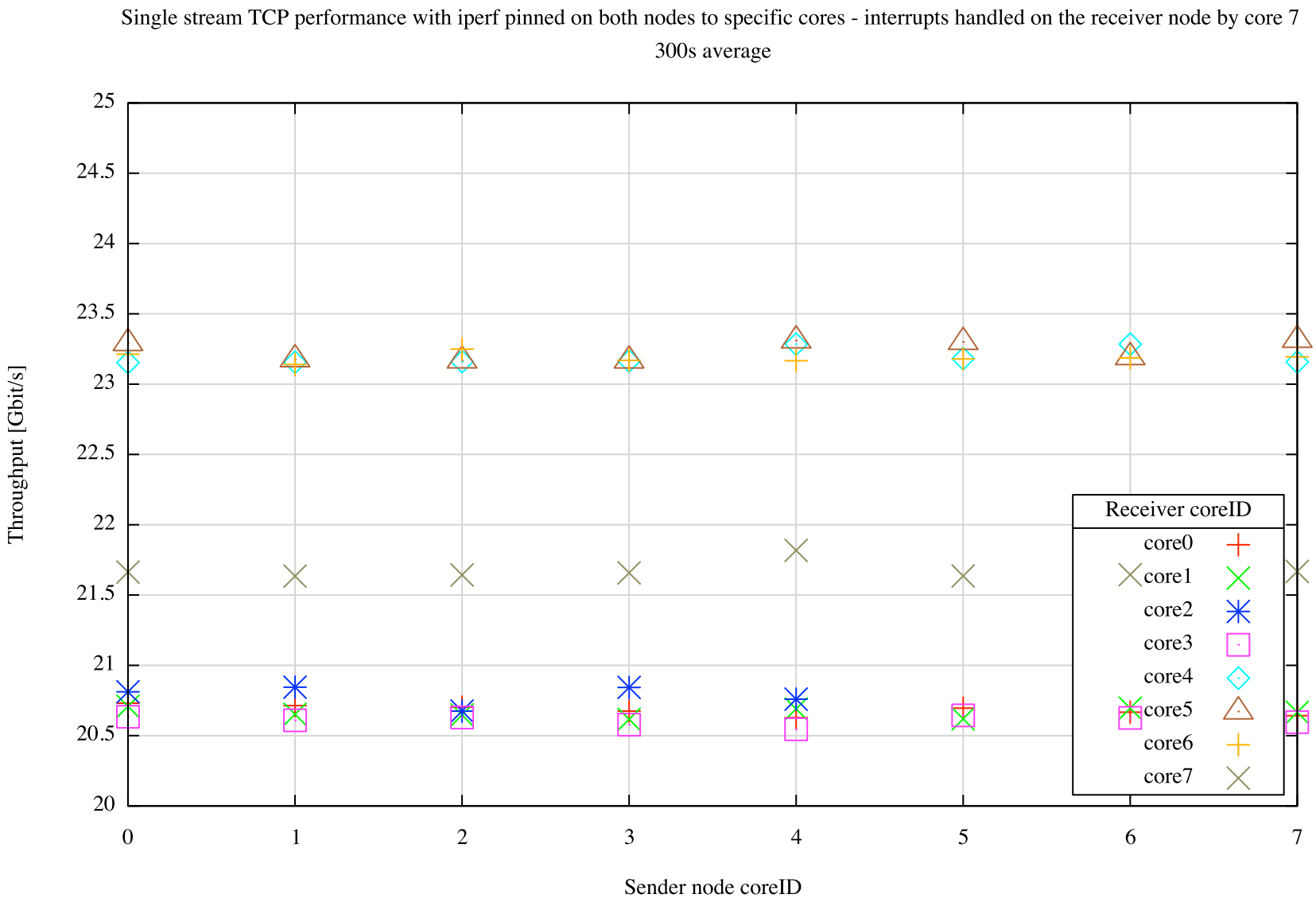


AMD Magny Cours (4P) I/O topology

Measurements

- Pinning of sender and receiver using `numactl`
- Patched version of `iperf` to keep source port fixed
- AMD Magny Cours as sender and receiver
- Intel Nehalem as receiver
- Single TCP flow with TCP Offload Engine enabled

Intel Nehalem Results

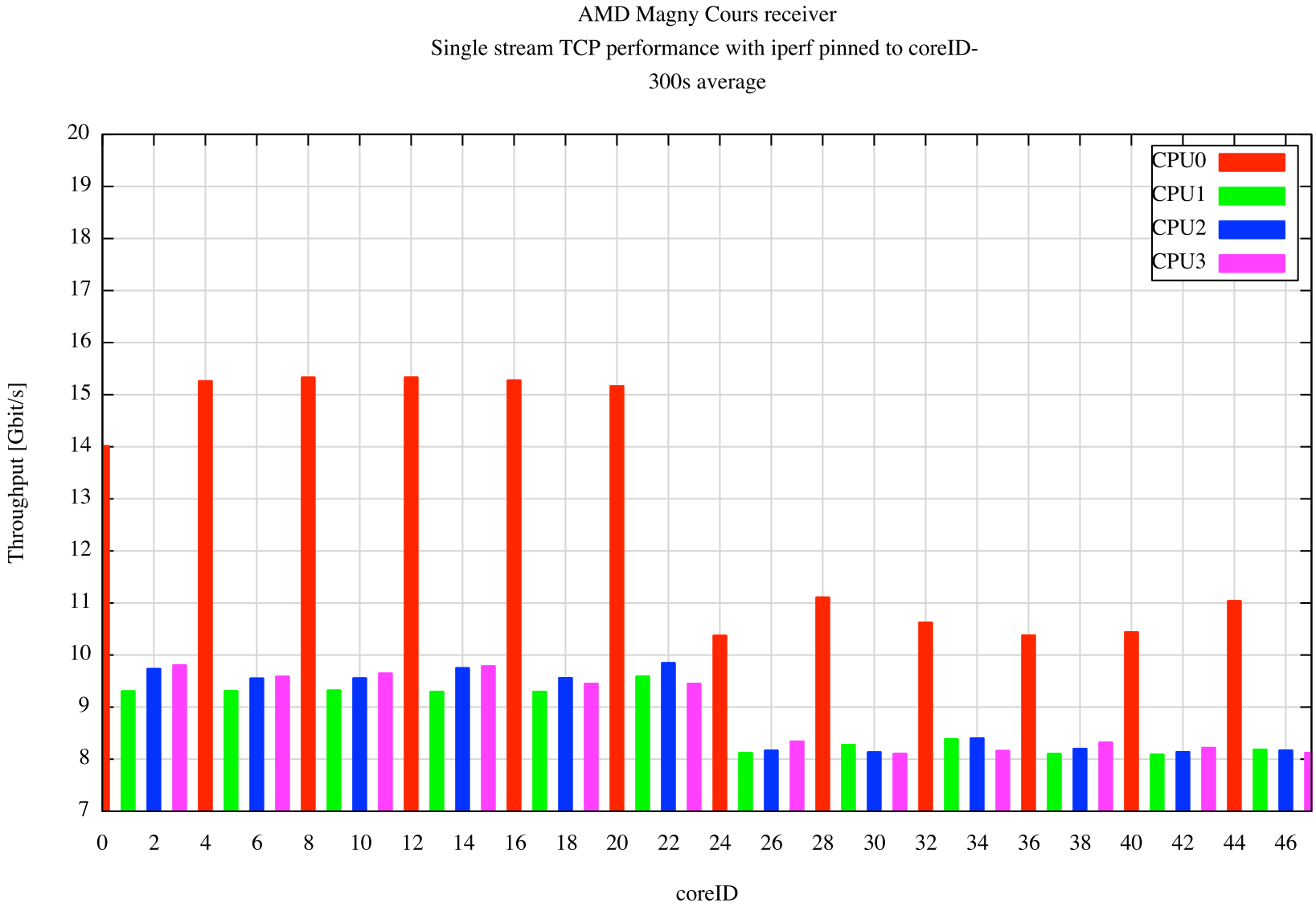


Two Intel Servers acting as sender and receiver

AMD Magny Cours Results

AMD server sender, Intel server receiver

AMD Magny Cours Results(2)



AMD server receiver, Intel server sender

Conclusions

- NUMA characteristics need to be considered when targeting maximum performance
- No easy way to steer traffic to the optimal core - IRQ pinning isn't sufficient
- PCI-E Interface upgrade will not resolve the problem as memory architecture is the real culprit

Thank you!

c.dumitru@uva.nl