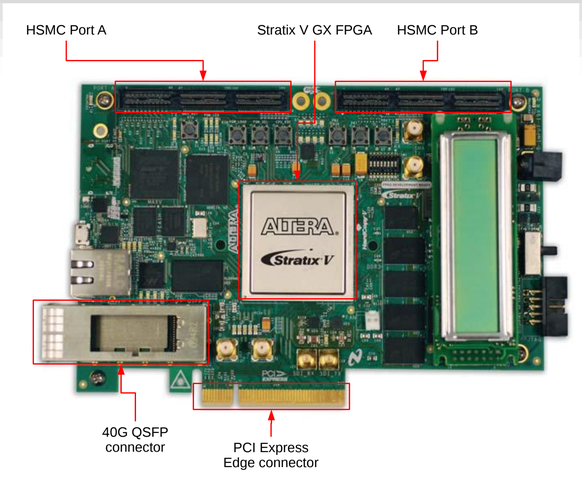
CHEP2016
San Francisco
10-14 October 2016

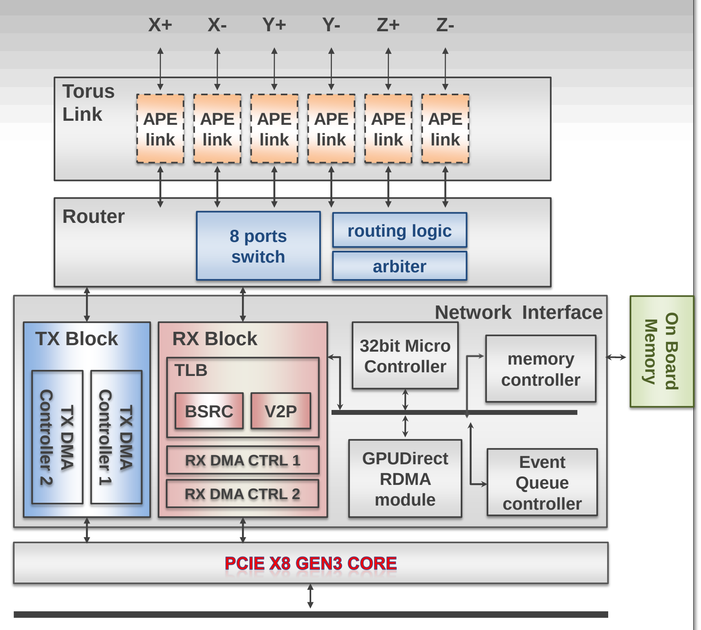
Latest generation interconnect technologies in APEnet+ networking infrastructure

R. Ammendola[1], A. Biagioni[2], P. Cretaro[2], O. Frezza[2], F. Lo Cicero[2], A. Lonardo[2], M. Martinelli[2], P. S. Paolucci[2], E. Pastorelli[2], D. Rossetti[3], F. Simula[2], P. Vicini[2]
[1]Sezione di Tor Vergata, Istituto Nazionale di Fisica Nucleare, Rome, Italy, [2]Sezione di Roma, Istituto Nazionale di Fisica Nucleare, Rome, Italy, [3]nVIDIA Corp, Santa Clara, CA, USA

APEnet overview



Development board

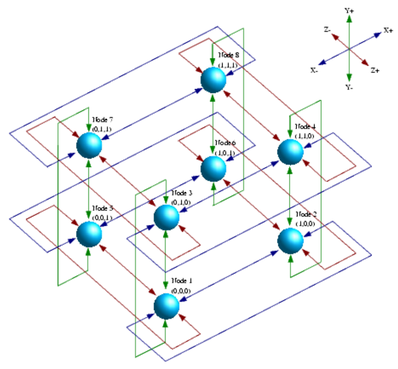


General APEnet architectural overview

APEnet+ is the high performance, low latency interconnect system developed at INFN targeting hybrid CPU/GPU-based HPC platforms

Based on DK-DEV-5SGXEA7N dev kit:

- New 28nm Stratix V FPGA
- 40Gb QSFP+ standard interconnect fabric
- HSMC expansion ports
- PCIe connector
- 1Gbit PHY
- **2D/3D toroidal mesh topology for point-to-point dead-lock free communications**



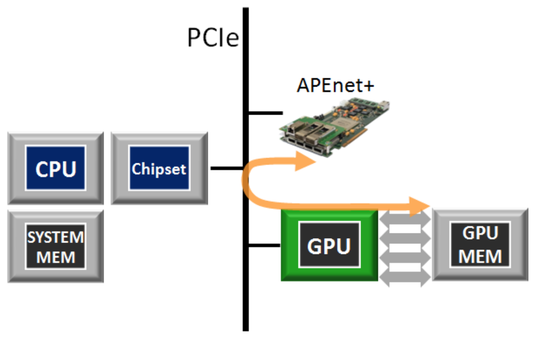
Preliminary BER measurement

Cable	BER	Data Rate
10m Mellanox optical cable	< 2.36 E-14	11.3 Gbps
1m Mellanox copper cable	< 1.10 E-13	10.0 Gbps

Enhanced embedded transceiver

- up to 14.1 Gbps
- X channel implemented using 40Gbps QSPF+ connector
- Y/Z channels implemented on the HSMC interfaces

- **RDMA** transfer protocol → CPU offload
- **GPUDirect RDMA** → GPU to GPU communication
- Latency reduction for small messages

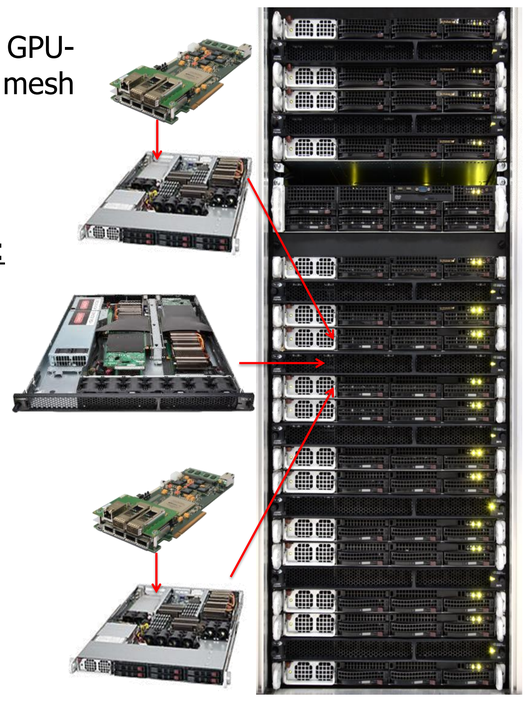


PCIe Gen3 x8

- Featuring 8.0 GT/s
- Encoding scheme 8b10b (20%) -> 128b130b (2%)
- PLDA PCIe CORE IP uses axi4 interface protocol

QUonG HPC System

QUantum chromodynamics on Gpu is a hybrid, GPU-accelerated x86_64 cluster with a 3D toroidal mesh topology, able to scale up to $10^4 \div 10^5$ nodes.



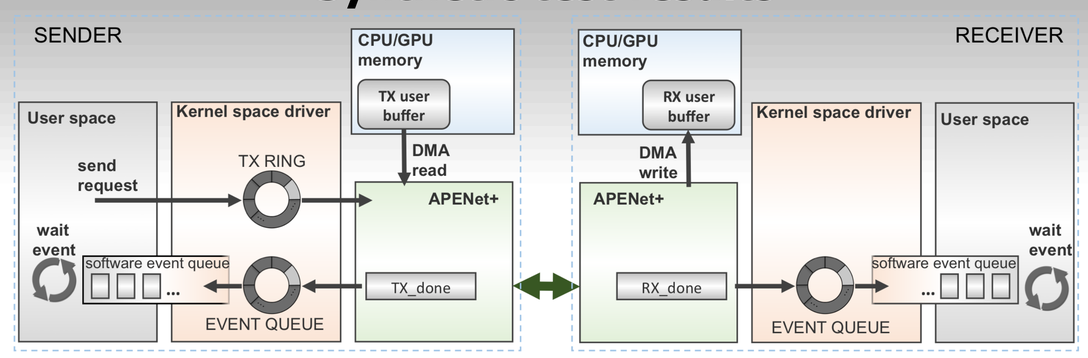
Current status of QUonG Hybrid Computing system:

- 16 nodes (4x4x1 topology)

Each node:

- Double Intel Xeon E5620
- 48GB System Memory
- 2x S2075 NVIDIA Fermi GPUs
- 40 Gb/s InfiniBand HCA
- 1 APEnet+ "V4" custom board (PCI gen2 on Stratix IV FPGA)

Synthetic test results



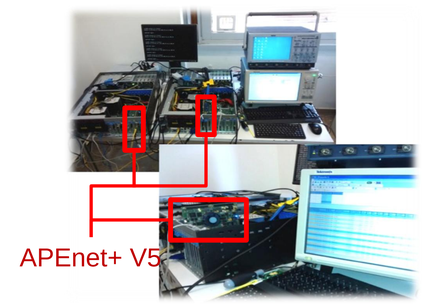
1. Buffers allocation (pin and lock memory)
   On tx side: buffer with data to transfer
   On rx side: buffer where data must be stored
2. Send request → descriptor with TX physical address and RX virtual address
3. Wait event
   On tx side: wait for the TX_DONE event
   On rx side: wait for RX_DONE event

On the transmitting side

4. APENet DMA-reads data from memory (physical address)
5. Data are actually sent
6. TX_DONE event is generated and DMA-written in memory. Then it's notified to the user

On the receiving side

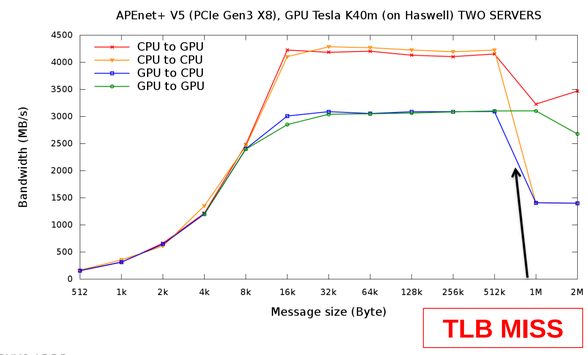
7. APENet DMA-writes data into memory (memory virtual address is translated into the correspondent physical address – CPU offloading!)
8. RX_DONE event is generated and DMA-written in memory. Then it is notified to the user



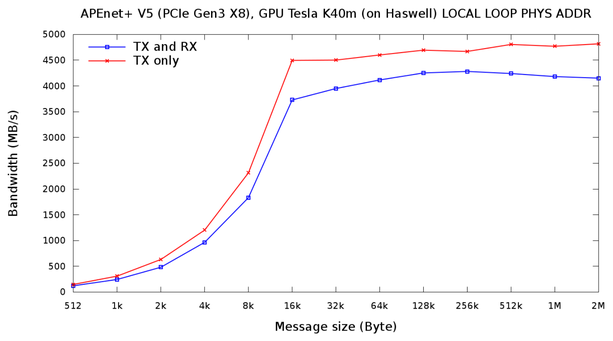
Testbed

- Dual Socket E5-2620v3 @ 2.40GHz
- Haswell
- GPU Nvidia K40m

Bandwidth (One-way)
all packets are sent in a once and time required for ALL completions (TX+RX) to land is measured and averaged.

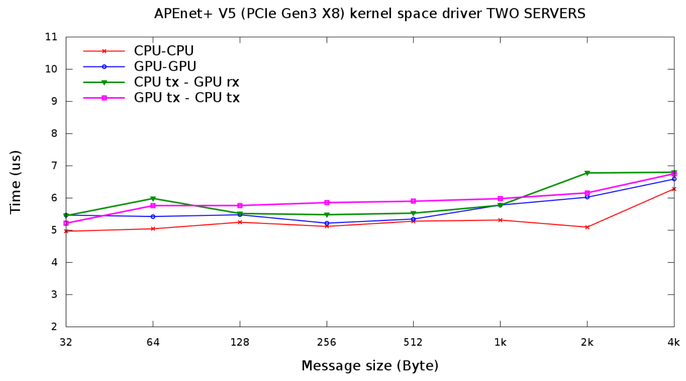


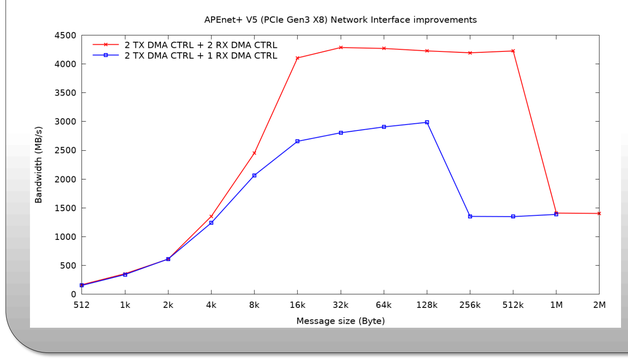
TLB MISS



Bandwidth (One-way) flush
all packets are sent in a once and time required only for the TX completions to land is measured and averaged (RX path flushed)

Latency (Roundtrip)
measured by sending a packet of a certain size and measuring the time required for completions to arrive.





Improvements

- Memory write process increases by ~45% by adding a second RX DMA Controller.
- TLB implementation is able to manage an increased amount of entries (128)

Istituto Nazionale di Fisica Nucleare

Contacts
APE project: http://apegate.roma1.infn.it/APE
APE coordinator: piero.vicini@roma1.infn.it
Presenter: andrea.biagioni@roma1.infn.it