Interconnection Network for Tightly Coupled Accelerators Architecture

Toshihiro Hanawa, Yuetsu Kodama,
Taisuke Boku, Mitsuhisa Sato

Center for Computational Sciences

University of Tsukuba, Japan

What is “Tightly Coupled Accelerators (TCA)” ?

Concept:

- **Direct connection between accelerators (GPUs) over the nodes**
  - Eliminate extra memory copies to the host
  - Reduce latency, improve strong scaling with small data size for scientific applications
- **Using PCIe as a communication link between accelerators over the nodes**
  - PCIe just performs packet transfer and direct device P2P communication is available.
- **PEACH2: PCI Express Adaptive Communication Hub ver. 2**
  - In order to configure TCA, each node is connected to other nodes through PEACH2 chip.

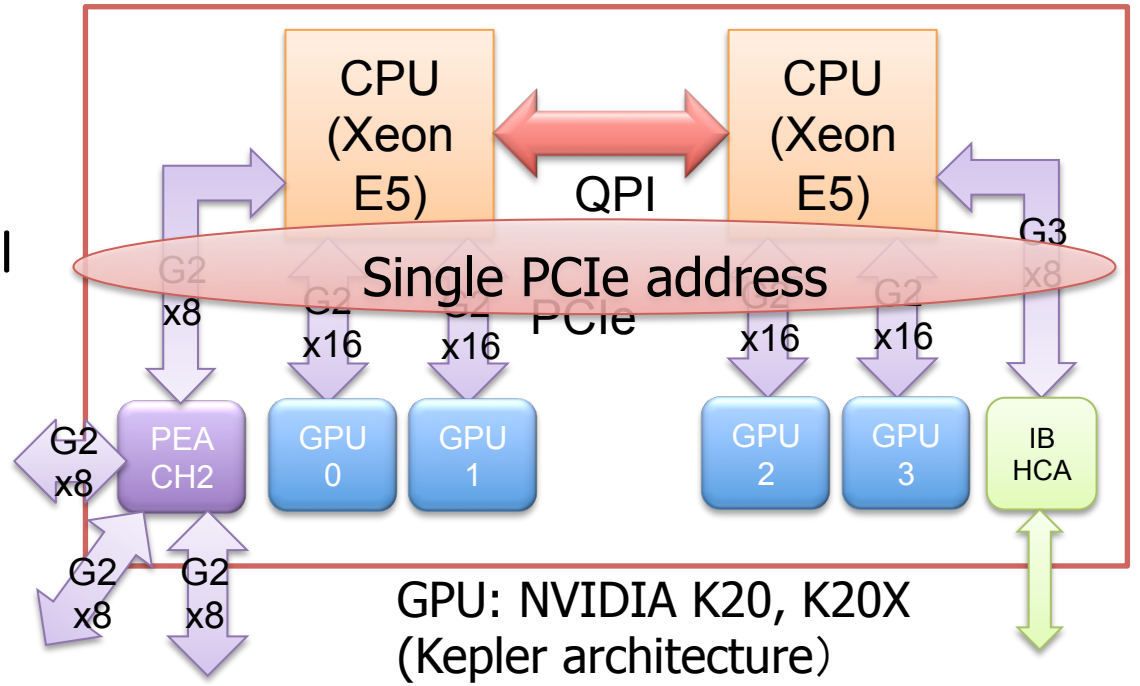
Design policy of PEACH2

- **Implement by FPGA with four PCIe Gen.2 IPs**
  - Altera Stratix IV GX
  - Prototyping, flexible enhancement
- **Sufficient communication bandwidth**
  - PCI Express Gen2 x8 for each port
  - Sophisticated DMA controller
    - Chaining DMA
- **Latency reduction**
  - Hardwired logic
  - Low-overhead routing mechanism
    - Efficient address mapping in PCIe address area using unused bits
    - Simple comparator for decision of output port

It is not only a proof-of-concept implementation, but it will also be available for product-run in GPU cluster.

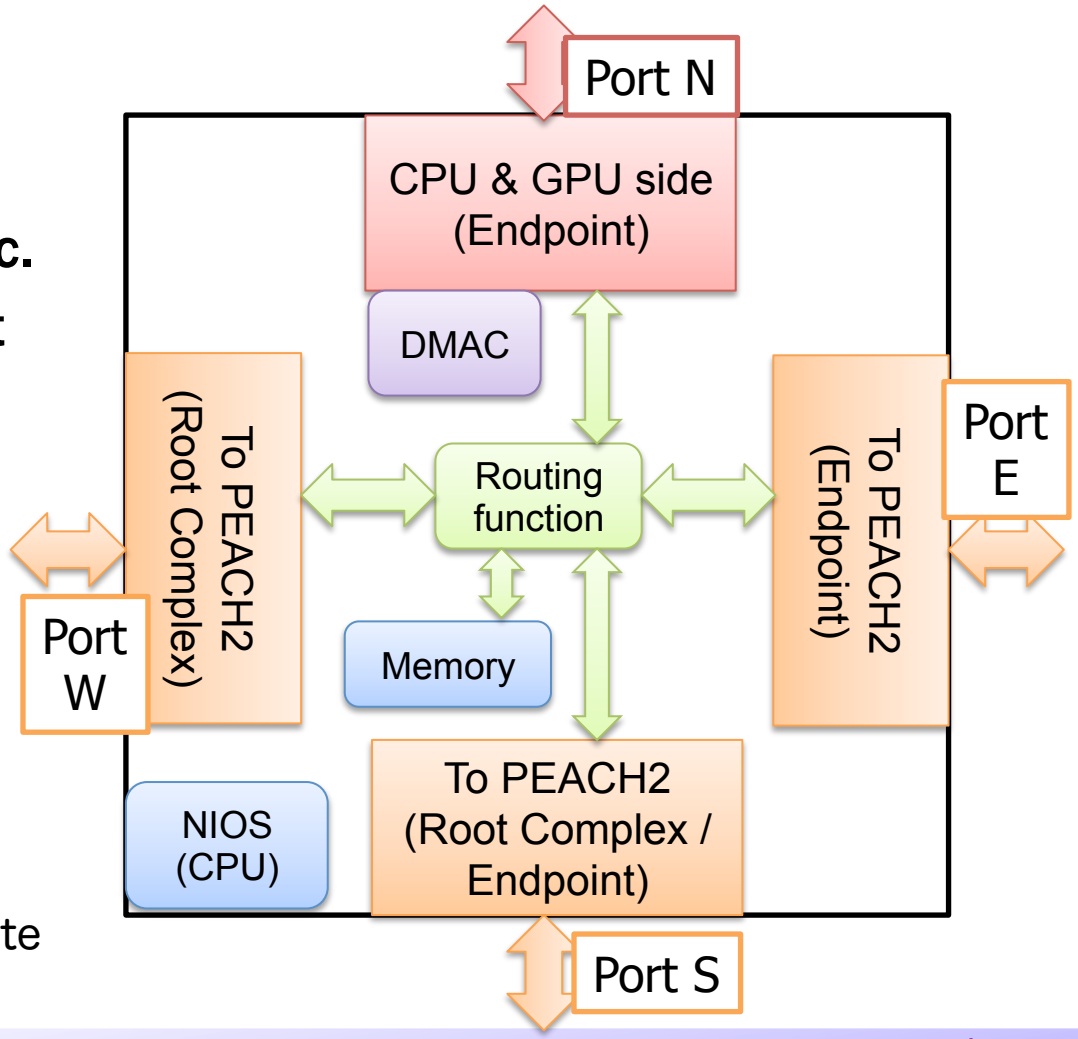
TCA node structure example

- **PEACH2 can access all GPUs**
  - NVIDIA Kepler architecture + CUDA 5.0 “GPUDirect Support for RDMA”
  - Performance over QPI is quite bad.
    => support only for GPU0, GPU1
- **Connect among 3 nodes using PEACH2**



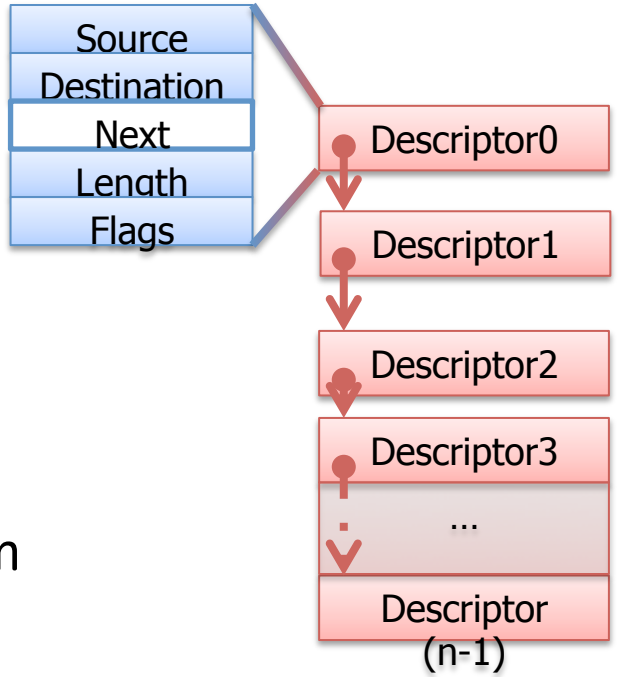
Overview of PEACH2 chip

- **Fully compatible with PCIe Gen2 spec.**
- **<u>Root and EndPoint must be paired</u> according to PCIe spec.**
- **Port N: connected to the host and GPUs**
- **Port E and W: form the ring topology**
- **Port S: connected to the other ring**
  - Selectable between Root and Endpoint
- **Write only except Port N**
  - Instead, “Proxy write” on remote node realizes pseudo-read.



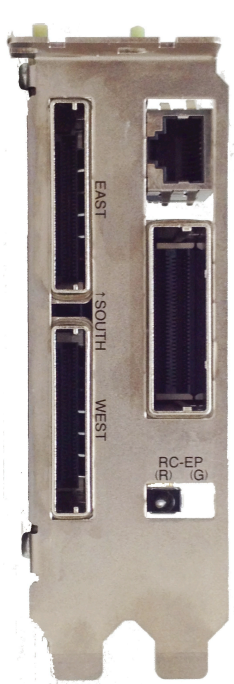
Communication by PEACH2

- **PIO**
  - CPU can store the data to remote node directly using mmap.
- **DMA**
  - Chaining mode
    - DMA requests are prepared as the DMA descriptors chained in the host memory.
    - Multiple DMA transactions are operated automatically according to the DMA descriptors by hardware.
  - Register mode
    - Up to 16 requests
    - No overhead to transfer descriptors from host
  - Block stride transfer function

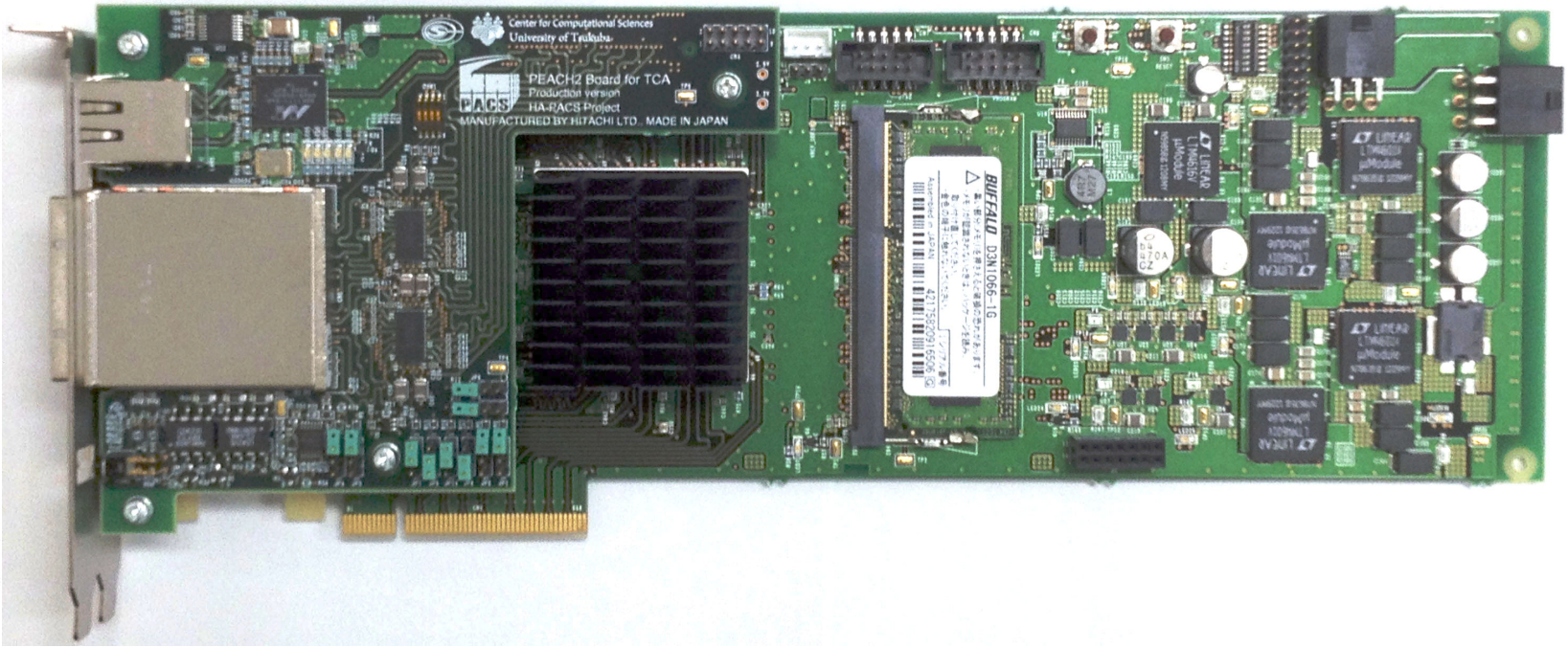


PEACH2 board (Production version for HA-PACS/TCA)

- **PCI Express Gen2 x8 peripheral board**
  - Compatible with PCIe Spec.

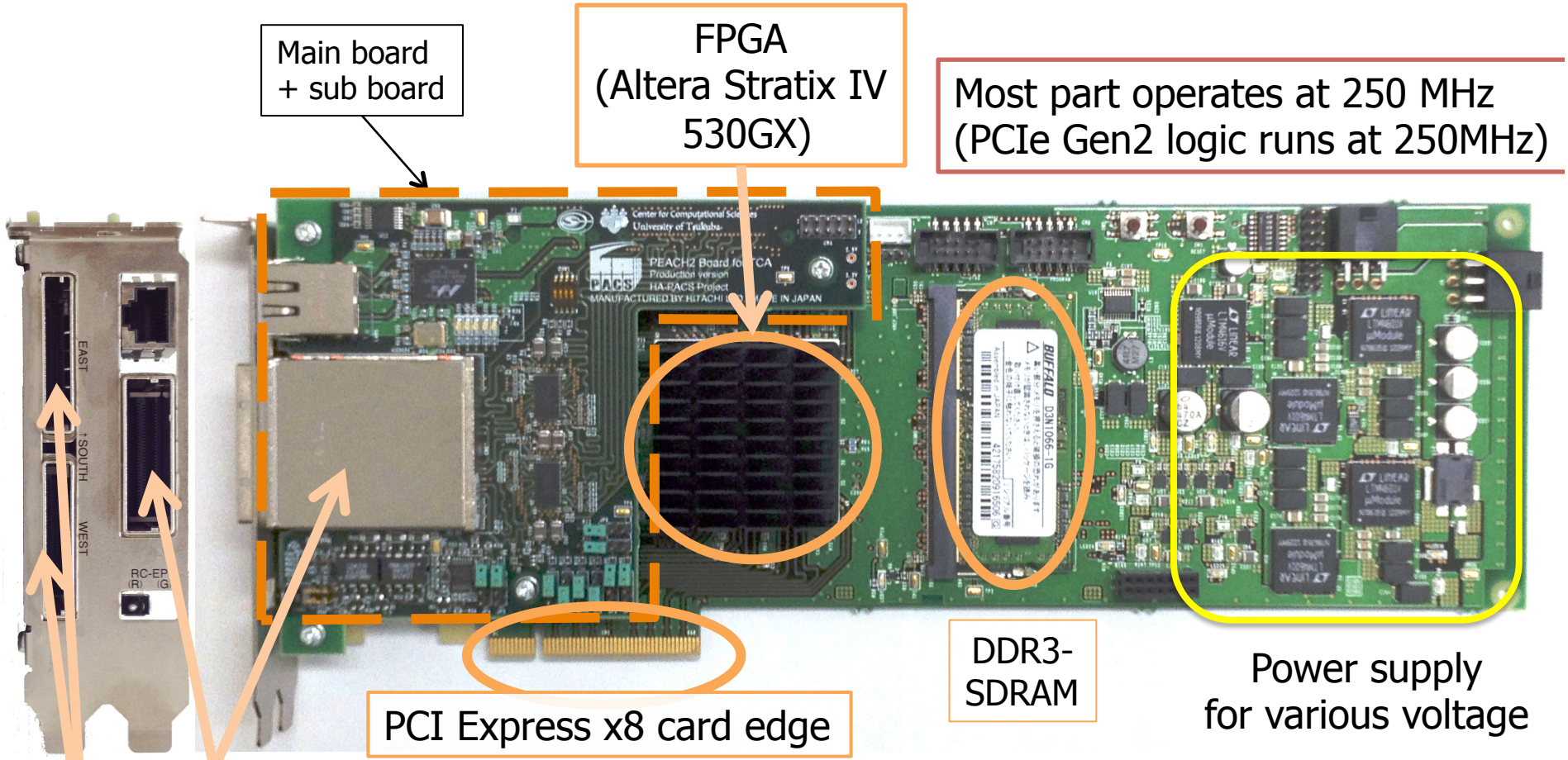


Side View



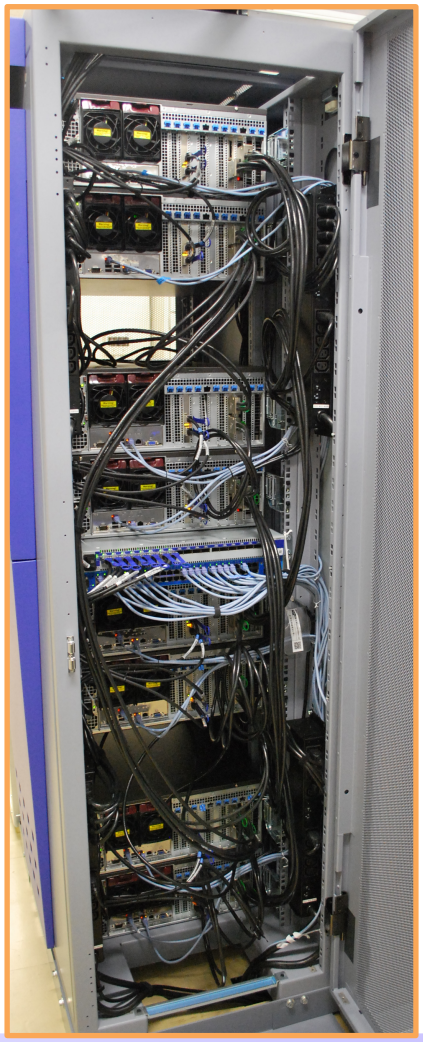
Top View

PEACH2 board (Production version for HA-PACS/TCA)



Performance Evaluation

- **Environment: 8node GPU cluster (TCAMINI)**
  - CPU: Intel Xeon-E5 (SandyBridge EP) 2.6GHz x2socket
  - MB: SuperMicro X9DRG-QF
  - Memory: DDR3 128GB
  - OS: CentOS 6.3 (kernel 2.6.32-279.22.1.el6.x86_64)
  - GPU: NVIDIA K20, GDDR5 5GB x1
  - CUDA: 5.0, NVIDIA-Linux-x86_64-310.32
  - PEACH2 board: Altera Stratix IV 530GX
  - MPI: MVAPICH2 1.9 with IB FDR10



Evaluation items

- **Ping-pong performance between nodes**
  - Latency and bandwidth
  - Written as application
  - Comparison with MVAPICH2 1.9 (with CUDA support) for GPU-GPU communication

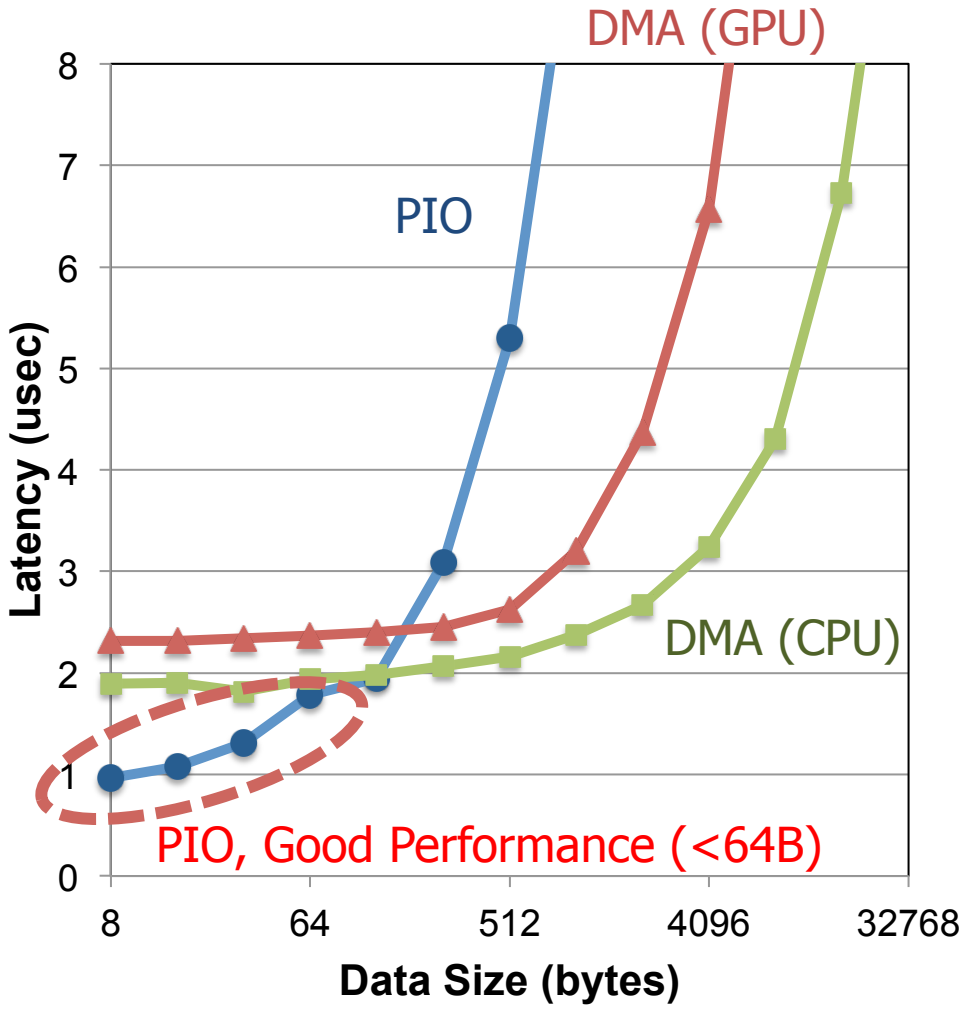
- In order to access GPU memory by the other device, "GPU Direct support for RDMA" in CUDA5 API is used.
  - Special driver named "TCA p2p driver" to enable memory mapping is developed.
- "PEACH2 driver" to control the board is also developed.


Ping-pong Latency

Minimum Latency (nearest neighbor comm.)

- **PIO: CPU to CPU: 0.9us**
- **DMA:CPU to CPU: 1.9us**
  GPU to GPU: 2.3us

(cf. MVAPICH2 1.9: 19 usec)



Ping-pong Latency

Minimum Latency

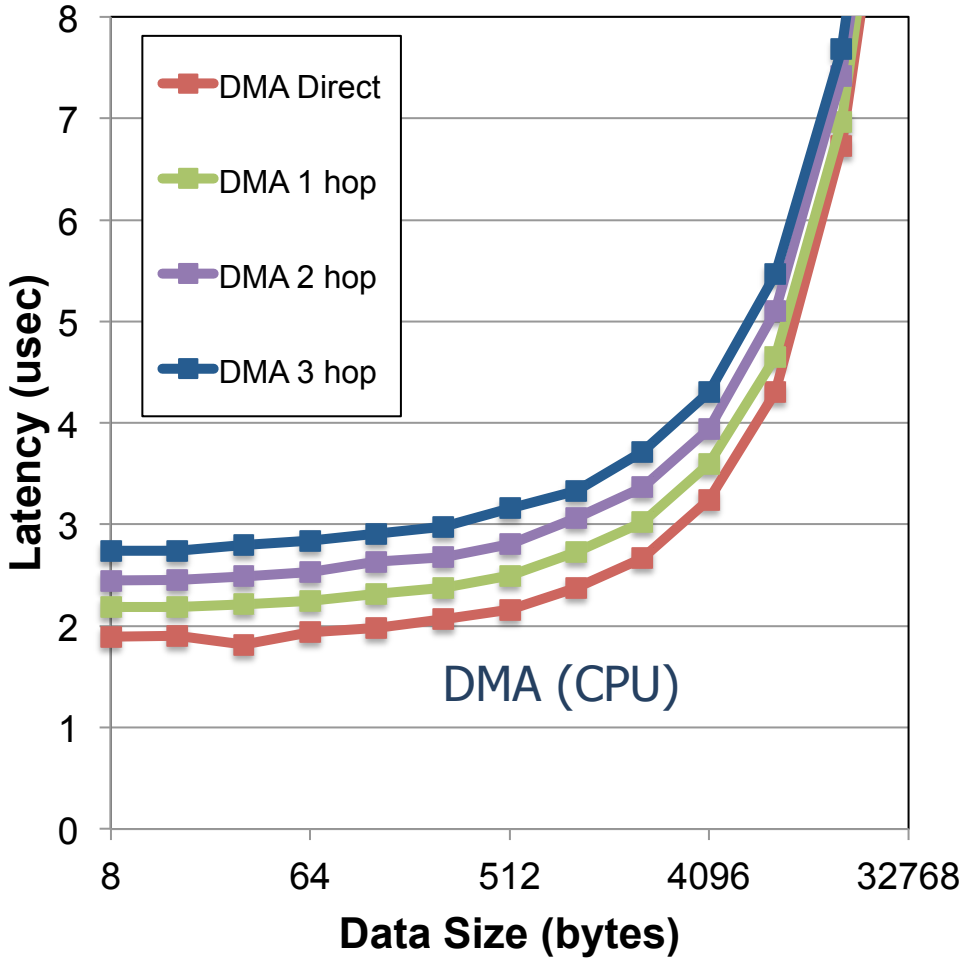
(nearest neighbor comm.)

- PIO: CPU to CPU: 0.9us
- DMA:CPU to CPU: 1.9us
  GPU to GPU: 2.3us

(cf. MVAPICH2 1.9:
19 usec)

Forwarding overhead

- **200~300 nsec**

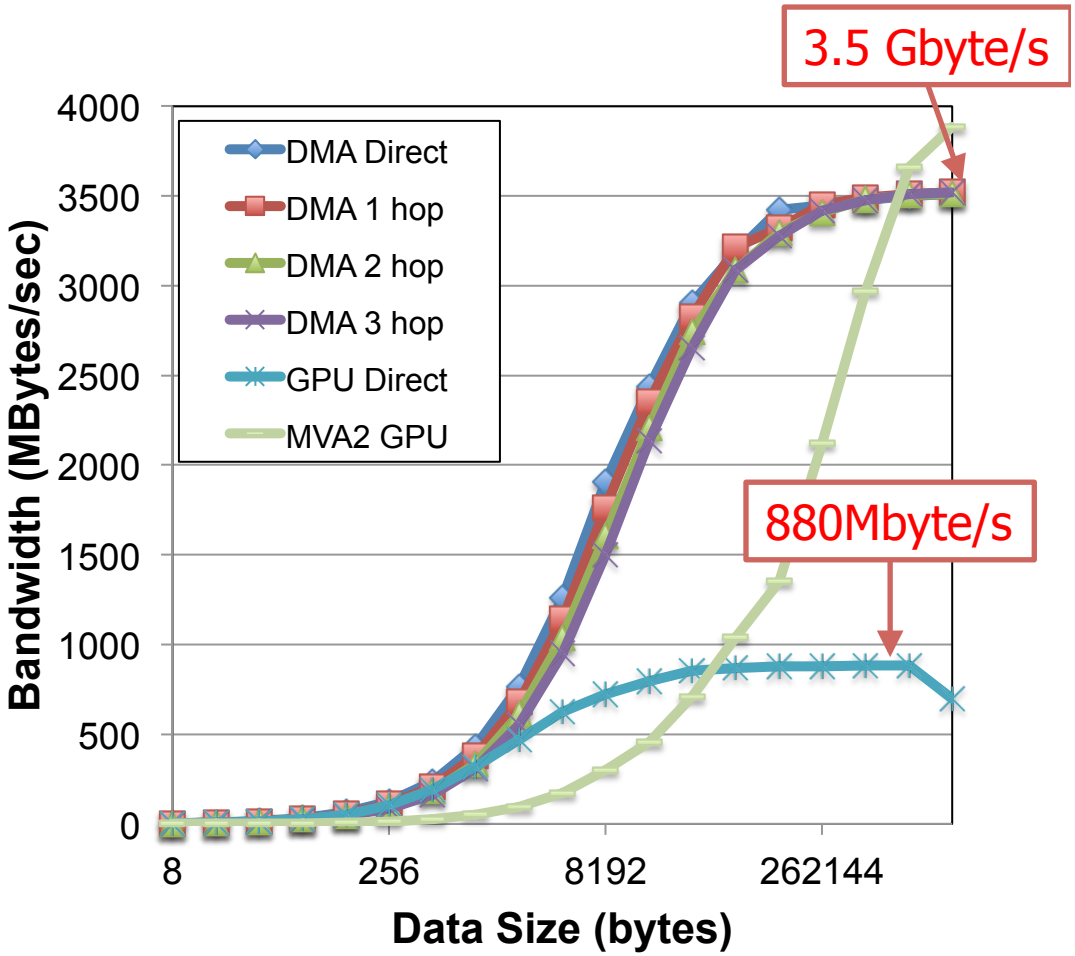


Ping-pong Bandwidth

- **Max. 3.5 GByte/sec**
  - 95% of theoretical peak
  - Converge to the same peak with various hop counts

Max Payload Size = 256byte
Theoretical peak:
4Gbyte/sec × 256 / (256 + 16 + 2 + 4 + 1 + 1) = 3.66 Gbyte/s

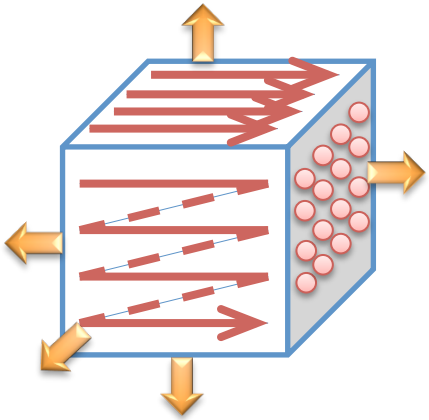
- **GPU to GPU DMA is saturated by up to 880MByte/sec.**
  - PCIe switch embedded in SandyBridge doesn't have enough resources for PCIe device read.
  - In IvyBridge, the performance is expected to be improved…
    - Performance will be expected as same as CPU to CPU case.



Programming for TCA cluster

- **Data transfer to remote GPU within TCA can be treated like local GPU.**
- **In particular, suitable for stencil computation**
  - Good performance at nearest neighbor communication due to direct network
  - Chaining DMA can bundle data transfers for every “Halo” planes
    - XY-plane: contiguous array
    - XZ-plane: block stride
    - YZ-plane: stride
  - In each iteration, DMA descriptors can be reused and only a DMA kick operation is needed

=> <u>Improve strong scaling with small data size</u>



Bundle to 1 DMA

Related Work

- **Non Transparent Bridge (NTB)**
  - NTB appends the bridge function to a downstream port of the PCI-E switch.
  - Inflexible, the host must recognize during the BIOS scan
  - It is not defined in the standard of PCI-E and is incompatible with the vendors.
- **APEnet+ (Italy)**
  - GPU direct copy using Fermi GPU, different protocol from TCA is used.
  - Latency between GPUs is around 5us?
  - Original 3-D Torus network, QSFP+ cable
- **MVAPICH2 + GPUDirect**
  - CUDA5, Kepler
  - Latency between GPUs is reported as 4.5us at Jun., but currently not released (4Q of 2013?)

Summary

- **TCA: Tightly Coupled Accelerators**
  - TCA enables direct communication among accelerators as an element technology becomes a basic technology for next gen's accelerated computing in exa-scale era.
- **PEACH2 board: Implementation for realizing TCA using PCIe technology**
  - Bandwidth: max. 3.5 Gbyte/sec between CPUs (over 95% of theoretical peak)
    Min. Latency: 0.9 us (PIO), 1.9 us (DMA between CPUs), 2.3 us (DMA between GPUs)
  - GPU-GPU communication over the nodes can be demonstrated with 8 node cluster.
  - By the ping-pong program, PEACH2 can achieve lower latency than existing technology, such as MVAPICH2 in small data size.
- **HA-PACS/TCA with 64 nodes will be installed on the end of Oct. 2013.**
  - Actual proof system of TCA architecture with 4 GPUs per each node
  - Development of the HPC application using TCA, and production-run
    - Performance comparison between TCA and InfiniBand
    - Hybrid communication scheme using TCA and InfiniBand will be investigated.