GPU Cluster with Proprietary Interconnect Utilizing GPU Direct Support for RDMA

Toshihiro Hanawa
Information Technology Center, The University of Tokyo

Yuetsu Kodama, Taisuke Boku, Mitsuhisa Sato
Center for Computational Sciences, University of Tsukuba
Agenda

- Background
- HA-PACS Project
- Introduction of HA-PACS / TCA
  - Organization of TCA
  - PEACH2 Board designed for TCA
- Performance Evaluation and Comparison
- Summary
Advantageous Features
- High peak performance / cost ratio
- High peak performance / power ratio

Examples of HPC System:
- GPU Clusters and MPPs in TOP500 (Nov. 2013)
  - 2nd: Titan (NVIDIA K20X, Rpeak=27 PFLOPS)
  - 6th: Piz Daint (NVIDIA K20X, Rpeak=7.8 PFLOPS)
  - 11th: TSUBAME2.5 (NVIDIA K20X, Rpeak=5.6 PFLOPS)
  - 39 systems use NVIDIA GPUs.

GPU Clusters in Green500 (Nov. 2013) (“Greenest” Supercomputers ranked in Top500)
- 1st: TSUBAME-KFC (NVIDIA K20X, 4.5 GF/Watt)
- 2nd: Wilkes (NVIDIA K20, 3.6 GF/Watt)
- 3rd: HA-PACS/TCA (NVIDIA K20X, 3.5 GF/Watt)

Top10 were occupied by NVIDIA K20 families!
Issues of GPU Cluster

- Problems of GPGPU for HPC: Increase of communication latency between GPUs over nodes
  - Memory size limitation --- Ex) K20X: 6GByte vs. CPU: 32 – 128 Gbyte
    ⇒ To compute large scale problems by GPU, many GPUs across multiple compute nodes are used.
  - Data I/O performance limitation : Ex) K20X: PCIe gen2 x16
    Peak Performance : 8GB/s (I/O) ⇔ 1.3 TFLOPS (Computation)
  - Extra memory copy to host memory is required.
    - GPU mem ⇒ CPU mem ⇒ (MPI) ⇒ CPU mem ⇒ GPU mem
      (Now, “GPU Direct for RDMA” technology can eliminate memory copy to the host.)
  - Ultra-low latency between GPUs is important for next generation’s HPC

Our target is developing a direct communication system between external GPUs for a feasibility study for future accelerated computing.
⇒ “Tightly Coupled Accelerators (TCA)” architecture
HA-PACS Project

- HA-PACS (Highly Accelerated Parallel Advanced system for Computational Sciences)
  - 8th generation of PAX/PACS series supercomputer
  - FY2011-2013, operation until FY2016(?)

- Promotion of computational science applications in key areas in CCS-Tsukuba
  - Target field: QCD, astrophysics, QM/MM (quantum mechanics / molecular mechanics, bioscience)

HA-PACS is not only a “commodity GPU cluster” but also experiment platform

- HA-PACS base cluster
  - for development of GPU-accelerated code for target fields, and performing product-run
  - Now in operation since Feb. 2012

- HA-PACS/TCA (TCA = Tightly Coupled Accelerators)
  - for elementary research on direct communication technology for accelerated computing
  - Our original communication chip named “PEACH2” was installed in each node.
  - Now in operation since Nov. 2013
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.
  - Transfer PCIe packets over the nodes as-is

Implementation:
- **PEACH2**: PCI Express Adaptive Communication Hub ver. 2
  - In order to configure TCA, each node is connected to other nodes through PEACH2 chip.
PEACH2 can access all GPUs
- NVIDIA Kepler architecture + CUDA 5.0 “GPUDirect Support for RDMA”
- Performance over QPI is miserable. => support only for GPU0, GPU1

Connect among 3 nodes using remaining PEACH2 port

Similar to ordinary GPU cluster configuration except PEACH2
- 80 PCIe lanes are required
PEACH2 can access all GPUs
- NVIDIA Kepler architecture + CUDA 5.0 “GPUDirect Support for RDMA”
- Performance over QPI is miserable. => support only for GPU0, GPU1
- Connect among 3 nodes using remaining PEACH2 port

Similar to ordinary GPU cluster configuration except PEACH2
- 80 PCIe lanes are required
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 (40Gbps = IB QDR)
  - Sophisticated DMA controller
    - Chaining DMA, Block-stride transfer function

- 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.
Overview of PEACH2 chip

- Fully compatible with PCIe Gen2 spec.
- Root and EndPoint must be paired according to PCIe spec.
- Port N: connected to the host and GPUs
- Port E and W: make 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

1. PIO
   - CPU store => actual communication, ultra low latency
   - Suitable for small data

2. DMA (Chaining mode)
   - DMA requests are prepared as the descriptors chained in the host memory.
   - Multiple DMA transactions are operated automatically according to the descriptors by hardware.

3. DMA (Register mode)
   - No overhead to transfer descriptors from host
   - Up to 16 requests

   Block-stride DMA is available in each mode.
   - Transfer size, gap length, repeat count
PEACH2 board (Production version for HA-PACS/TCA)

- Main board + sub board
- FPGA (Altera Stratix IV 530GX)
- Most part operates at 250 MHz (PCIe Gen2 logic runs at 250MHz)
- Power supply for various voltage
- PCI Express x8 card edge
- PCIe x16 cable connector
- PCIe x8 cable connector
- DDR3-SDRAM
HA-PACS/TCA Compute Node

Front View
(8 node / rack)
3U height

Rear View

PEACH2 Board is installed here
### Spec. of HA-PACS base cluster & HA-PACS/TCA

<table>
<thead>
<tr>
<th></th>
<th>Base cluster (Feb. 2012)</th>
<th>TCA (Nov. 2013)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Node</strong></td>
<td>CRAY GreenBlade 8204</td>
<td>CRAY 3623G4-SM</td>
</tr>
<tr>
<td><strong>MotherBoard</strong></td>
<td>Intel Washington Pass</td>
<td>SuperMicro X9DRG-QF</td>
</tr>
<tr>
<td><strong>CPU</strong></td>
<td>Intel Xeon E5-2670 x 2 socket (SandyBridge-EP, 2.6GHz 8 core) x2</td>
<td>Intel Xeon E5-2680 v2 x 2 socket (IvyBridge-EP, 2.8GHz 10 core) x2</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>DDR3-1600 128 GB</td>
<td>DDR3-1866 128 GB</td>
</tr>
<tr>
<td><strong>GPU</strong></td>
<td>NVIDIA M2090 x4</td>
<td>NVIDIA K20X x 4</td>
</tr>
<tr>
<td><strong># of Nodes</strong></td>
<td>268 (26)</td>
<td>64 (10)</td>
</tr>
<tr>
<td></td>
<td><strong>Interconnect</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mellanox InfiniBand QDR x 2 (Connect X-3)</td>
<td>Mellanox InfiniBand QDR x2 + PEACH2</td>
</tr>
<tr>
<td><strong>Peak Perf.</strong></td>
<td>802 TFlops</td>
<td>364 TFlops</td>
</tr>
<tr>
<td><strong>Power</strong></td>
<td>408 kW</td>
<td>99.3 kW</td>
</tr>
</tbody>
</table>

Totally, HA-PACS is over 1PFlops system!
HA-PACS/TCA (Compute Node)

- **4 Channels**
  - 1,866 MHz
  - 59.7 GB/sec

- **AVX**
  - (2.8 GHz x 8 flop/clock)
  - 22.4 GFLOPS x20
  - = 448.0 GFLOPS

- **Total:** 5.688 TFLOPS

- **1.31 TFLOPS x4**
  - = 5.24 TFLOPS

- **4 x NVIDIA K20X**
  - (6 GB, 250 GB/s)x4
  - = 24 GB, 1 TB/s

- **4 Channels**
  - 1,866 MHz
  - 59.7 GB/sec

- **(16 GB, 14.9 GB/s)x8**
  - = 128 GB, 119.4 GB/s

- **Legacy Devices**

- **PEACH2 board**
  - (Proprietary Interconnect for TCA)

**Red:** upgraded from base-cluster to TCA
HA-PACS/TCA (since Nov. 2013) + Base cluster

LINPACK: 277 Tflops (Efficiency 76%)
3.52GFLOPS/W #3 Green500 at Nov. 2013
Configuration of TCA Sub-cluster (16 nodes/group)

- Each group consists of 2 racks, 16 nodes. HA-PACS/TCA includes 4 TCA groups.
  - Orange: Ring
  - Red: Cross link between 2 rings

- In TCA sub-cluster, 32 GPUs can be treated seamlessly.
  - limited to 2 GPUs under the same socket per node
Evaluation Results

- Ping-pong performance between nodes
  - Latency and bandwidth
  - Written as application

- Comparison
  - MVAPICH2-GDR 2.0b (with/without GPU Direct support) for GPU-GPU communication on TCA nodes
    - A InfiniBand QDR link (40Gbps) is used, which has the same performance as PEACH2.
  - Performance over QPI on TCA nodes
  - SandyBridge platform

- 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: 0.8 us
- DMA: CPU to CPU: 1.8 us
  GPU to GPU: 2.3 us

cf. MV2-GDR 2.0b: 6.5 us (w/ GDR), 17 us (w/o GDR)
Ping-pong Latency

Minimum Latency
(nearest neighbor comm.)
- PIO: CPU to CPU: 0.8 us
- DMA: CPU to CPU: 1.8 us
  GPU to GPU: 2.3 us

Forwarding overhead
- 200-300 nsec
- BW converges to the same peak with various hop counts

Data Size (bytes)
Latency (usec)

- DMA Direct
- DMA 1 hop
- DMA 2 hop
- DMA 3 hop

Mar. 25, 2014
GPU Technology Conference 2014
THE UNIVERSITY OF TOKYO
University of Tsukuba
Max. 3.5 GByte/sec
- 95% of theoretical peak
- Converge to the same peak if hop count increases

Max Payload Size = 256 byte
Theoretical peak (detailed):
4GB/sec × 256 / (256 + 24) = 3.66 GB/s

GPU - GPU DMA performance is up to 2.6 GByte/sec.
- better than MV2GDR under < 1MB
- Over QPI: limited to 360MB/s
- SB(SandyBridge): limited to 880MB/s due to PCIe sw perf.
In particular, TCA is suitable for stencil computation

- Good performance at nearest neighbor communication due to direct network
- DMA supports block-stride transfer, and chaining DMA can bundle data transfers for each “Halo” plane
  - $ij$-plane: contiguous array
  - $ik$-plane: block stride
  - $jk$-plane: stride
- In each iteration, DMA descriptors can be reused and only a DMA kick operation is needed

Ultra low latency by PIO is useful for short messages in any kind of applications.

We will provide CUDA-like APIs as primitive, and a PGAS language “XMP-dev/TCA” for utilizing TCA effectively.
Future Work

- Offload functions in PEACH2
  - Reduction, etc.
- Prototype of PEACH3 is under development with PCIe Gen3 x8.
  - Altera Stratix V GX
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), 2.6 Gbyte/sec between GPUs
  - Min. Latency: 0.8 us (PIO), 1.8 us (DMA between CPUs), 2.3 us (DMA between GPUs)
  - GPU-GPU communication over the nodes can be utilized with 16 node sub-cluster.
  - By the ping-pong program, PEACH2 can achieve lower latency than existing technology, such as MVAPICH2 in small data size.