Center for Computational Sciences University of Tsukuba (site update)

Taisuke Boku

Director, Center for Computational Sciences

University of Tsukuba

CCS at University of Tsukuba

- Center for Computational Sciences
- Established in 1992
  - 12 years as Center for Computational Physics
  - Reorganized as Center for Computational Sciences in 2004
- Daily collaborative researches with two kinds of faculty members (43 in total)
  - Computational Scientists
    who have NEEDS (applications)
  - Computer Scientists
    who have SEEDS (system & solution)
- One of national supercomputer centers under MEXT, but we are Research Center (others are service centers)

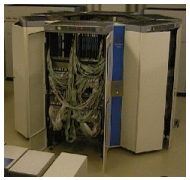
History of PAX/PACS series at U. Tsukuba

- 1977: research started by T. Hoshino and T. Kawai
- 1978: PACS-9 (with 9 nodes) completed
- 1996: CP-PACS, the first vendor-made supercomputer at CCS, ranked as #1 in TOP500

1978
1st gen: PACS-9

1980
2nd gen. PACS-32

1989
5th gen, QCDPAX



1996
6th gen: CP-PACS
Ranked #1 in TOP500

2006
7th gen: PACS-CS

2012~2013
8th gen: GPU cluster HA-PACS

Year	Name	Performance
1978	PACS-9	7 KFLOPS
1980	PACS-32	500 KFLOPS
1983	PAX-128	4 MFLOPS
1984	PAX-32J	3 MFLOPS
1989	QCDPAX	14 GFLOPS
1996	CP-PACS	614 GFLOPS
2006	PACS-CS	14.3 TFLOPS
2012~13	HA-PACS	1.166 PFLOPS
2014	COMA (PACS-IX)	1.001 PFLOPS

- *co-design* by computer scientists and computational scientists toward "practically high speed computer"
- Application-driven development
- Sustainable development experience

Three programs to share supercomputers at CCS

- MCRP – Multidisciplinary Cooperative Research Program
  - 50% of yearly node*hour, FREE (CCS covers)
  - peer review by CCS external committee
- HPCI – High Performance Computing Infrastructure
  - Japan's national program to share supercomputers under MEXT
  - 30% of yearly node*hour, FREE for users (MEXT pays)
  - peer review by HPCI project review committee
  - In 2020, special program for COVID-19 (5 projects at CCS systems)
- General Use
  - certain amount of charge per node*hour, but very cheap
  - 20% of node*hour
  - light review by CCS

- MCRP 2019 accepted 78 projects and >600 users for two supercomputers (Oakforest-PACS and Cygnus), and contributed >400 technical papers

Oakforest-PACS (OFP)

U. Tokyo convention　U. Tsukuba convention

The largest “Cluster” with KNL & OPA at first appearance

JCAHPC

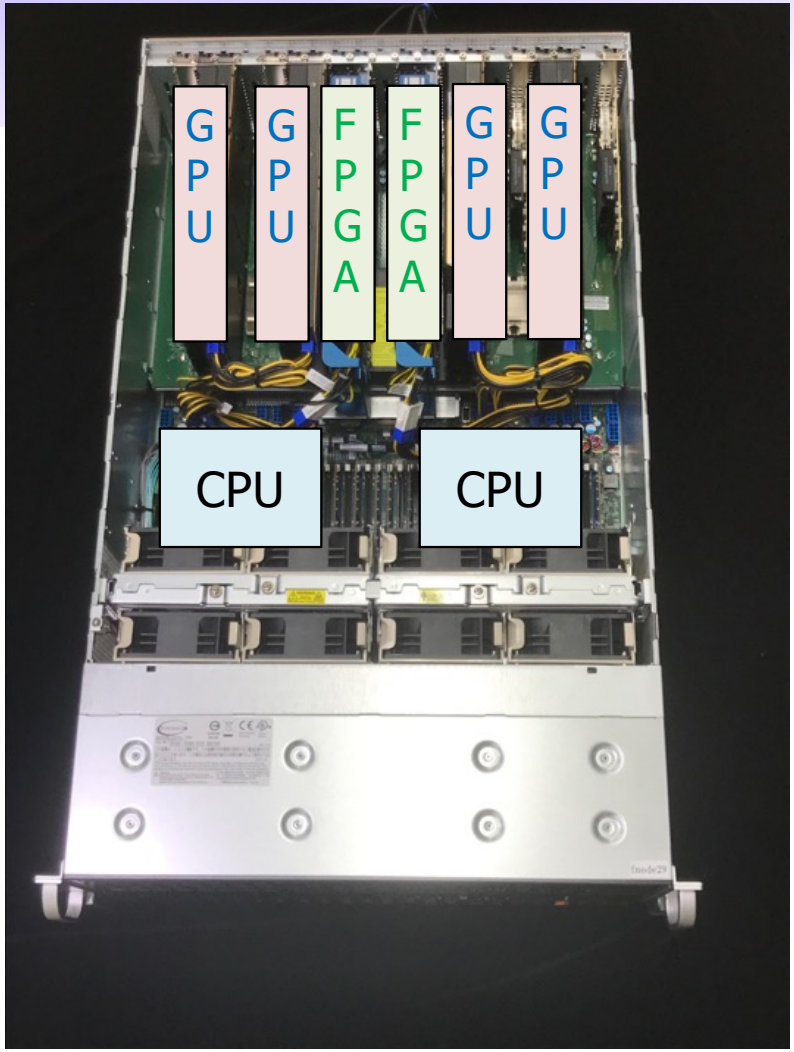
- JCAHPC: Joint Center for Advanced HPC
- virtual organization with U. Tsukuba and U. Tokyo to procure, manage and operate the OFP system
- OFP system is installed at Kashiwa Campus of U. Tokyo

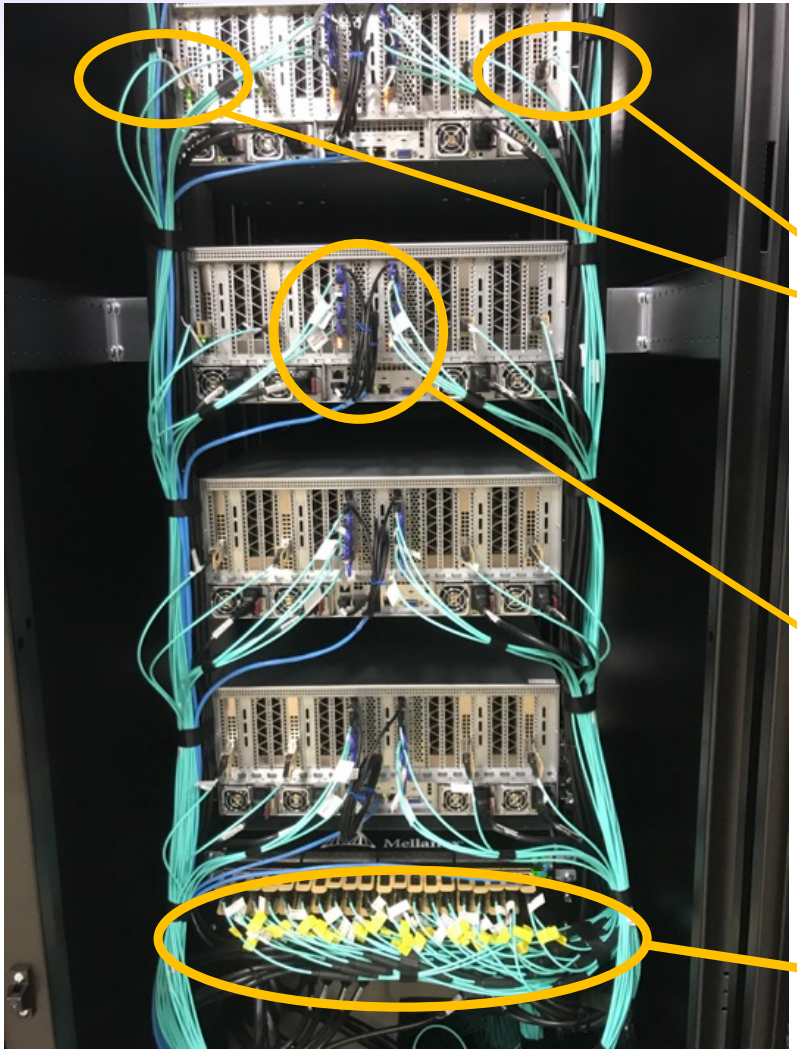
- 25 PFLOPS peak
- 8208 KNL CPUs (Xeon Phi 7250) = 558,144 cores
- FBB Fat-Tree by OmniPath
- HPL 13.55 PFLOPS
  World #6 as on Nov. 2016, #1 in Japan
- HPCG #3, Green500 #6
- 25PB DDN Luster HDD, 1PB Burst Buffer SSD
  (#1 in IO500 bandwidth)
- Fujitsu integration

- Full operation started Dec. 2016
- Official Program started on April 2017
  ⇒ will be shutdown on March 2022
  ⇒ planning to introduce post-OFP in early 2023

Cygnus (PACS-X) by U. Tsukuba only

The world first supercomputer with Multi-Hybrid (GPU + FPGA) Accelerating Architecture (not experimental one)



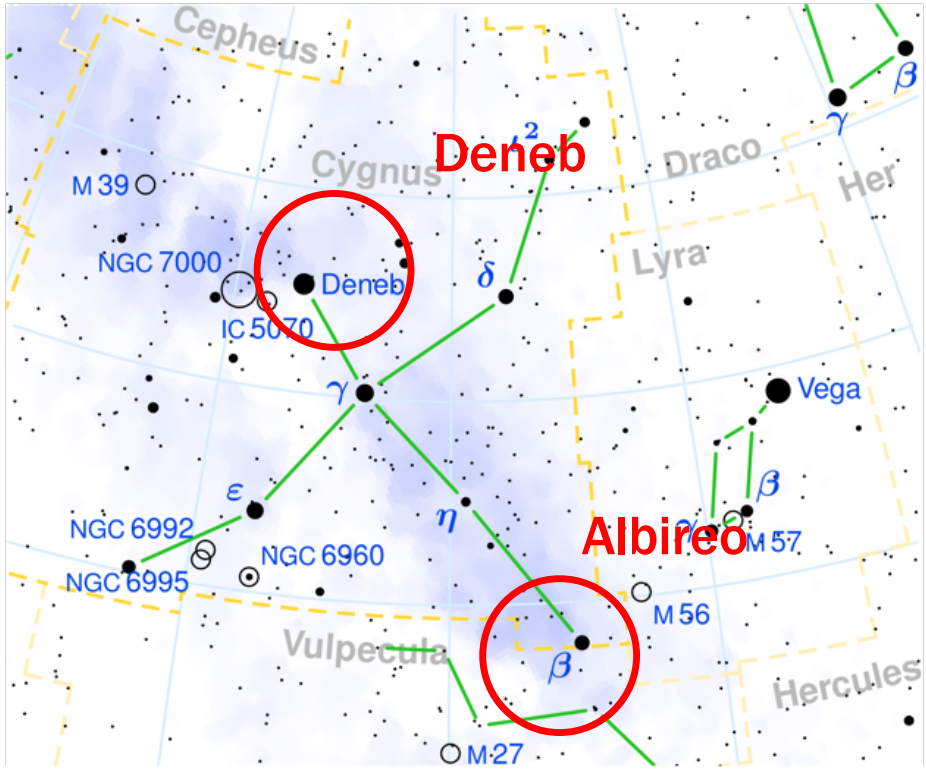


IB HDR100 x4
⇨ HDR200 x2

100Gbps x4
FPGA optical
network

IB HDR200
switch (for
full-bisection
Fat-Tree)

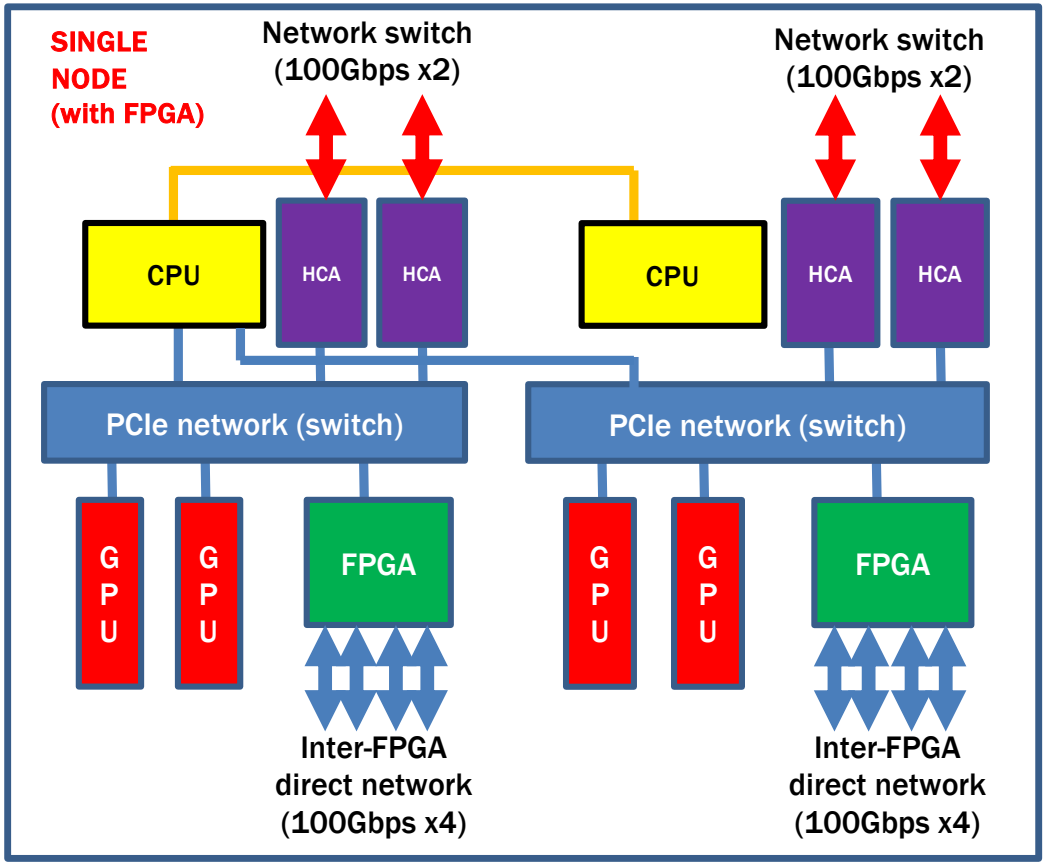
Cygnus Constellation



- Cygnus constellation has two major starts: Deneb and Albireo
- Deneb is alpha-star (largest)
- Albireo is beta-star and it is "double-star"
  ⇒ looks like two accelerators (GPU and FPGA) work together

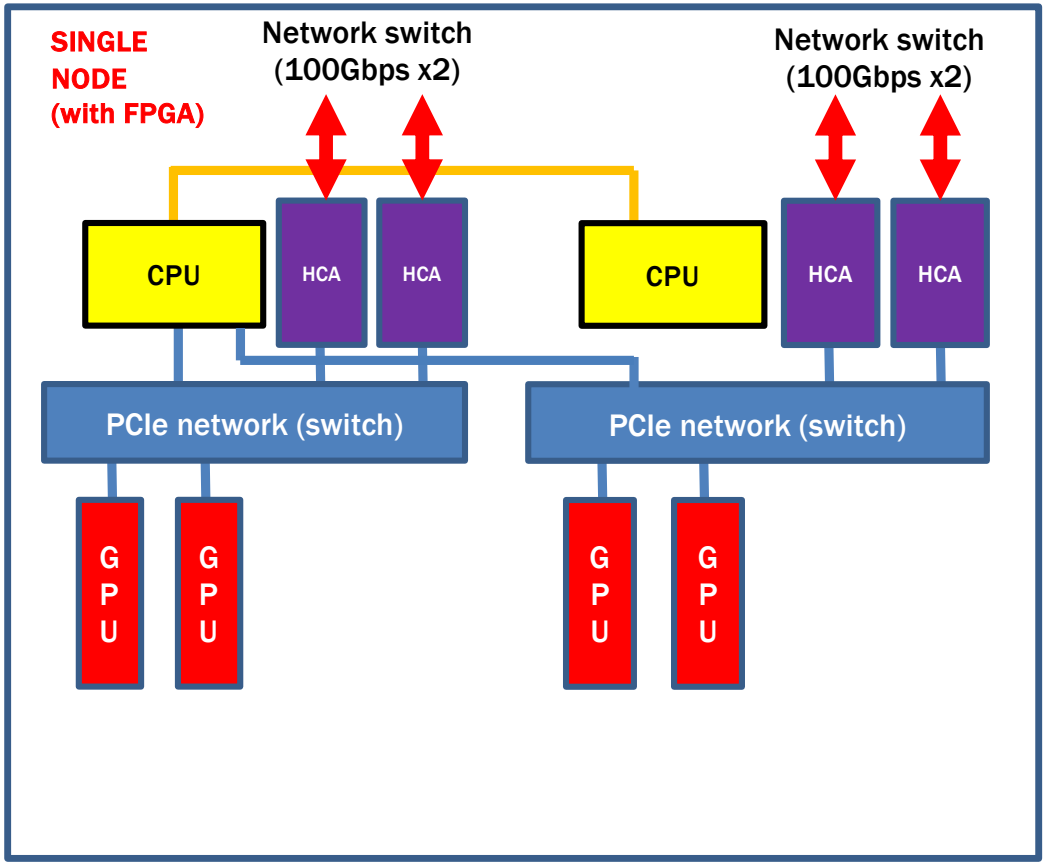
Single node configuration (Albireo)

- Each node is equipped with both IB EDR and FPGA-direct network
- Some nodes are equipped with both FPGAs and GPUs, and other nodes are with GPUs only



Single node configuration (Deneb)

- Each node is equipped with both IB EDR and FPGA-direct network
- Some nodes are equipped with both FPGAs and GPUs, and other nodes are with GPUs only

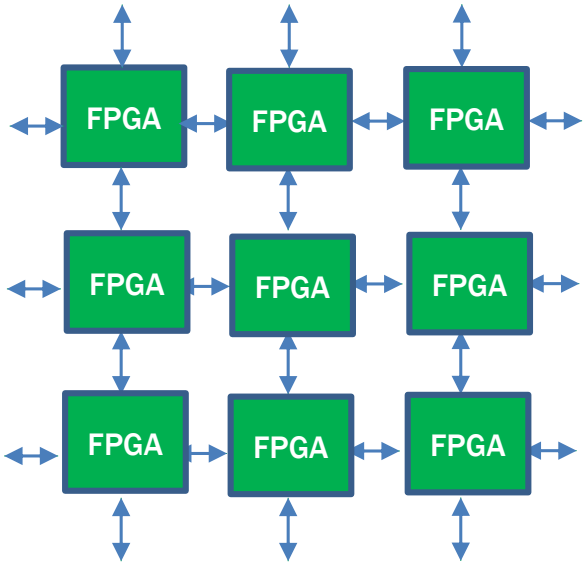


Specification of Cygnus

Item	Specification
Peak performance	2.4 PFLOPS DP (GPU: 2.24 PFLOPS, CPU: 0.16 PFLOPS + FPGA: 0.64 SP FLOPS) 1.6 PFLOPS Linpack (#265 at Jun. 2019)
# of nodes	81 (32 Albireo nodes, 49 Deneb nodes)
CPU / node	Intel Xeon Gold x2 sockets
GPU / node	NVIDIA Tesla V100 x4 (PCIe)
FPGA / node	Nallatech (Bittware) 520N with Intel Stratix10 x2 (each with 100Gbps x4 links)
NVMe	Intel NVMe 1.6TB, driven by NVMe-oF Target Offload
Global File System	DDN Lustre, RAID6, 2.5 PB
Interconnection Network	Mellanox InfiniBand HDR100 x4 = 400Gbps/node (SW=HDR200)
Total Network B/W	4 TB/s
Programming Language	CPU: C, C++, Fortran, OpenMP GPU: OpenACC, CUDA FPGA: OpenCL, Verilog HDL
MPI	MVAPICH2, IntelMPI with GDR + original FPGA-GPU-communication library
System Integrator	NEC

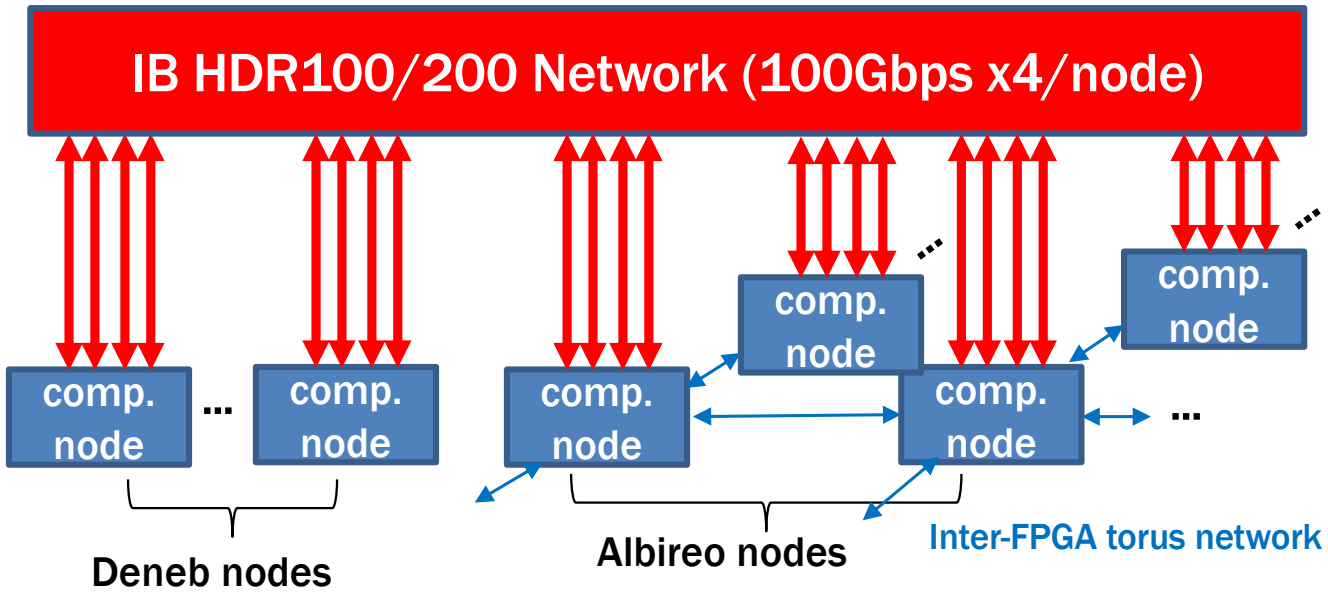
Two types of interconnection network

Inter-FPGA direct network (only for Albireo nodes)



64 of FPGAs on Albireo nodes (2 FPGAS/node) are connected by 8x8 2D torus network without switch

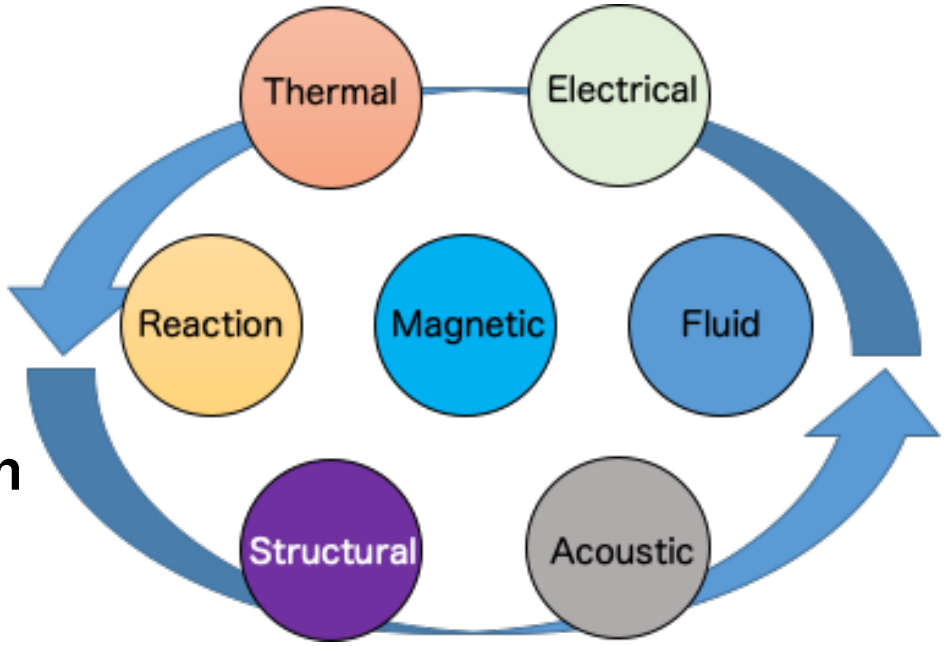
InfiniBand HDR100/200 network for parallel processing communication and shared file system access from all nodes



For all computation nodes (Albireo and Deneb) are connected by full-bisection Fat Tree network with 4 channels of InfiniBand HDR100 (combined to HDR200 switch) for parallel processing communication such as MPI, and also used to access to Lustre shared file system.

Why Multi-Hybrid ?

- Many multi-physical simulation to combine several sorts of different phenomena on a system is required in advanced physics
  - space – particle reaction
  - fluid dynamics with chemical reaction
  - macroscopic/microscopic hybrid simulation molecular simulation
- Characteristics and dynamism of parallelism drastically changes during the simulation
  - SIMD friendly or pipeline friendly
  - small fraction of low degree parallelism in a code makes “last one mile” under Amdahl‘s Law



CIRCUS = Computation + Communication on FPGA

- FPGA is possible to combine computation and communication in a single framework of pipelined data stream
  - loop computation is pipelined according to the index
  - all the computation part is implemented on logic elements except buffering on memory
  - possible to access IP by chip provides (ex. Intel) for optical link driving
- making all to be programmable on OpenCL
  - scientific users never write Verilog HDL -> perhaps OK with OpenCL
  - key issue for practical HPC cluster: OpenCL-enabled features such ash
    - FPGA communication link
    - GPU/FPGA DMA

CIRCUS: Communication Integrated Reconfigurable CompUting System

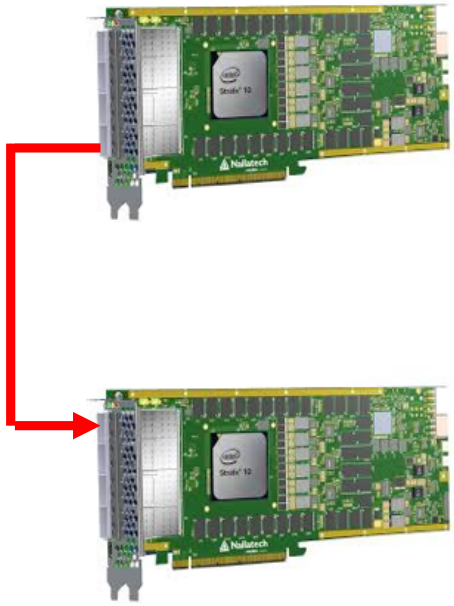
CoE user level programming

sender code on FPGA1

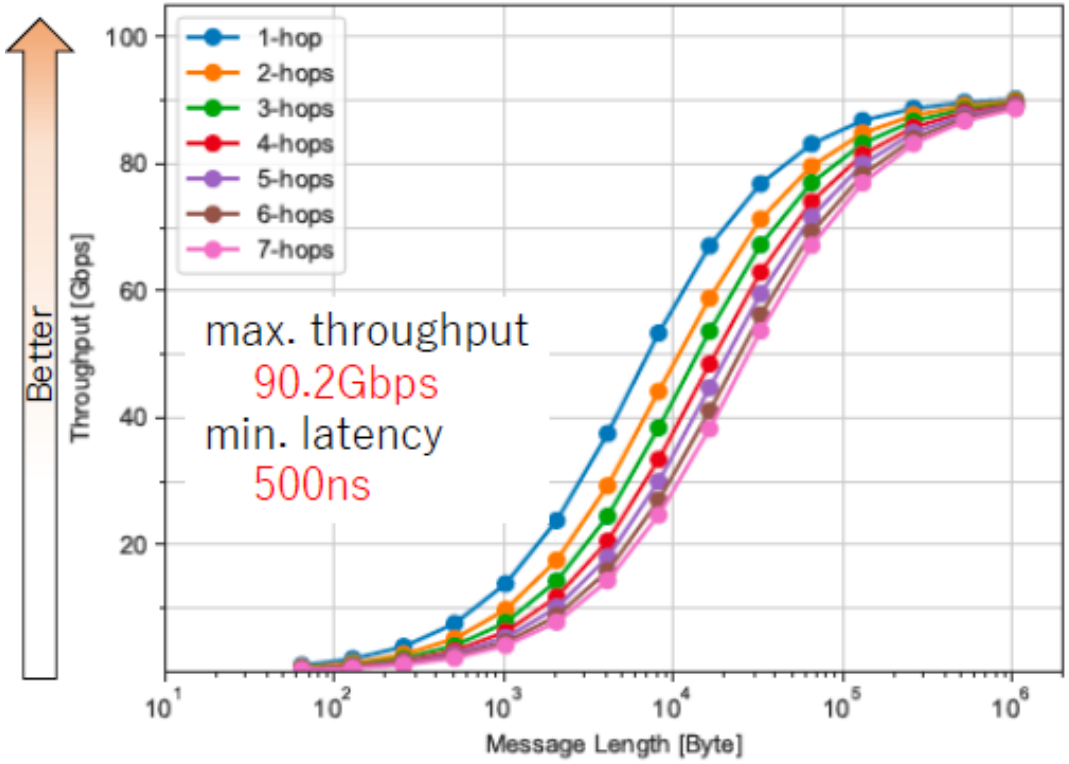
```
__kernel void sender(__global float* restrict x, int n) {
    for (int i = 0; i < n; i++) {
        float v = x[i];
        write_channel_intel(network_out, v);
    }
}
```

receiver code on FPGA2

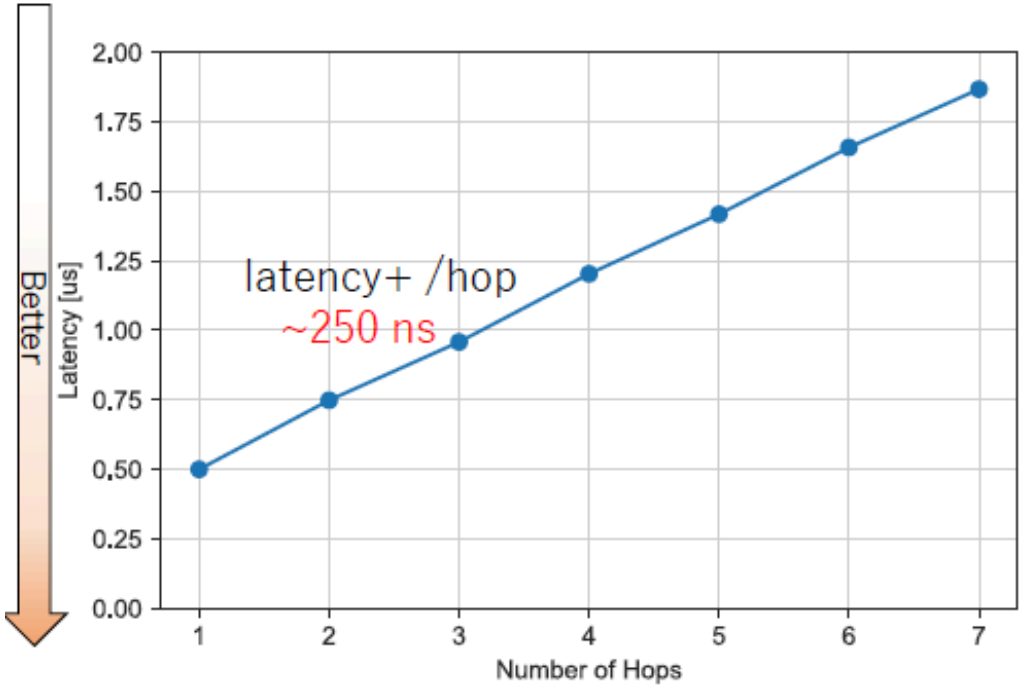
```
__kernel void receiver(__global float* restrict x, int n) {
    for (int i = 0; i < n; i++) {
        float v = read_channel_intel(network_in);
        x[i] = v;
    }
}
```



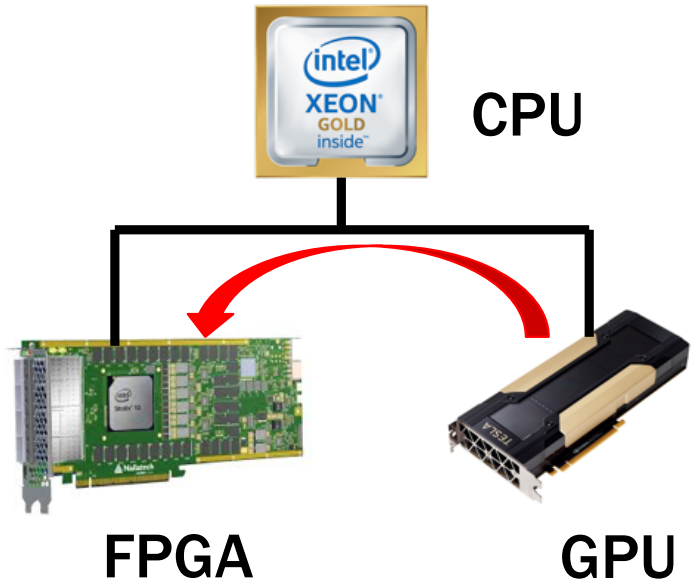
Communication bandwidth (on Stratirx10 by CIRCUS)



Throughput（1hop~7hops）



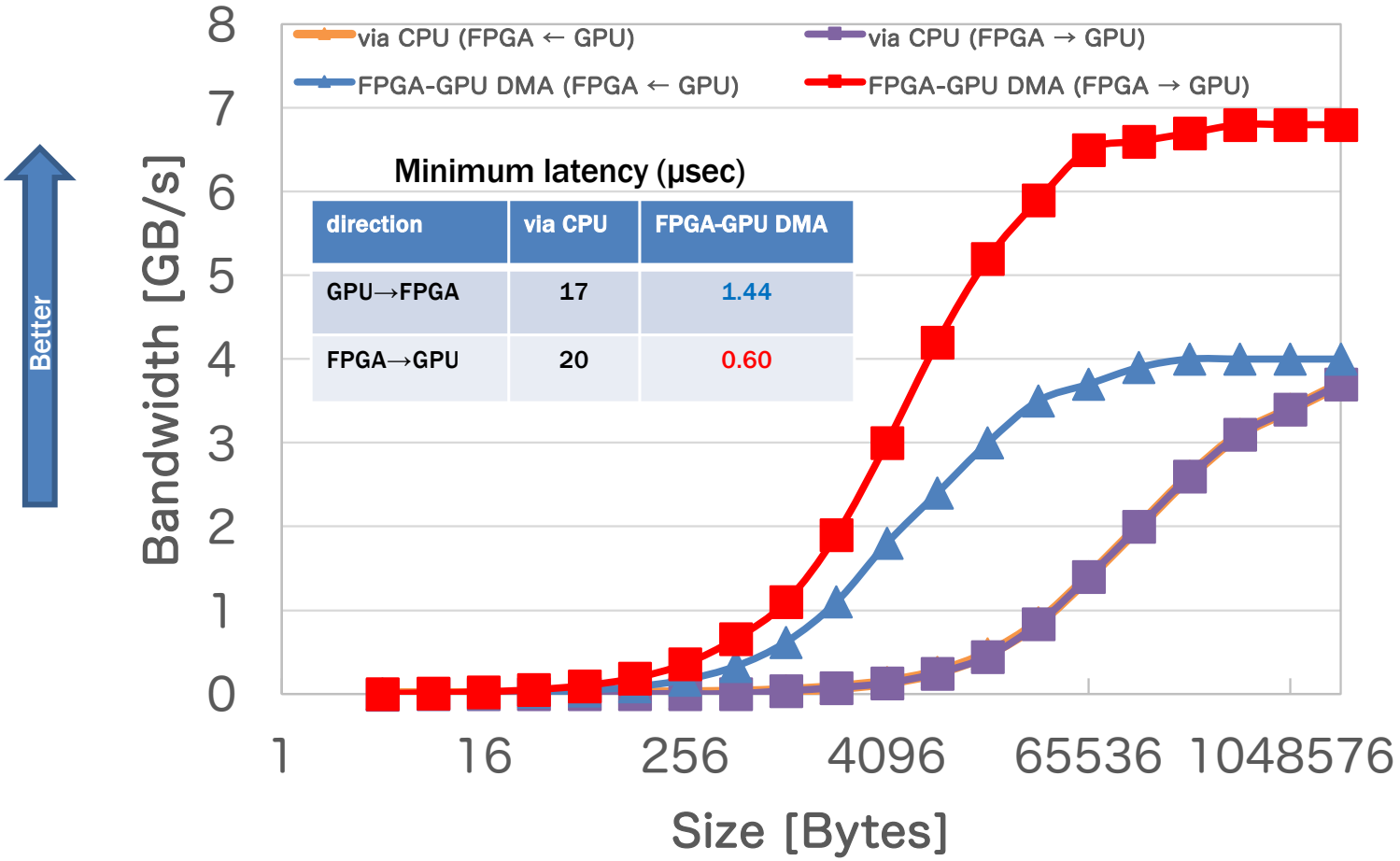
Latency（1hop~7hops）




```
__kernel void fpga_dma(__global float *restrict fpga_mem,
                       const ulong gpu_memadr,
                       const uint id_and_len)
{
  cldesc_t desc;
  // DMA transfer GPU -> FPGA
  desc.src = gpu_memadr;
  desc.dst = (ulong)(&fpga_mem[0]);
  desc.id_and_len = id_and_len;
  write_channel_intel(fpga_dma, desc);
  ulong status = read_channel_intel(dma_stat);
}
```

GPU-to-FPGA DMA kick

Communication Bandwidth (on Arria10 – V100)

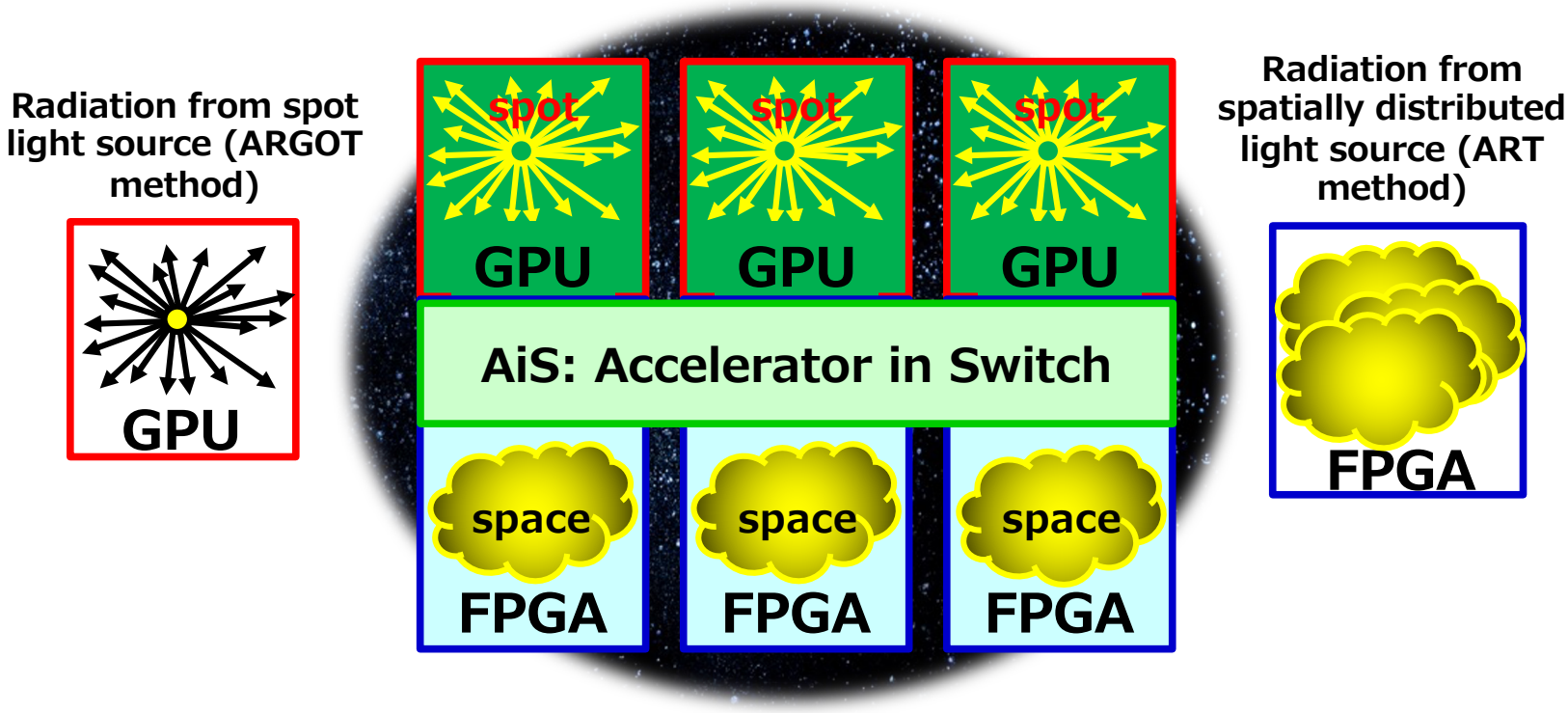


[Reference]

- Ryohei Kobayashi, Norihisa Fujita, Yoshiki Yamaguchi, Ayumi Nakamichi, Taisuke Boku, "GPU-FPGA Heterogeneous Computing with OpenCL-enabled Direct Memory Access", Proc. of Int. Workshop on Accelerators and Hybrid Exascale Systems (AsHES2019) in IPDPS2019 (to be published), May 20th, 2019.

ARGOT code: radiation transfer simulation

GPU works well on ARGOT method, but not efficient on ART method



GPU-only vs GPU-FPGA coworking