# Oakforest-PACS and PACS-X: Present and Future of CCS Supercomputers


Taisuke Boku

Deputy Director, Center for Computational Sciences

University of Tsukuba

# Oakforest-PACS



# JCAHPC



- Joint Center for Advanced High Performance Computing (http://jcahpc.jp)

- Very tight collaboration for “post-T2K” with two universities
  - For main supercomputer resources, *uniform specification* to ***single shared system***
  - Each university is financially responsible to introduce the machine and its operation
    -> unified procurement toward single system with ***largest scale in Japan***
  - To manage everything smoothly, a joint organization was established
    -> JCAHPC

# Oakforest-PACS (OFP)

U. Tokyo convention  U. Tsukuba convention

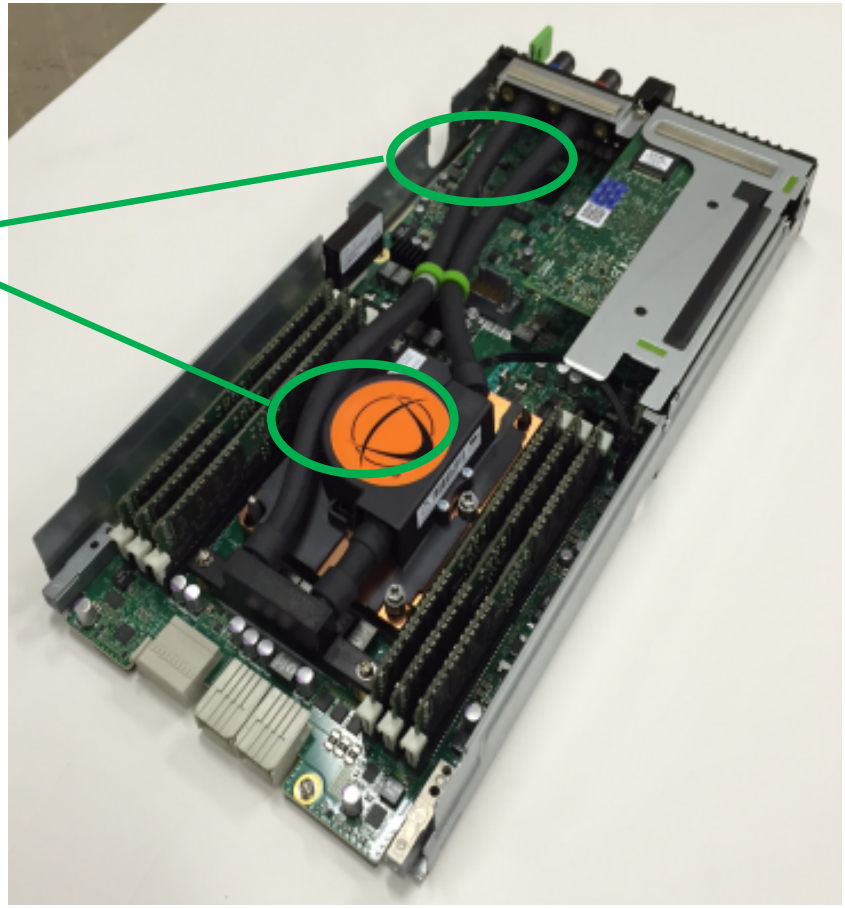
⇒ **Don't call it just "Oakforest" !**
**"OFP" is much better**


- 25 PFLOPS peak
- 8208 KNL CPUs
- FBB Fat-Tree by OmniPath
- HPL 13.55 PFLOPS
  #1 in Japan
  #6 in World
- HPCG #3 in World
- Green500 #6 in World

- Full operation started Dec. 2016
- Official Program started on April 2017

# Computation node & chassis

JCAHPC




Computation node (Fujitsu next generation PRIMERGY)
with single chip Intel Xeon Phi (Knights Landing, 3+TFLOPS)
and Intel Omni-Path Architecture card (100Gbps)

Chassis with 8 nodes, 2U size

# Water cooling pipes and IME (burst buffer)

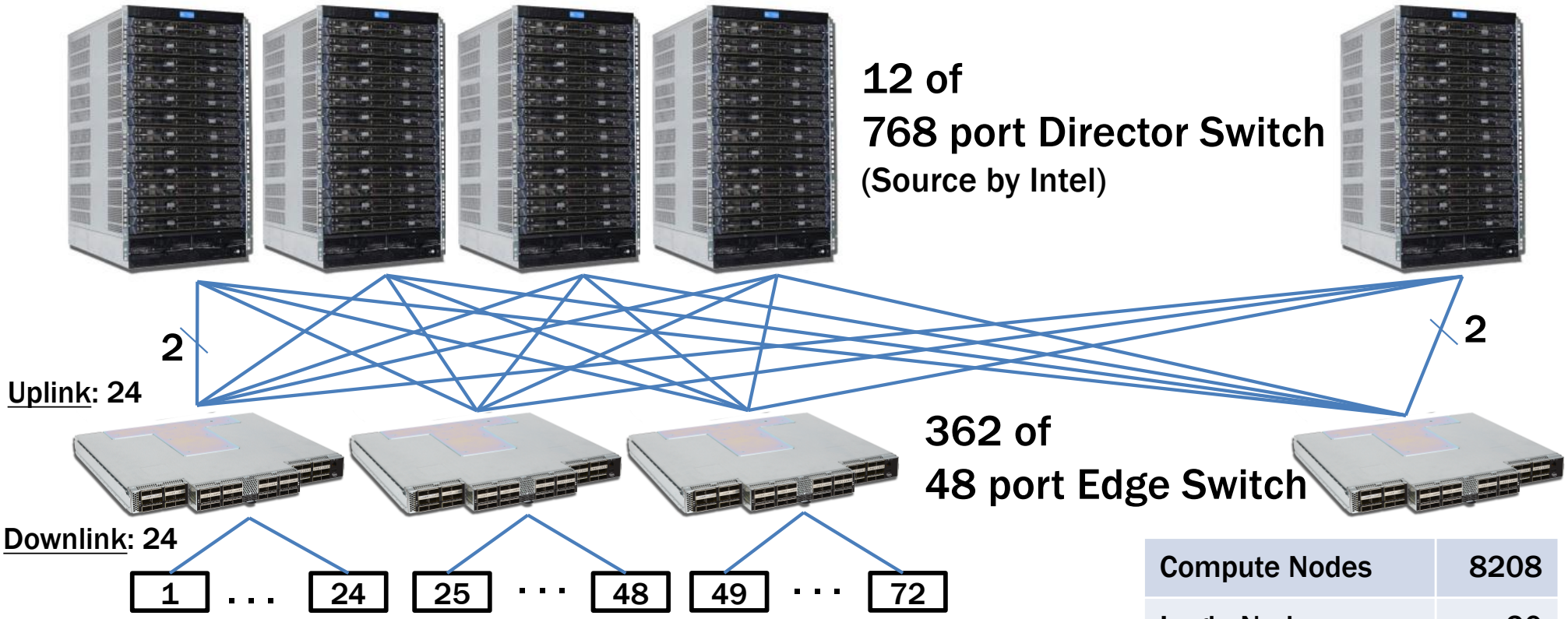
# Specification of Oakforest-PACS

| | | | |
|---|---|---|---|
| Total peak performance | | | 25 PFLOPS |
| Total number of compute nodes | | | 8,208 |
| Compute node | Product | | Fujitsu Next-generation PRIMERGY server for HPC (under development) |
| | Processor | | Intel® Xeon Phi™ (Knights Landing)<br>Xeon Phi 7250 (1.4GHz TDP) with 68 cores |
| | Memory | High BW | 16 GB, > 400 GB/sec (MCDRAM, effective rate) |
| | | Low BW | 96 GB, 115.2 GB/sec (DDR4-2400 x 6ch, peak rate) |
| Inter-connect | Product | | Intel® Omni-Path Architecture |
| | Link speed | | 100 Gbps |
| | Topology | | Fat-tree with full-bisection bandwidth |
| Login node | Product | | Fujitsu PRIMERGY RX2530 M2 server |
| | # of servers | | 20 |
| | Processor | | Intel Xeon E5-2690v4 (2.6 GHz 14 core x 2 socket) |
| | Memory | | 256 GB, 153 GB/sec (DDR4-2400 x 4ch x 2 socket) |

# Specification of Oakforest-PACS (I/O)

| | | | |
|---|---|---|---|
| Parallel File System | Type | | Lustre File System |
| | Total Capacity | | 26.2 PB |
| | Meta data | Product | DataDirect Networks MDS server + SFA7700X |
| | | # of MDS | 4 servers x 3 set |
| | | MDT | 7.7 TB (SAS SSD) x 3 set |
| | Object storage | Product | DataDirect Networks SFA14KE |
| | | # of OSS (Nodes) | 10 (20) |
| | | Aggregate BW | ~500 GB/sec |
| Fast File Cache System | Type | | Burst Buffer, Infinite Memory Engine (by DDN) |
| | Total capacity | | 940 TB (NVMe SSD, including parity data by erasure coding) |
| | Product | | DataDirect Networks IME14K |
| | # of servers (Nodes) | | 25 (50) |
| | Aggregate BW | | ~1,560 GB/sec |

# Full bisection bandwidth Fat-tree by Intel® Omni-Path Architecture




Firstly, to reduce switches&cables, we considered :
- All the nodes into subgroups are connected with FBB Fat-tree
- Subgroups are connected with each other with >20% of FBB

But, HW quantity is not so different from globally FBB, and globally FBB is preferredfor flexible job management.

| | |
|---|---|
| Compute Nodes | 8208 |
| Login Nodes | 20 |
| Parallel FS | 64 |
| IME | 300 |
| Mgmt, etc. | 8 |
| **Total** | **8600** |

# Facility of Oakforest-PACS system

JCAHPC

| | | | |
|---|---|---|---|
| Power consumption | | | 4.2 MW (including cooling)<br>→ actually 3.2MW (in average) |
| # of racks | | | 102 |
| Cooling system | Compute Node | Type | Warm-water cooling<br>Direct cooling (CPU)<br>Rear door cooling (except CPU) |
| | | Facility | Cooling tower & Chiller |
| | Others | Type | Air cooling |
| | | Facility | PAC |

# Software of Oakforest-PACS

JCAHPC

| | Compute node | Login node |
|---|---|---|
| OS | CentOS 7, McKernel | Red Hat Enterprise Linux 7 |
| Compiler | gcc, Intel compiler (C, C++, Fortran) | |
| MPI | Intel MPI, MVAPICH2 | |
| Library | Intel MKL<br>LAPACK, FFTW, SuperLU, PETSc, METIS, Scotch, ScaLAPACK, GNU Scientific Library, NetCDF, Parallel netCDF, Xabclib, ppOpen-HPC, ppOpen-AT, MassiveThreads | |
| Application | mpijava, XcalableMP, OpenFOAM, ABINIT-MP, PHASE system, FrontFlow/blue, FrontISTR, REVOCAP, OpenMX, xTAPP, AkaiKKR, MODYLAS, ALPS, feram, GROMACS, BLAST, R packages, Bioconductor, BioPerl, BioRuby | |
| Distributed FS | | Globus Toolkit, Gfarm |
| Job Scheduler | Fujitsu Technical Computing Suite | |
| Debugger | Allinea DDT | |
| Profiler | Intel VTune Amplifier, Trace Analyzer & Collector | |

# TOP500 list on Nov. 2016 (first appearance)

| # | Machine | Architecture | Country | Rmax (TFLOPS) | Rpeak (TFLOPS) | MFLOPS/W |
|---|---|---|---|---|---|---|
| 1 | TaihuLight, NSCW | MPP (Sunway, SW26010) | China | 93,014.6 | 125,435.9 | 6051.3 |
| 2 | Tianhe-2 (MilkyWay-2), NSCG | Cluster (NUDT, CPU + KNC) | China | 33,862.7 | 54,902.4 | 1901.5 |
| 3 | Titan, ORNL | MPP (Cray, XK7: CPU + GPU) | United States | 17,590.0 | 27,112.5 | 2142.8 |
| 4 | Sequoia, LLNL | MPP (IBM, BlueGene/Q) | United States | 17,173.2 | 20,132.7 | 2176.6 |
| 5 | Cori, NERSC-LBNL | MPP (Cray, XC40: KNL) | United States | 14,014.7 | 27,880.7 | ??? |
| 6 | Oakforest-PACS, JCAHPC | Cluster (Fujitsu, KNL) | Japan | 13,554.6 | 25,004.9 | 4985.1 |
| 7 | K Computer, RIKEN AICS | MPP (Fujitsu) | Japan | 10,510.0 | 11,280.4 | 830.2 |
| 8 | Piz Daint, CSCS | MPP (Cray, XC50: CPU + GPU) | Switzerland | 9,779.0 | 15,988.0 | 7453.5 |
| 9 | Mira, ANL | MPP (IBM, BlueGene/Q) | United States | 8,586.6 | 10,066.3 | 2176.6 |
| 10 | Trinity, NNSA/ LABNL/SNL | MPP (Cray, XC40: MIC) | United States | 8,100.9 | 11,078.9 | 1913.7 |

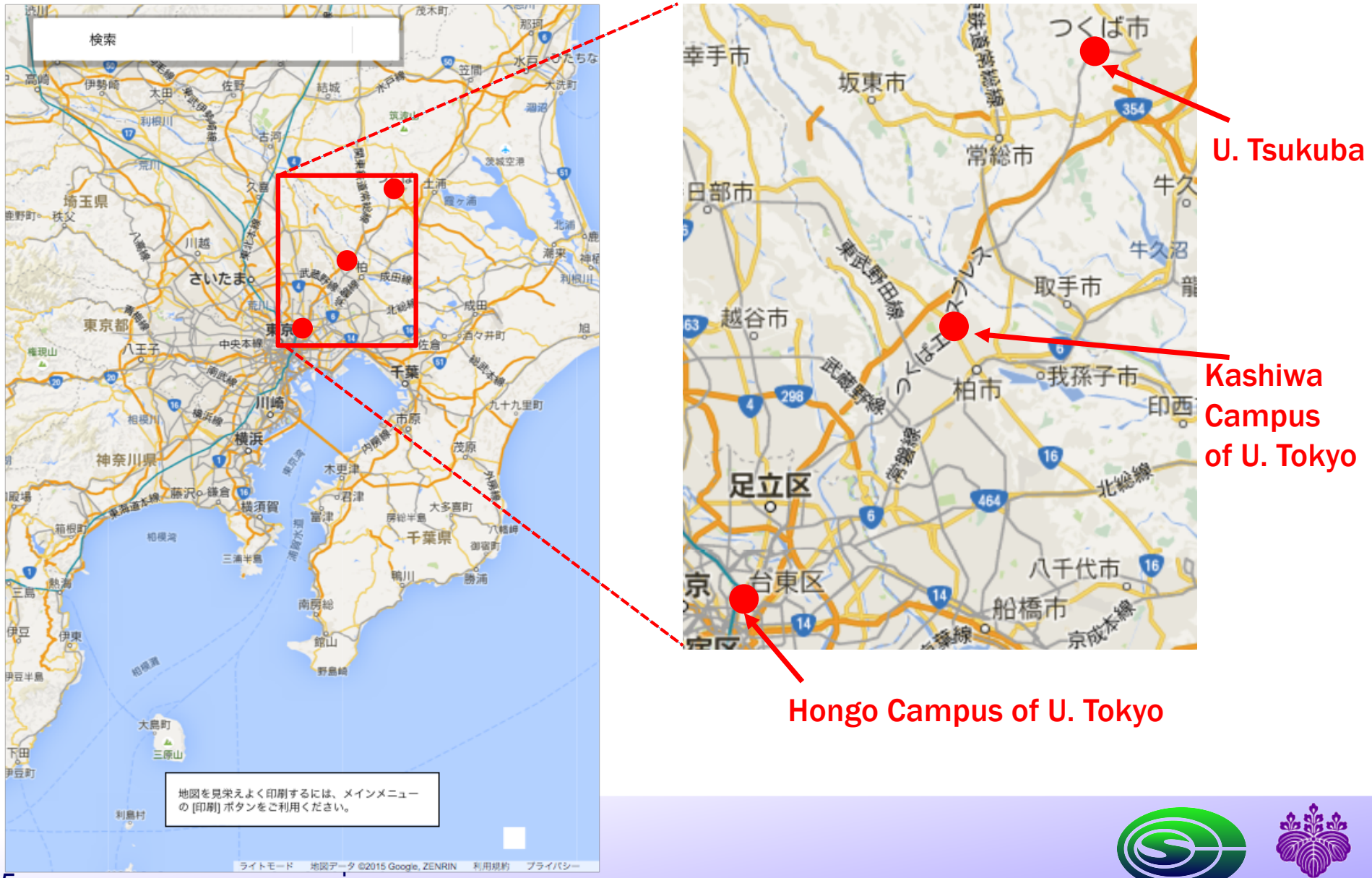
# HPCG on Nov. 2016 (first appearance)

| Rank | Site | Computer | Cores | Rmax Pflops | HPCG Pflops | HPCG /HPL | % of Peak |
|---|---|---|---|---|---|---|---|
| 1 | RIKEN Advanced Institute for Computational Science | K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect | 705,024 | 10.5 | 0.603 | 5.7% | 5.3% |
| 2 | NSCC / Guangzhou | Tianhe-2 NUDT, Xeon 12C 2.2GHz + Intel Xeon Phi 57C + Custom | 3,120,000 | 33.8 | 0.580 | 1.7% | 1.1% |
| 3 | Joint Center for Advanced High Performance Computing Japan | Oakforest-PACS – PRIMERGY CX600 M1, Intel Xeon Phi 7250 68C 1.4GHz, Intel OmniPath, Fujitsu | 557,056 | 24.9 | 0.385 | 2.8% | 2.8% |
| 4 | National Supercomputing Center in Wuxi, China | Sunway TaihuLight – Sunway MPP, SW26010 260C 1.45GHz, Sunway, NRCPC | 10,649,600 | 93.0 | 0.3712 | 0.4% | 0.3% |
| 5 | DOE/SC/LBNL/NERSC USA | Cori – XC40, Intel Xe Cray Aries, Cray | | | | | |
| 6 | DOE/NNSA/LLNL USA | Sequoia – IBM Blue 16C 1.6GHz, 5D Toru | | | | | |
| 7 | DOE/SC/Oak Ridge Nat Lab | Titan - Cray XK7 , Op 2.200GHz, Cray Gem NVIDIA K20x | | | | | |
| 8 | DOE/NNSA/LANL/SNL | Trinity - Cray XC40, I custom, Cray | | | | | |
| 9 | NASA / Mountain View | Pleiades - SGI ICE X, 2680v2, E5-2680v3, FDR, HPE/SGI | | | | | |
| 10 | DOE/SC/Argonne National Laboratory | Mira - BlueGene/Q, 1.60GHz, 5D Torus, | | | | | |

# OFP resource sharing program (nation-wide)

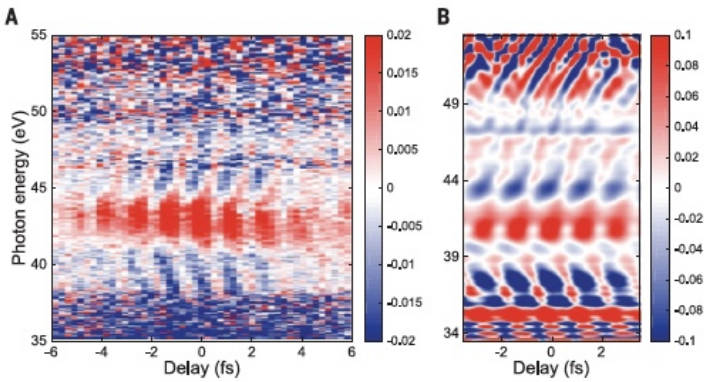
- JCAHPC (20%)
  - HPCI – HPC Infrastructure program in Japan to share all supercomputers (free!)
  - Big challenge special use (full system size)
- U. Tsukuba (23.5%)
  - Interdisciplinary Academic Program (free!)
  - Large scale general use
- U. Tokyo (56.5%)
  - General use
  - Industrial trial use
  - Educational use
  - Young & Female special use

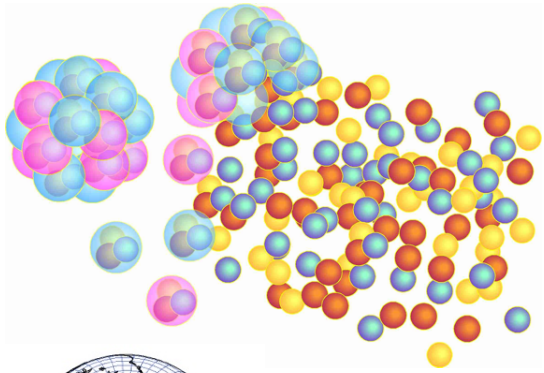
# Machine location: Kashiwa Campus of U. Tokyo

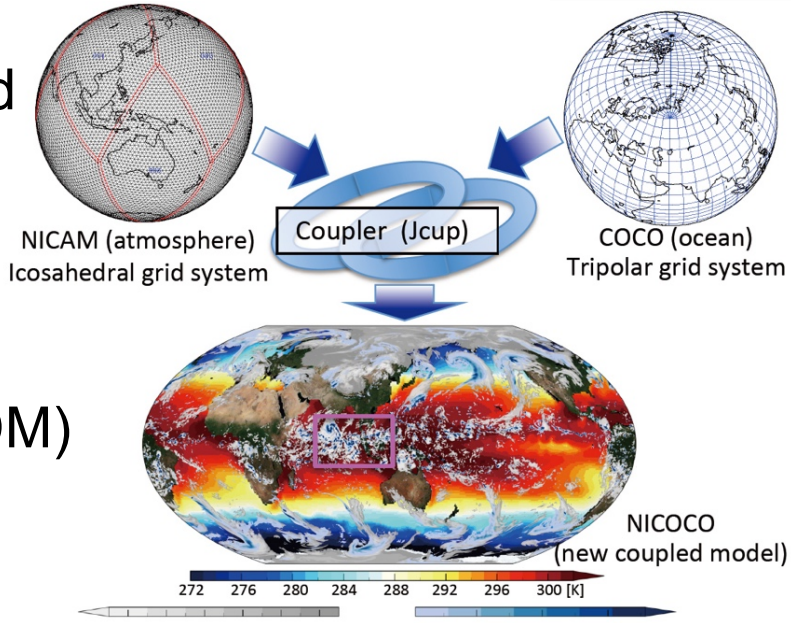


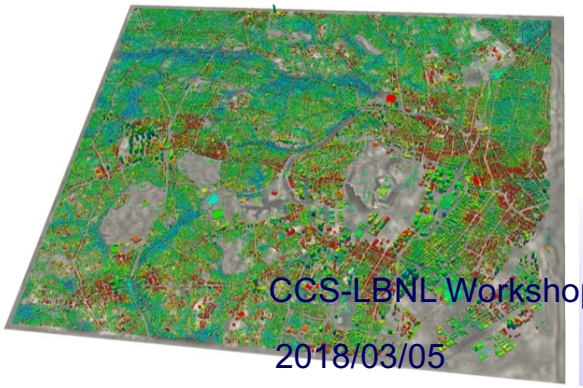
# Oakforest-PACS

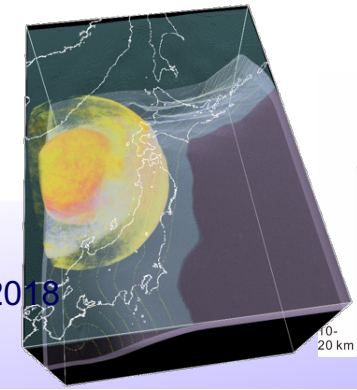
- **SALMON/ARTED**
  - Electron Dynamic
- **Lattice QCD**
  - Particle Physics
- **NICAM-COCO**
  - Atmosphere/Ocean combined
- **GHYDRA**
  - 3D Earth Quake (FEM)
- **Seism3D/OpenSWPC**
  - 3D Wide Region Seismic (FDM)

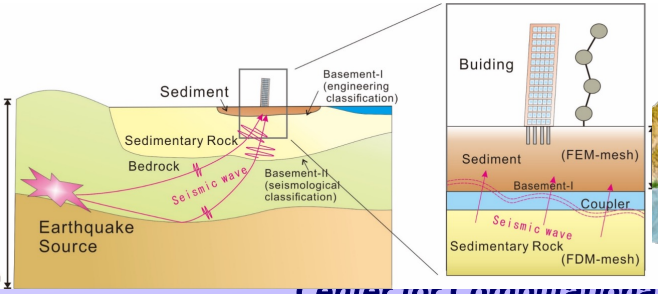


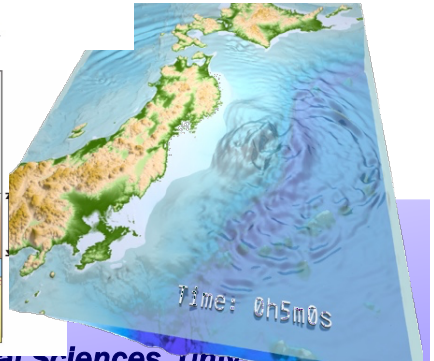





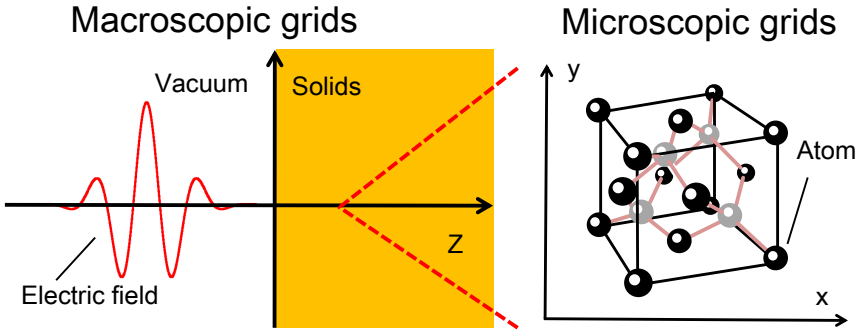


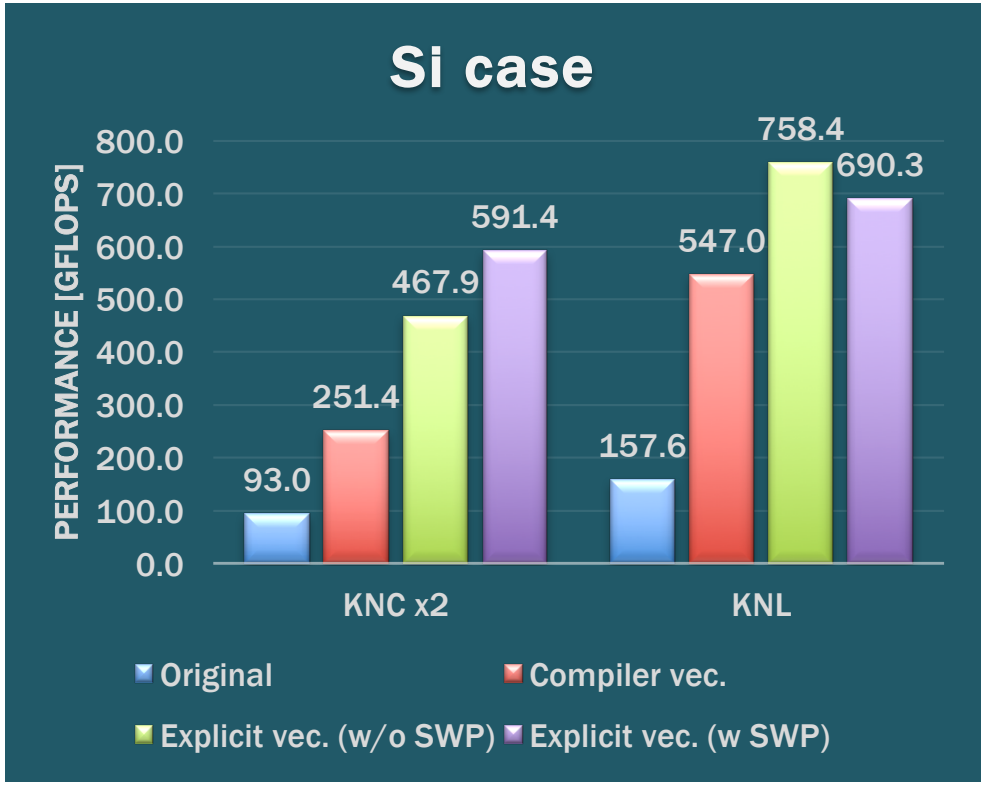
# Xeon Phi tuning on ARTED (with Y. Hirokawa under collaboration with Prof. K. Yabana, CCS) → SALMON now

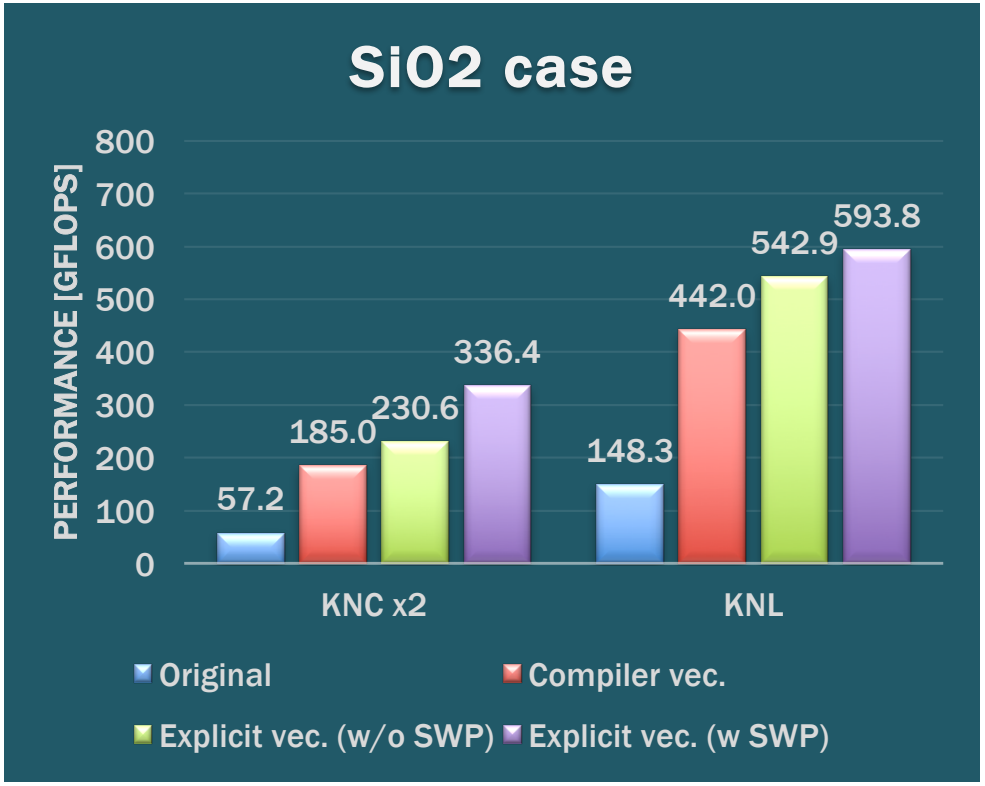
- ARTED – Ab-initio Real-Time Electron Dynamics simulator
- Multi-scale simulator based on RTRSDFT (Real-Time Real-Space Density Functional Theory) developed in CCS, U. Tsukuba to be used for Electron Dynamics Simulation
  - RSDFT : basic status of electron (no movement of electron)
  - RTRSDFT : electron state under external force
- In RTRSDFT, RSDFT is used for ground state
  - RSDFT : large scale simulation with 1000～10000 atoms (ex. K-Computer)
  - RTRSDFT : calculate a number of unit-cells with 10 ~ 100 atoms




RSDFT : Real-Space Density Functional Theory
RTRSDFT : Real-Time RSDFT

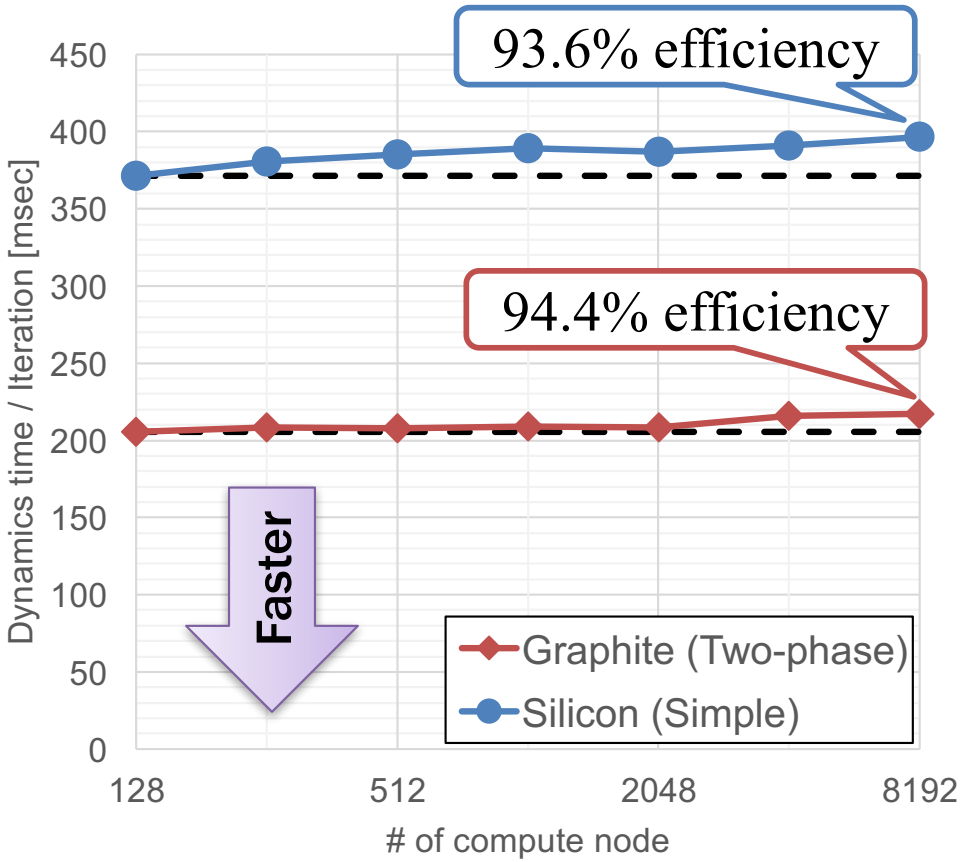
# Stencil computation (3D) performance

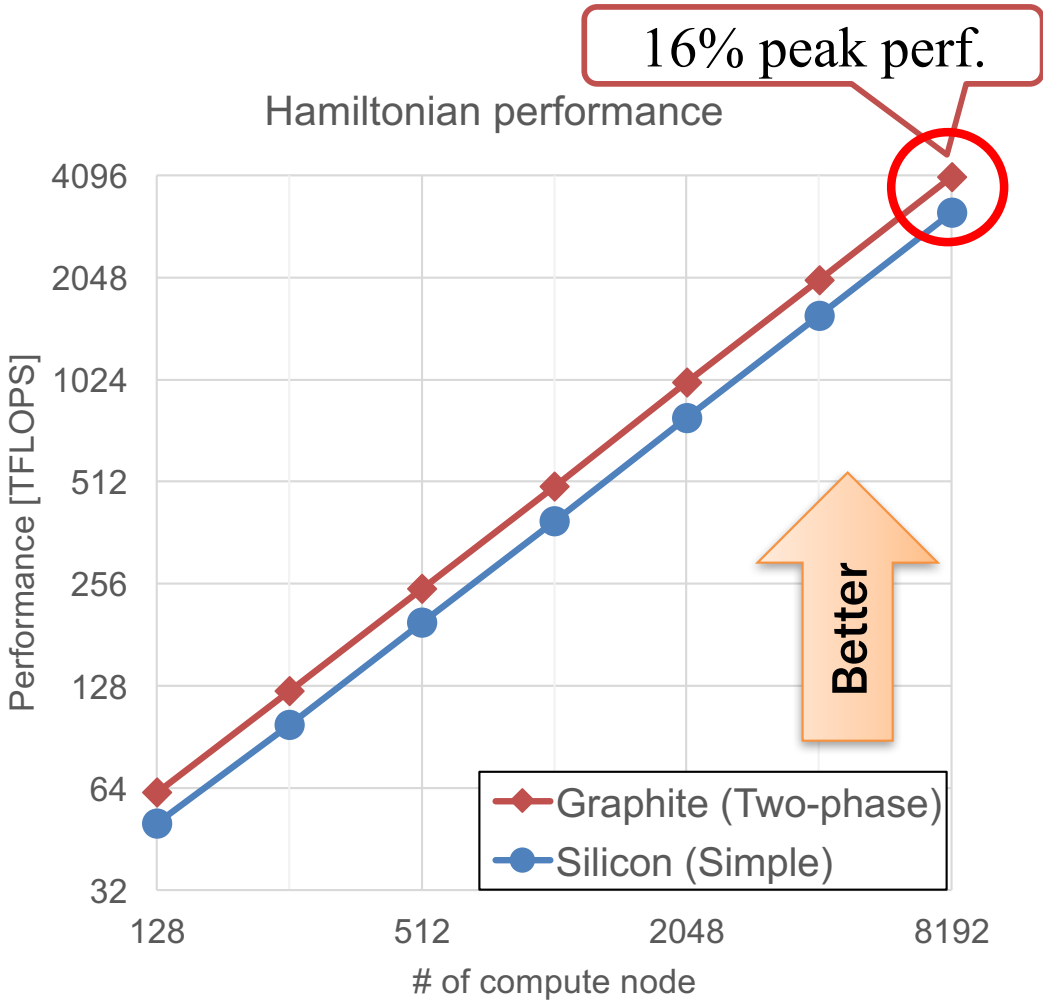



>2x faster than KNC (at maximum) -> up to 25% of theoretical peak of KNL

# Weak Scaling



# PACS-X

# Accelerators in HPC

- Traditionally...
  - Cell Broadband Engine, ClearSpeed, GRAPE. ...
  - then GPU (most popular)
- Is GPU perfect ?
  - good for many applications (replacing vector machines)
  - depending on very wide and regular computation
    - large scale SIMD (STMD) mechanism in a chip
    - high bandwidth memory (GDR5, HBM) and local memory
  - bad for
    - not enough parallelism
    - not regular computation (warp spliting)
    - frequent inter-node communication (kernel switch, go back to CPU)

# FPGA in HPC

- Goodness of recent FPGA for HPC
  - True codesigning with applications (essential)
  - Programmability improvement: OpenCL, other high level languages
  - High performance interconnect: 40Gb~100Gb
  - Precision control is possible
  - Relatively low power
- Problems
  - Programmability: OpenCL is not enough, not efficient
  - Low standard FLOPS: still cannot catch up to GPU
    -> “never try what GPU works well on”
  - Memory bandwidth: 2-gen older than high end CPU/GPU
    -> be improved by HBM (Stratix10)

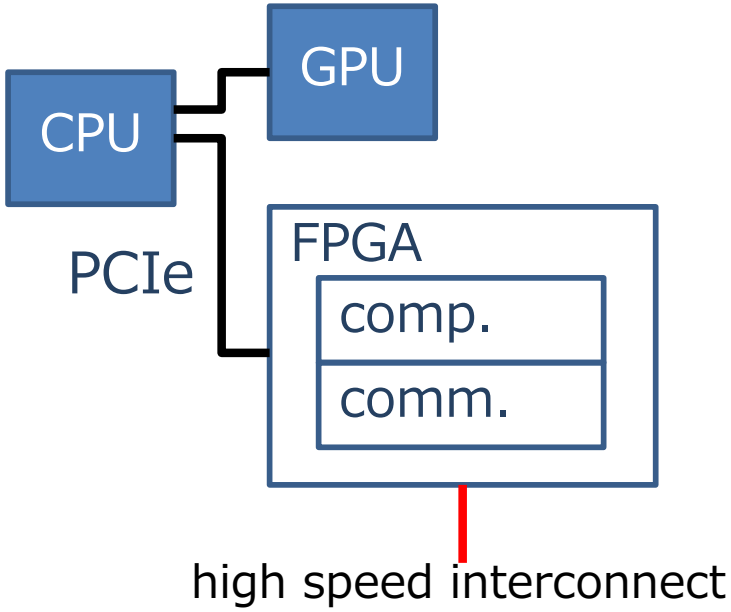
# Simple pros/cons

| | performance (FLOPS) | external communication (sec, B/s) | programming cost |
|---|---|---|---|
| CPU | △ | ○ | ◎ |
| GPU | ◎ | △ | ○ |
| FPGA | ○ | ◎ | ×→△? |

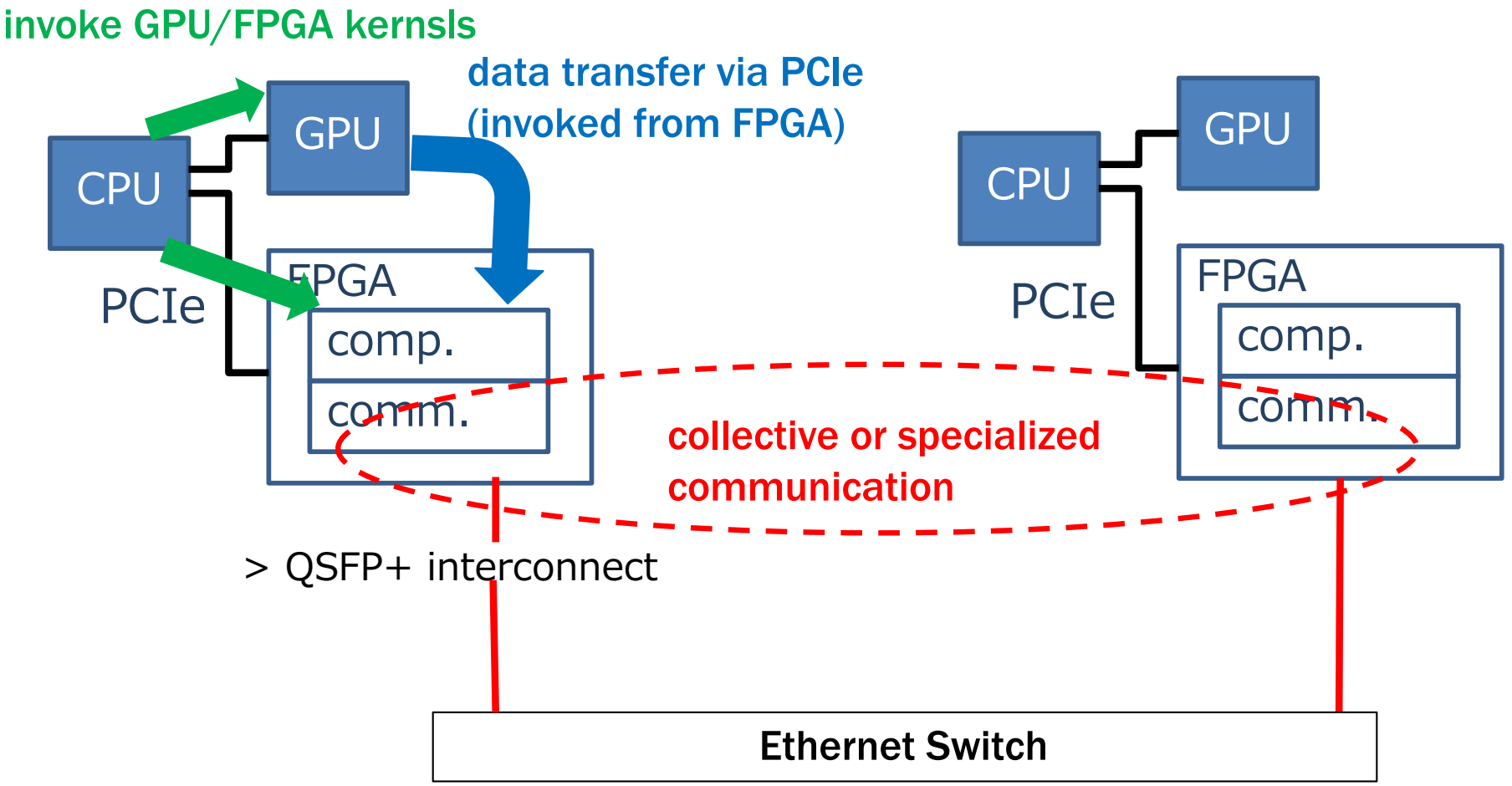
**How to compensate with each other toward large degree of strong scaling ?**

# AiS

- AiS: Accelerator in Swtich
  - Using FPGA not only for computation offloading but also for communication
  - Combining computation offloading and communication among FPGAs for ultra-low latency on FPGA computing
  - Especially effective on communication-related small/medium computation (such as collective communication)
  - Covering GPU non-suited computation by FPGA
  - OpenCL-enable programming for application users



# AiS computation model

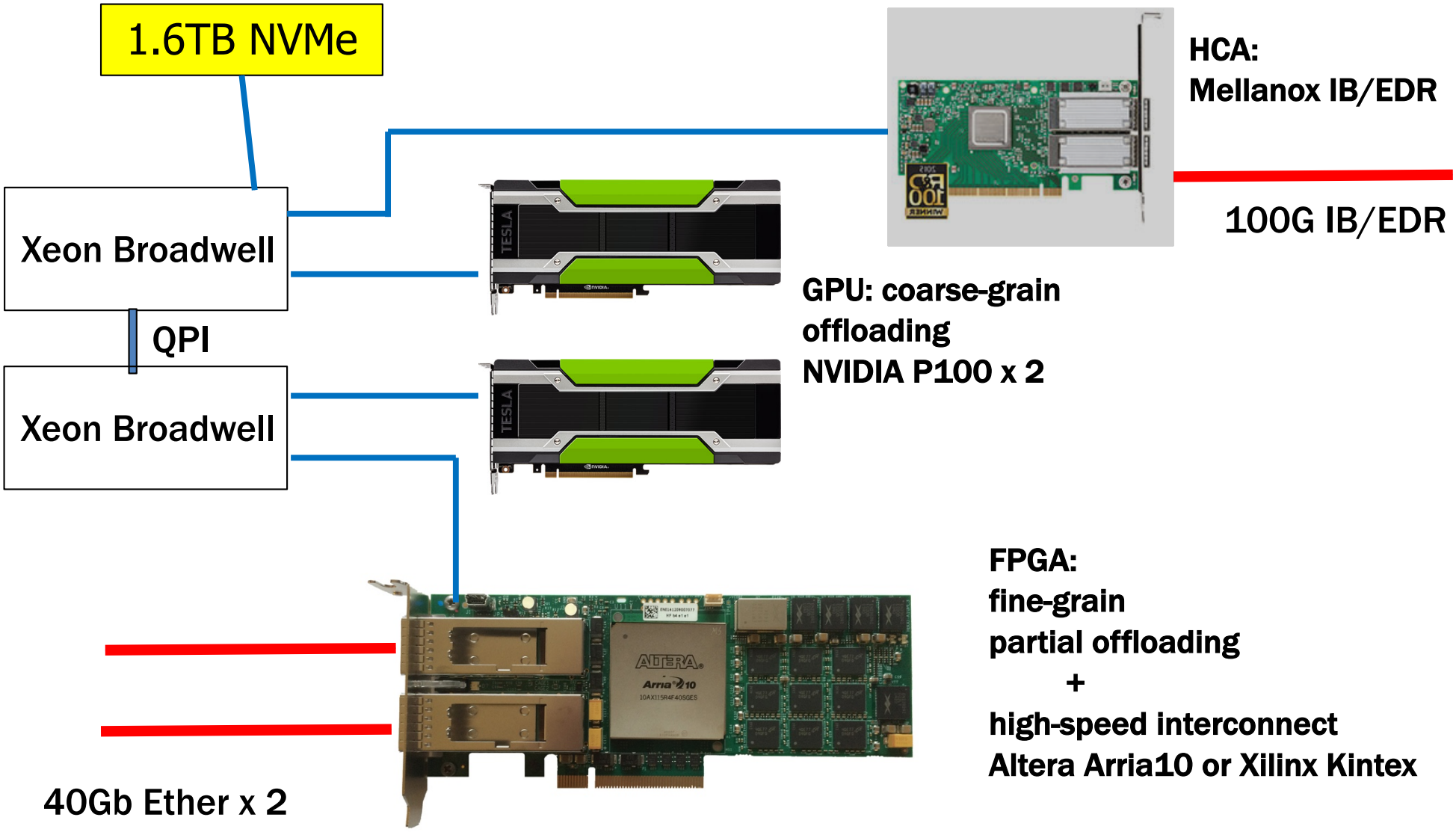


# PACS-X (ten) Project at CCS, U. Tsukuba

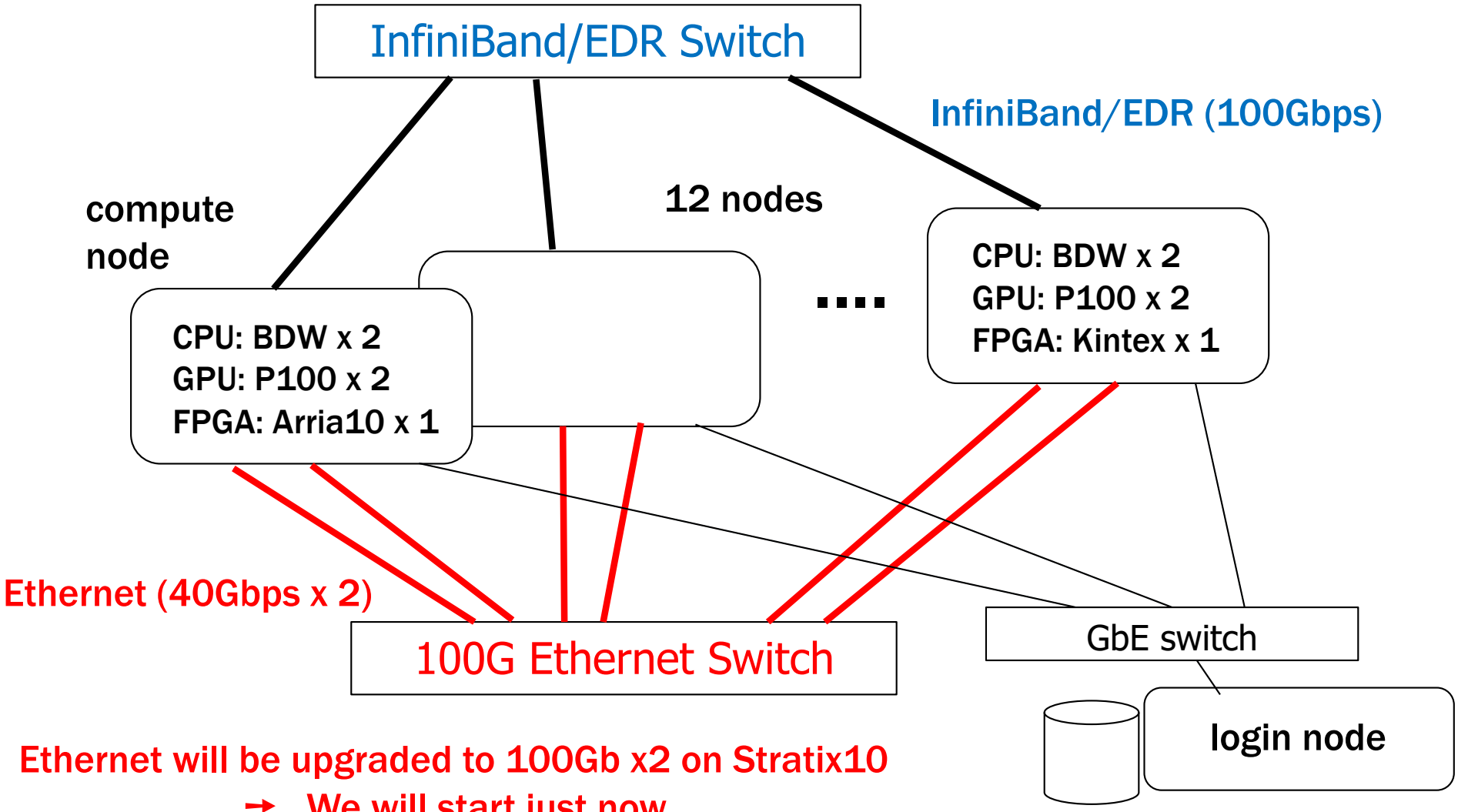
- PACS (Parallel Advanced system for Computational Sciences)
  - a series of co-design base parallel system development both on system and application at U. Tsukuba (1978~)
  - recent systems focus on accelerators
    - PACS-VIII: HA-PACS (GPU cluster, Fermi+Kepler, PEACH2, 1.1PFLOPS)
    - PACS-IX: COMA (MIC cluster, KNC, 1PFLOPS)
- Next generation of TCA implementation
  - PEACH2 with PCIe is old and with several limitation
  - new generation of GPU and FPGA with high speed interconnection
  - more tightly co-designing with applications
  - system deployment started from 2016

➡ **PPX: Pre-PACS-X**

# PPX: testbed under AiS concept (12 nodes)



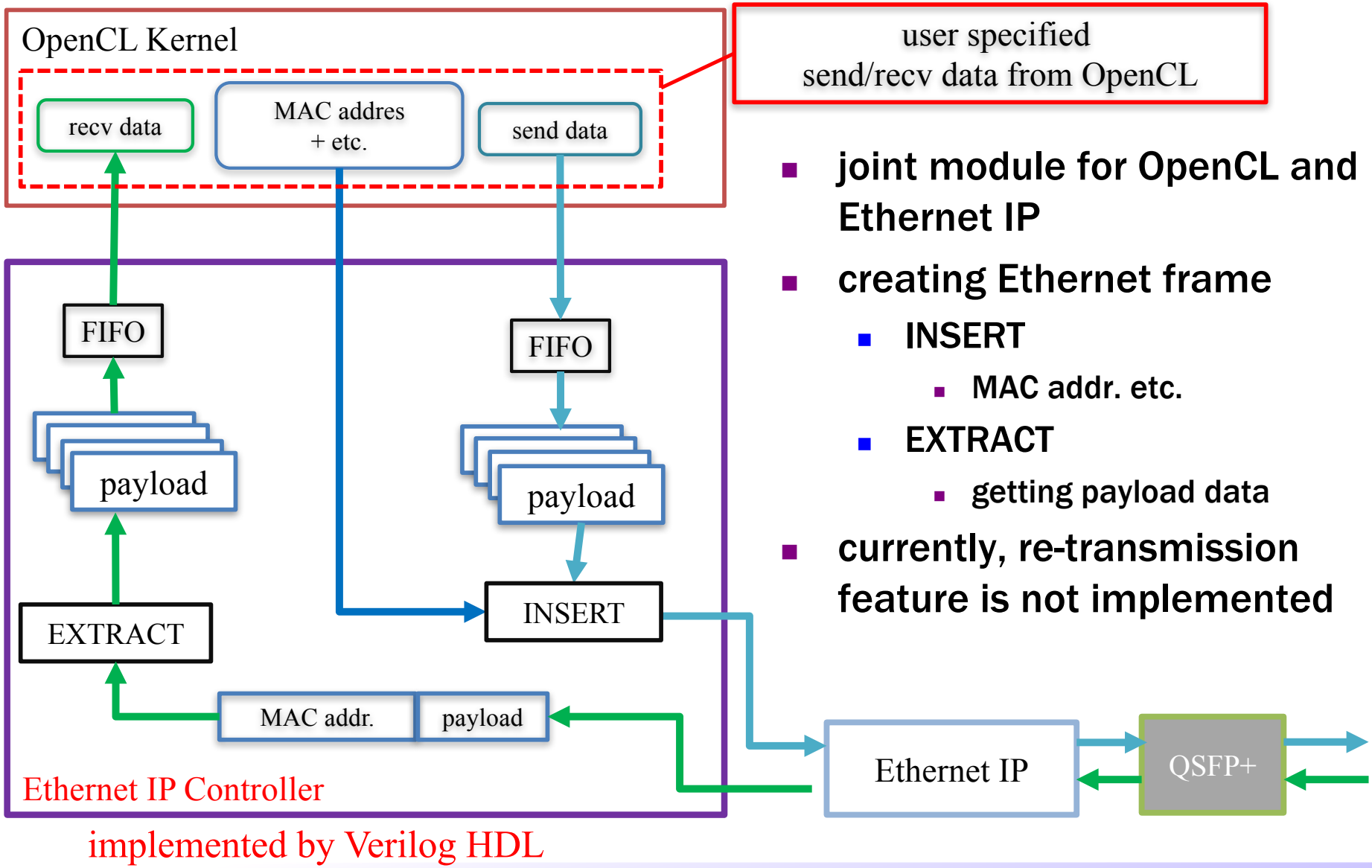
# PPX mini-cluster system




Ethernet will be upgraded to 100Gb x2 on Stratix10

→ We will start just now

# Ethernet IP Controller




- joint module for OpenCL and Ethernet IP
- creating Ethernet frame
  - INSERT
    - MAC addr. etc.
  - EXTRACT
    - getting payload data
- currently, re-transmission feature is not implemented

# OpenCL code example for pingpong

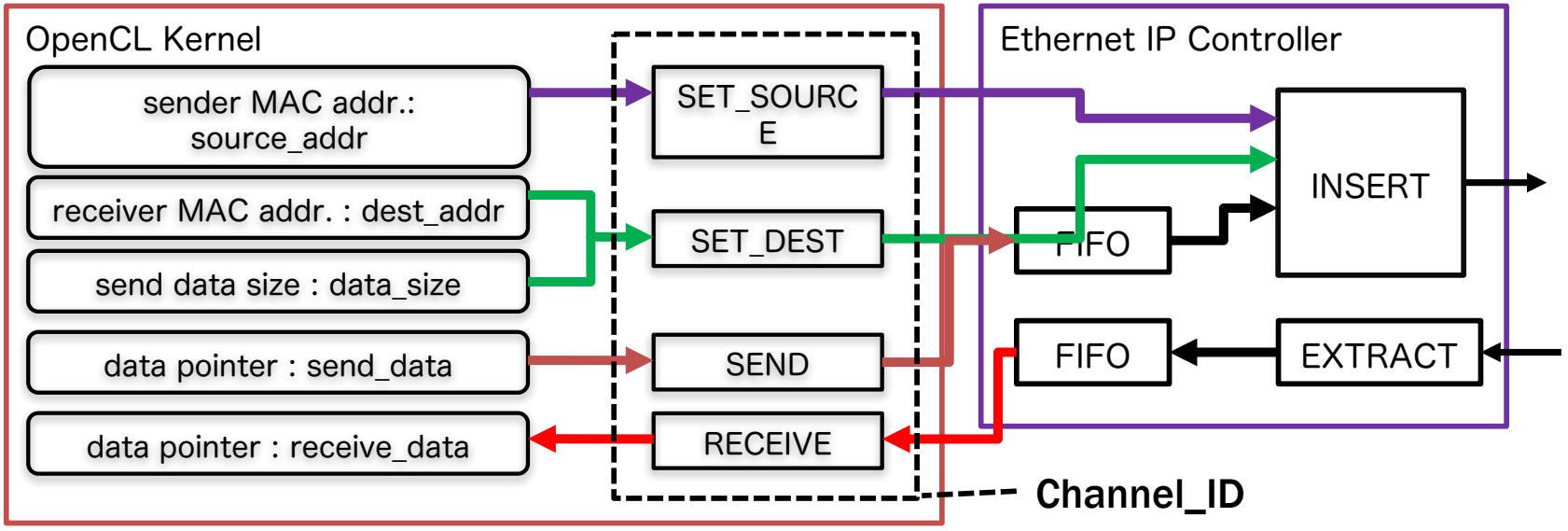
```
write_channel_intel (SET_SOURCE , source_addr);
write_channel_intel (SET_DEST , (int2)(data_size , dest_addr));

for( i = 0 ; i < data_size ; i ++)
      write_channel_intel (SEND , send_data[i] );
```

**sender**

```
for( i = 0 ; i < data_size ; i ++)
      receive_data[i] = read_channel_intel (RECEIVE);
```

**receiver**

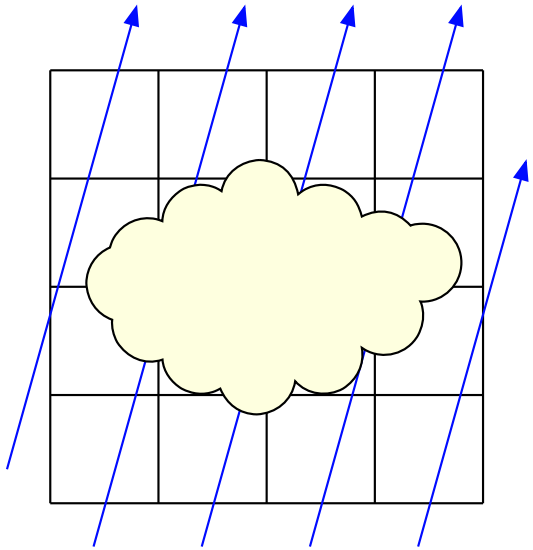


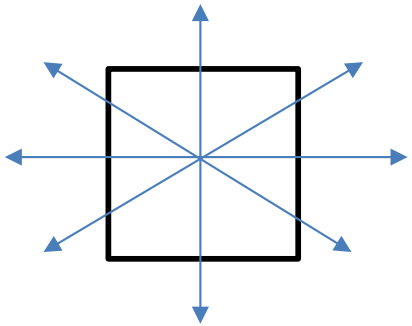
# AiS application example: ARGOT

- ARGOT (Accelerated Radiative transfer on grids using Oct-Tree)
  - Radiative transfer simulation code developed in CCS
  - Two basic computing methods for radiation transfer
    - ARGOT method
      - from a light source
    - ART
      - from spatially spread light sources
- CPU version and GPU version with MPI
- ART method occupies >90% of computation even on GPU, and we need more speedup
  → making FPGA offloading in AiS concept

# ART method

- radiative transfer computing on spatially spread light sources
- ray-tracing on 3-D space with grid decomposed partitions
  - rays are in parallel
  - different input angles
  - no reflection nor refraction (different from 3-D graphics ray-tracing)
  - HEALPix algorithm for ray generation
- large scale for parallel processing
  - mesh size: $100^3$~$1000^3$
  - ray angles: 768~ >1000





# Performance (single FPGA) on ART method

| Device | Perf. [M mesh/sec] | vs CPU |
|---|---|---|
| CPU | 117.49 | - |
| GPU | 105.28 | 0.90 |
| FPGA@228.57MHz (w/o autorun) | 593.11 | 5.05 |
| FPGA@236.11MHz (w/ autorun) | **1714.97** | **14.60** |

- up to 14.6x faster than CPU (GPU doesn't speed up)
- 93% of computation time of ARGOT is dominated by ART method
  → 7.48x speedup on entire code is expected
- It is just on BRAM (in-house memory) and changing the code for DDR utilization

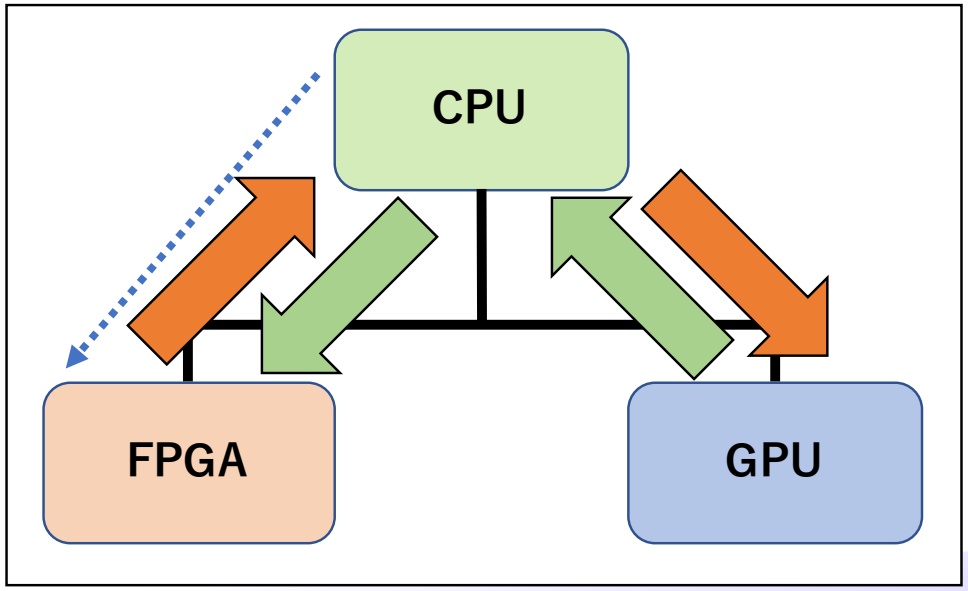
# Circuit resource utilization

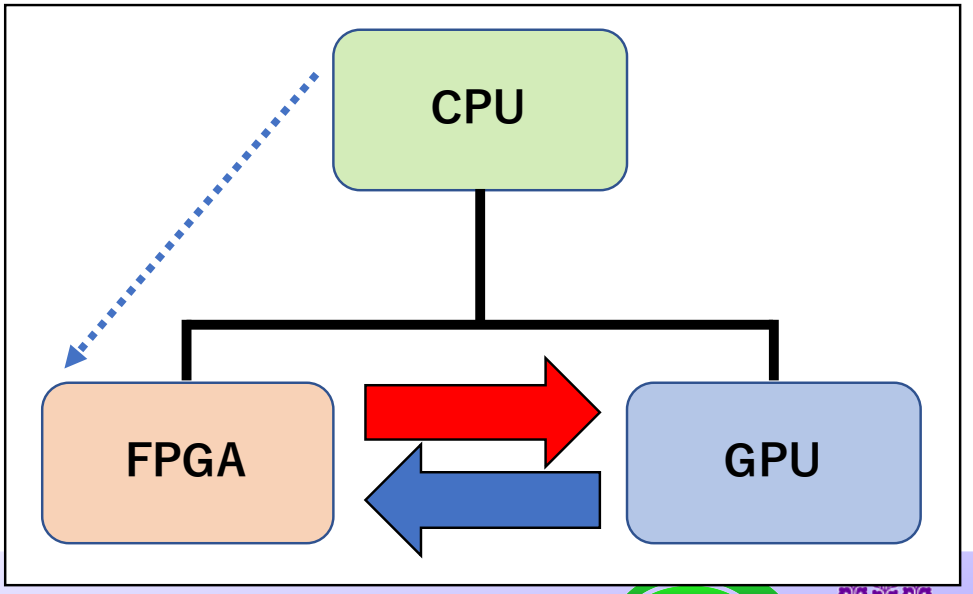
| | ALMs | Registers | M20K | MLAB | MLAB size | DSP | Freq. |
|---|---|---|---|---|---|---|---|
| w/o autorun | 228,610 (54%) | 473,747 (55%) | 1,839 (68%) | 4,330 | 47,968 bits | 536 (35%) | 228.57 MHz |
| w/ autorun | 228,835 (54%) | 467,225 (55%) | 1,716 (63%) | 7,350 | 138,288 bits | 536 (35%) | 236.11 MHz |
| difference | +225 | -6,255 | -123 | +3,020 | +90,320 | 0 | +7.54 |

- largest resource use is on M20K (63%)
  - actually 53.3% (without BSP use)
- DSP utilization is only 53%
- We can achieve up to 2x more speed

# GPU-FPGA communication

- FPGA is only ready for CPU-FPGA data exchange through PCIe
- We developed FPGA-GPU DMA module to be activated by FPGA
  - Currently, still needs CPU help → FPGA standalone



# Performance comparision

## Latency (minimum)

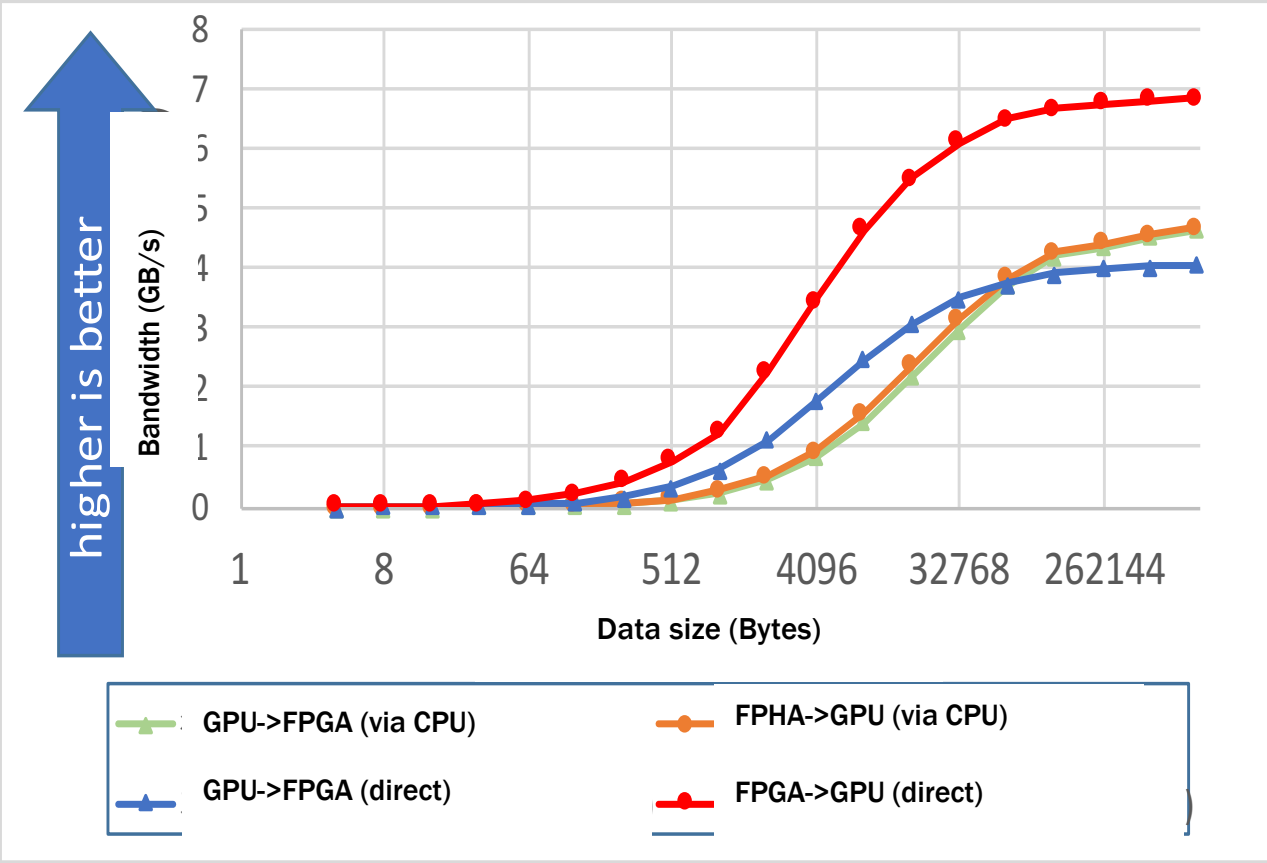
### GPU -> FPGA DMA

| Communication | Min. Latency |
|---|---|
| GPU-CPU-FPGA | 3.92 µsec |
| GPU-FPGA | 1.45 µsec |

### FPGA -> GPU DMA

| Communication | Min. Latency |
|---|---|
| FPGA-CPU-GPU | 3.63 µsec |
| FPGA=GPU | 0.43 µsec |

## Bandwidth



# Next Step

- Precision controlling
  - for ART and ARGOT, SP is too much, HP is not balanced
  - finding best (e, m, s) combination
    e=exponent  m=mantissa  s=(exponent digit shift)
- Combining communication and computation
  - OpenCL computing kernels binding with OpenCL Ethernet communication layer kernels with OpenCL Channel (by Intel SDK)
- Combining GPU and FPGA
  - Basic system is done, but not yet combined for full-control by FPGA
- Final work: How to Program ???
  - FPGA-parallel programming framework with high level parallel language XcalableACC
  - Will be started as a new theme for DOE-MEXT collaboration with FTG of ORNL (Jeff Vetter) to welcome OpenARC compiler for OpenACC->FPGA

# PACS-X

- Procurement started from last week toward complete installation on March 2019
- System consists of two parts
  - Base System: only with CPU+GPU (2 GPU devices/node) and ordinary InfiniBand or similar (200Gbps?)
  - FPGA-ready System: CPU+GPU+FPGA, additionally with 2-4 channels of 100Gbps Ethernet
    - By switch ?
    - By directly connected 2D-Torus ?
  - Node configuration balance is still under designing
- Development of GPU based applications and GPU+FPGA applications is under going

# Summary

- OFP is on stable operation phase to support many CSE research with nation wide researchers under collaboration with U. Tokyo
- FPGA for HPC is very attractive theme for next generation of accelerated platform with combined computation and communication
- 360-degree system to cover highly parallel SPMT computing by GPU and flexible processing on FPGA with communication feature
- OpenCL-enabled programming including communication for application users
- CCS, U. Tsukuba is moving forward to realize AiS concept on next generation multi-hetero supercomputing through PACS-X Project