COMA (PACS-IX) Project

Taisuke Boku

Deputy Directory, HPC Division

Center for Computational Sciences

University of Tsukuba

Two Streams of Supercomputers at CCS

- Service oriented general purpose machine with regular budget
  → T2K-Tsukuba & follow-up
  - Supercomputer rental budget to support 4-5 years period as national shared supercomputer resource (including HPCI)
  - High performance, commodity base and easy to program
  - Large scale general purpose parallel processing
  - Latest system: T2K-Tsukuba

- Research & mission oriented project machine with special budget
  → PAX/PACS series
  - Supercomputer development for specific application area
  - Peak performance centric
  - "Highly skilled" high performance system for high-end computing
  - Latest system: HA-PACS

Hisotry of PAX (PACS) MPP series

- Launched in 1977 (Prof. Hoshino and Prof. Kawai)
- First machine was completed in 1979
- 6th generation machine CP-PACS was ranked #1 in TOP500 in Nov. 1996

1978
1st PACS-9

1980
2nd PAXS-32

1989
5th QCDPAX

1996
6th CP-PACS as world fastest machine

2006
7th PACS-CS bandwidth aware

2012
8th HA-PACS GPU accelerated

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	HA-PACS	802 TFLOPS

- High end supercomputer based on MPP architecture towards "practical machine" under collaboration with computational scientists and computer scientists
- Development in Application-driven
- Continuation of R & D by an organization

PACS (PAX) Series

- MPP system R&D continued at U. Tsukuba for more than 30 years
- Naming history
  - (original) PACS : Processor Array for Continuum Simulation
  - PAX : Parallel Array eXperiment
  - (recent) PACS : Parallel Advanced system for Computational Sciences
- Coupling of need from applications and seeds from the latest HPC technology, the machines have been developed and operated with the effort by application users on programming
  → a sort of application oriented machine
  (not for a single application)
- HA-PACS is the first system in the series to introduce accelerating devices (GPUs)
- **CCS has been focusing on the accelerating devices for ultra high performance to provide to "high-end" users who require extreme computing facilities**

Next PACS with another accelerating devices

- HA-PACS : large scale highly dense GPU cluster
- Another accelerating device today – many-core arch.
- Intel Xeon Phi (MIC: Many Integrated Core)
  - a number of simple cores
  - currently delivered as an accelerating device attached to CPU through PCIe bus (similar to GPU)
  - each core is available to run ordinary Linux on x86 ISA
- CPU core of Xeon Phi (KNC: Knights Corner generation)
  - similar to Pentium4 class x86, approx. 1GHz of frequency
    ⇨ each core is relatively weak, but 512bit AVX SIMD instruction enhances floating point performance (FLOPS)
    ⇨ "throughput core" (vs "latency core")
  - 60 (or 61) cores + GDR5 memory to provide wide-bit high bandwidth memory access


Toward JCAHPC system procurement

- Joint system procurement and operation with U. Tokyo on 2015 at Kashiwa Campus of U. Tokyo
- Currently targeting Many-Core processor for CPU
  - Xeon Phi will be available as “main CPU”, not as “accelerator board”
  - Large performance improvement on each core, increasing # of cores, higher frequency, ...
  - We need much of experience to utilize throughput-core system for wide variety of HPC applications
  - Code tuning, new algorithm, new library, ...
  - Operating system for throughput-core (U. Tokyo)

- We need a test-bed for these purposes

What is COMA ?

- Cluster Of Many-core Architecture processor
- COMA
  - a famous “cluster of galaxies”
  - galaxy = cluster of stars (= many core)
  - cluster of galaxies = cluster of many-cores
- We will continue the name of PACS as a simple name of series of machines with “index code” (not unique name)
  ⇨ COMA is PACS-IX

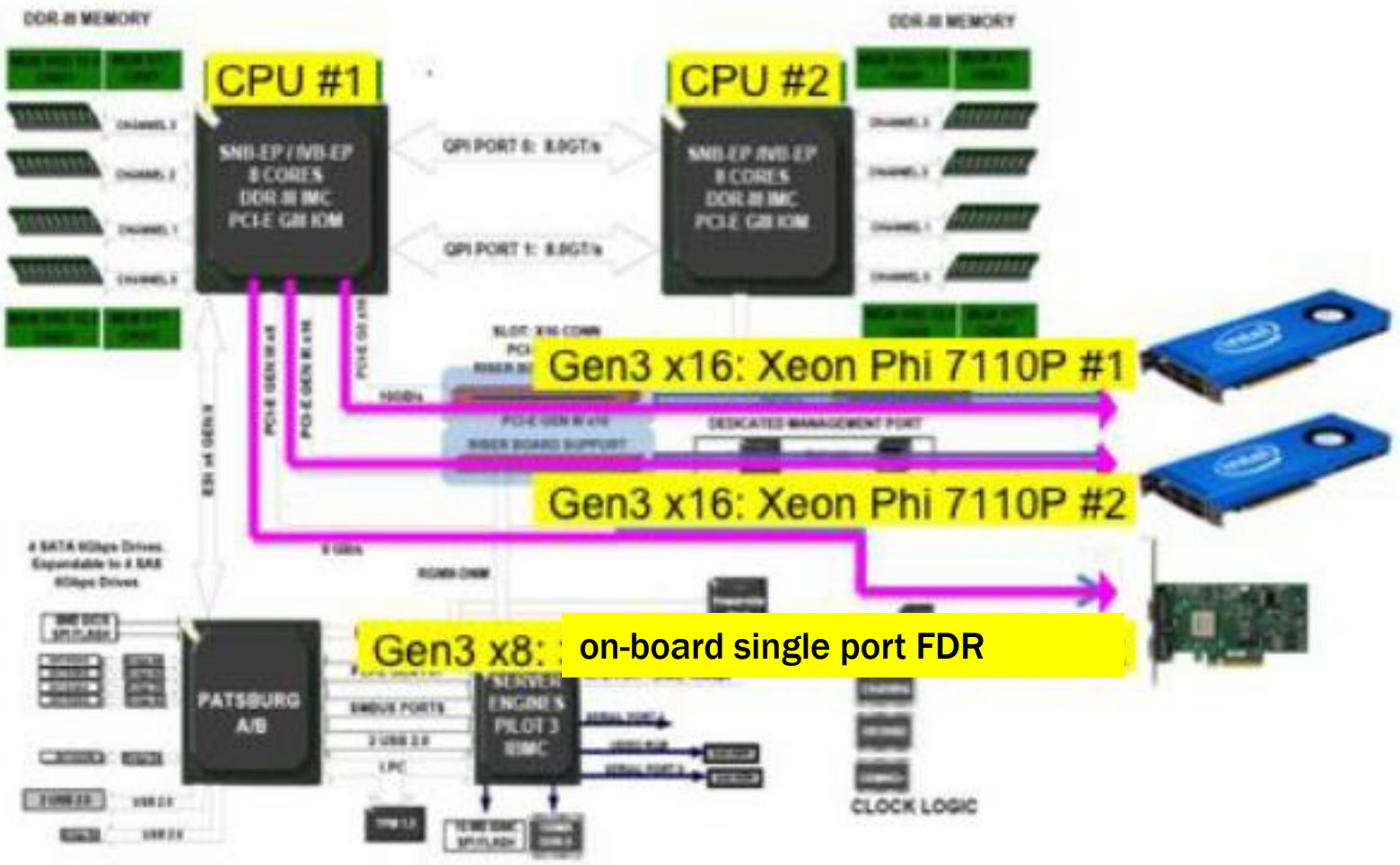

Basic specification of COMA

- 9th generation of PACS (PACS-IX)
  (operated with HA-PACS in parallel)
- operation starts after shutdown of T2K-Tsukuba
  - deployment at end of March 2014
  - operation starts at 15th April
- System configuration
  - computation node : general CPU + MIC
  - node construction
    - CPU x 2: Intel Xeon E5-2670v2
    - MIC x 2: Intel Xeon Phi 7110P
    - Memory: CPU=64GB MIC=16GB (8GB x 2)
    - Network: IB FDR Full-bisection b/w Fat Tree
  - # of nodes: 383+10=393
  - peak performance: CPU=157.2 TFlops MIC=843.8 TFlops
    TOTAL: 1001 TFlops = **1.001 PFLOPS**
- **System delivery: Cray Inc.**

Computation node

- CPU (x2): Intel Xeon E5-2670v2 (Ivy Bridge)
  - 10 core/CPU, 2.5GHz
  - 200GFLOPS x 2 = 400GFLOPS
  - memory: 64GB
    DDR3 1866MHz x 4chan x 2CPU = 119.4 GB/s
- MIC (x2): Intel Xeon Phi 7110P
  - 61 core/MIC, 1.1GHz
  - 1.0736 TFLOPS x 2 = 2.1472 TFLOPS
  - memory: 8GB x 2 = 16GB
    GDR5 352GB/s x 2 = 704 GB/s
- Interconnection: InfiniBand FDR (on-board)
  - Mellanox Connect-X3
- Local HDD: 1TB x 2 (RAID-1)

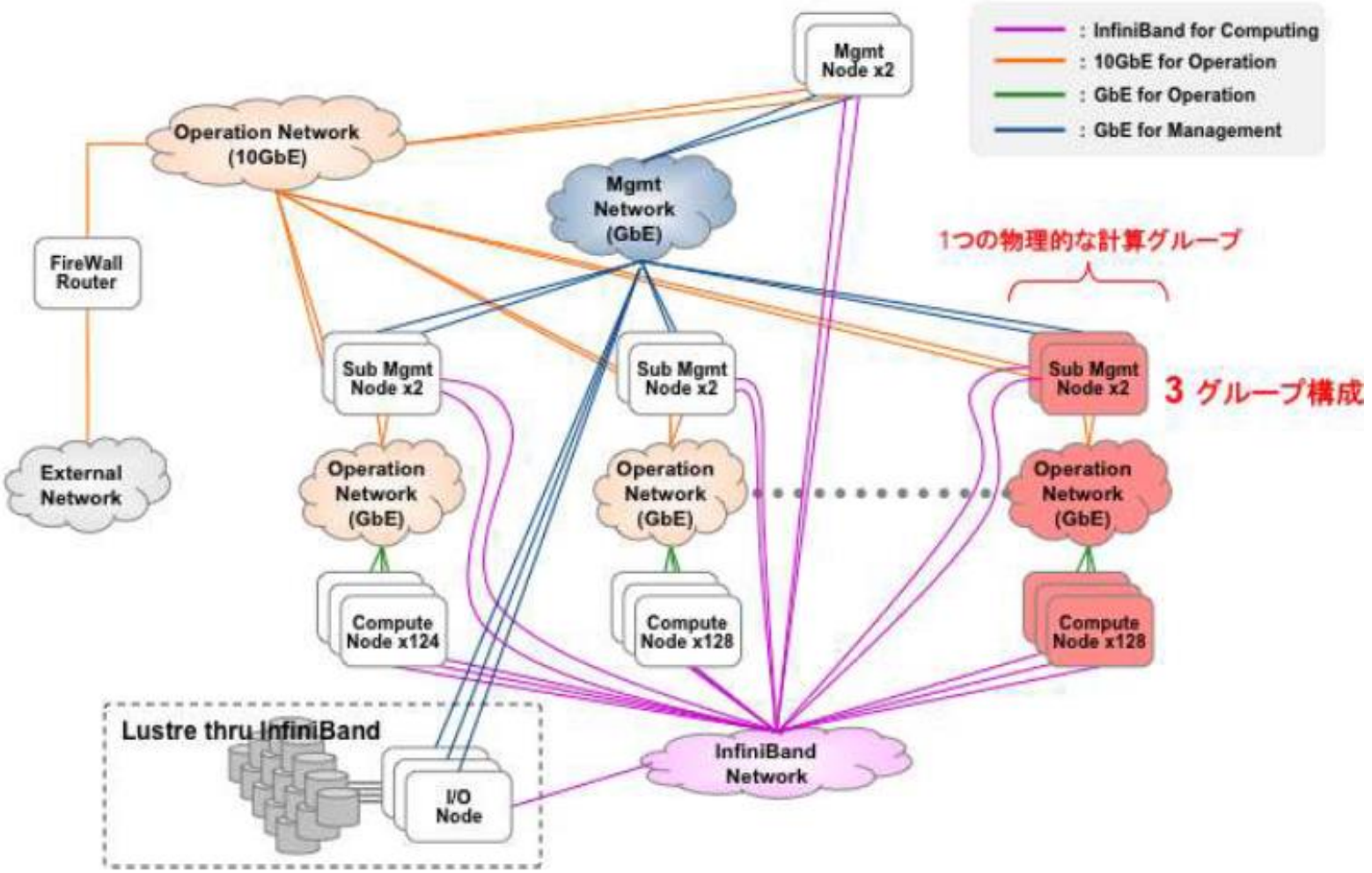
Block diagram of node



System performance summary

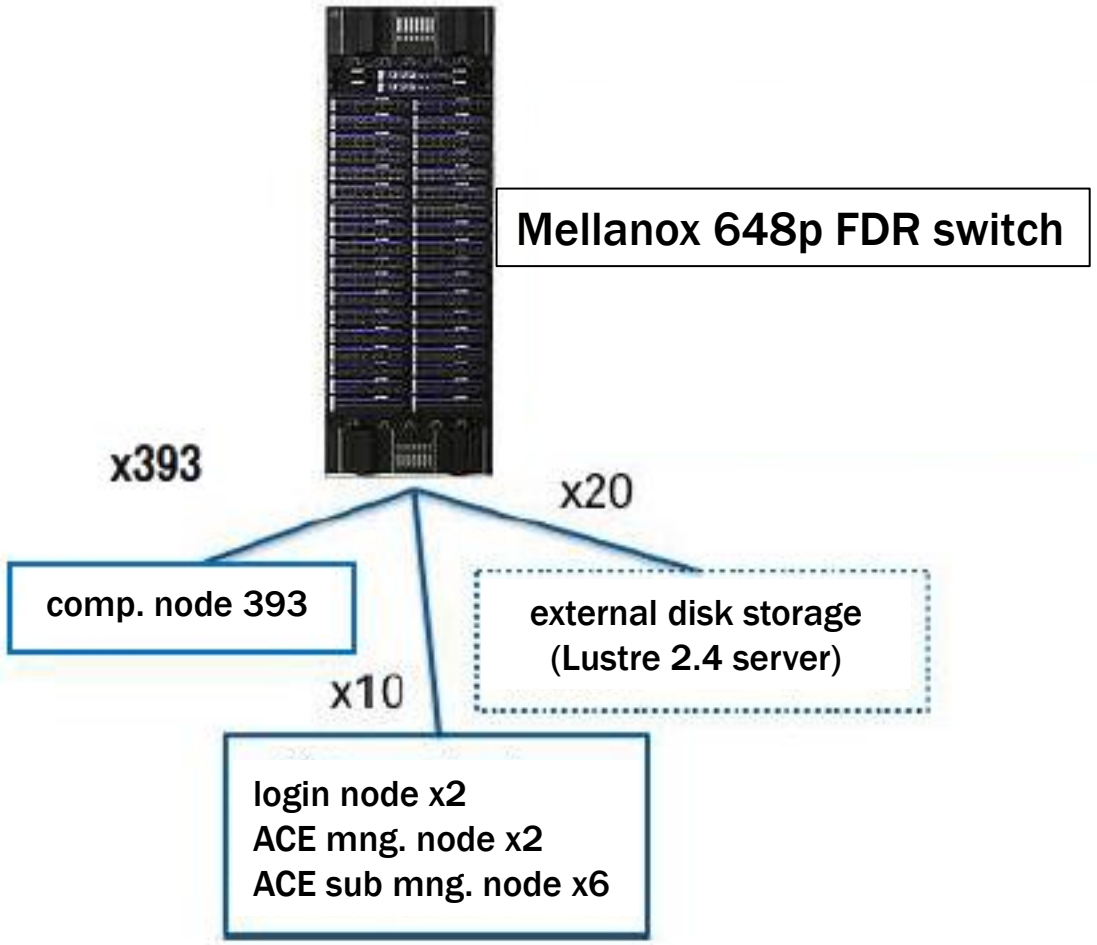
- # of nodes : 393
  - peak performance
    - CPU: 157.2 TFLOPS
    - MIC: 843.8 TFLOPS
    - TOTAL: 1001 TFLOPS = 1.001 PFLOPS
  - memory capacity
    - CPU: 25.1 TB
    - MIC: 6.3 TB
- Interconnection: Fat-Tree full-bisection b/w
  - Bisection bandwidth: 2.75 TB/s


Network construction

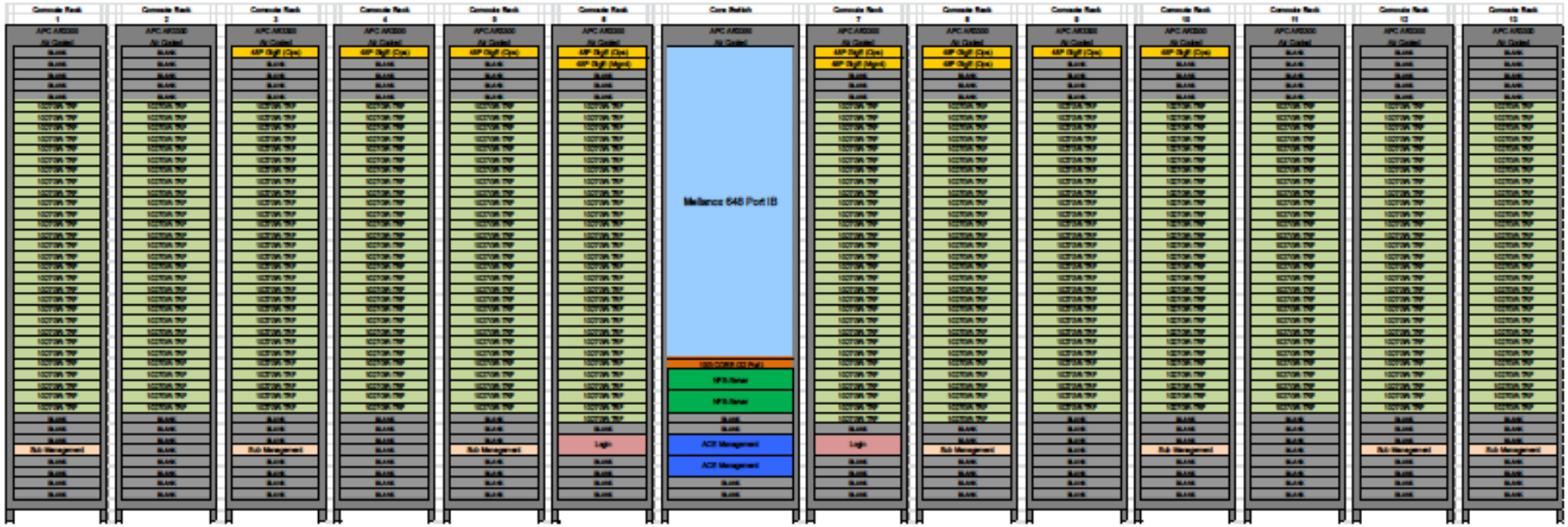


Interconnection Network for Computtion

- Mellanox 648p FDR switch
  - 2.75TB/s bisection b/w
- Brocade ICX6650-32
  - 32port 10GigE SFP+

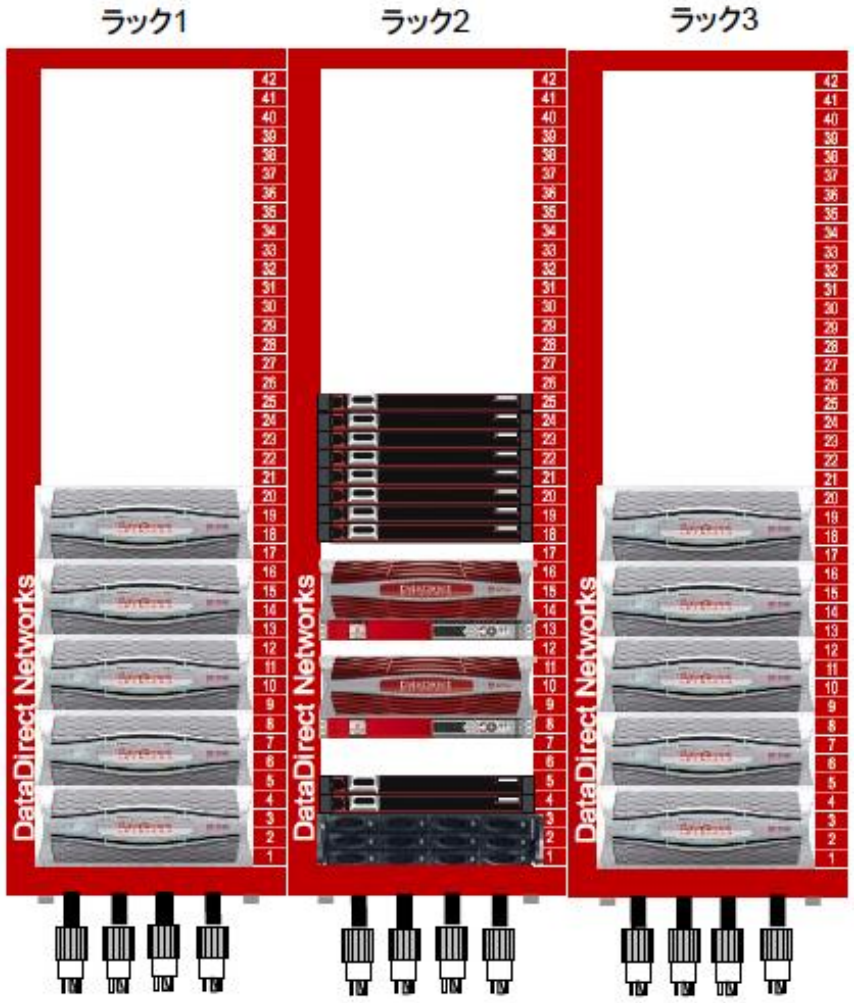


System Image



Storage

- Shared file server
  - RAID6 + Luster
  - OST: DDN SS8460 x 10
    - 4TB x 550基
  - OSS x 8
  - Controller: DDN SFA12K-40
  - InfiniBand FDR x8 (OSS)
  - User space: 1.5PB
  - Flat access from all computation nodes
- NFS server
  - 12TB for /home, etc.



Software (OS, programming)

- OS: Red Hat Enterprise Linux (login server)
- OS: CentOS (compute node)
- Cluster management: ACE
- Job scheduler: SLURM
- Programming environment:
  - Intel Cluster Studio XE2013 (Composer/XE)
  - Fortran95/C/C++
  - Intel MPI


Programmin on MIC

- Linux is permanently running on each MIC (KNC)
- Two programming model
  - Native Mode: Direct execution of code on each MIC
    - Multi-threaded code (OpenMP) can be executed
    - Max 240 threads with hardware thread control
    - No I/O (HDD) – HDD on host CPU is mounted through NFS
    - MPI is possible for MIC-to-MIC communication through host InfiniBand
  - Offload Mode: “offloaded” part of host CPU code runs on MIC
    - Composer XE (Intel extended compiler for MIC)
    - Code parts for “device” (MIC) is explicitly described
    - Writing OpenMP in offloaded part, it can be executed in thread parallel on many-cores


Example of offload programming

```
#include <stdio.h>
#include <omp.h>
#define SIZE 1000
int main()
{
  int inarray[SIZE], sum, validsum;
  int i;
  int nth;

  validsum=sum=0;
  for(i=0; i<SIZE; i++){
    inarray[i]=i;
    validsum+=i;
  }
  nth=0; // for checking
```

```
#pragma offload target(mic:0) in(inarray:
      alloc_if(1) free_if(0)) out(sum) out(nth)
  {
    int lsum = 0;
    int i;
#pragma omp parallel for default(none) ¥
          shared(inarray) reduction (+:lsum)
    for(i = 0; i < SIZE; i++)
      lsum += inarray[i];
    sum = lsum;
#pragma omp parallel
#pragma omp master
    nth = omp_get_num_threads();
  }
  printf("sum = %d   validsum = %d¥n", sum, validsum);
  printf("num thread = %d¥n", nth);
}
```

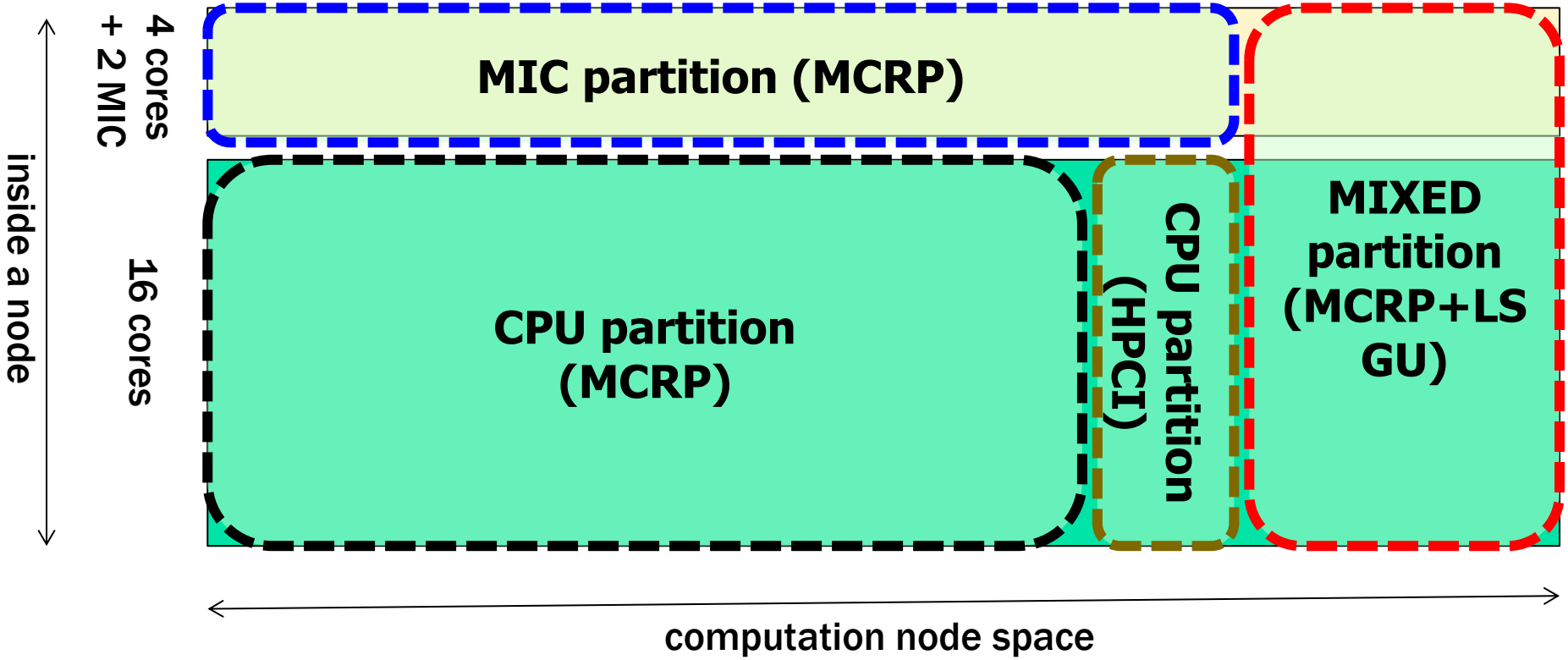
Job control on MIC

- Three types of partitions (job queue)
  - CPU partition
    - using 16 cores out of 20 on general CPUs on each node
    - ordinary programming for multi-core (OpenMP + MPI)
  - MIC partition
    - dedicating 4 cores out of 20 on general CPUs to control two MICs
    - (maybe) running with offload model
  - Mixed partition
    - all CPU and MIC resources is available for the job
    - hybrid program, work sharing, MIC native mode execution


Operation program of COMA

- Multi-disciplinary Collaborative Research Program
  - supporting advanced computational science/engineering
  - free of charge (application + ext. review)
  - CPU, MIC, Mixed
- HPCI
  - networking nation-wide all supercomputers under MEXT by single-sign-on system
  - free of charge (application + HPCI review committee)
  - CPU
- Large Scale General Use
  - assigning job to node-to-node manner
  - charge CPU utilization cost
  - Mixed
- Official operation starts on 15th of April, 2014

System partitioning for job queue



User support and education (planned)

- Tutorial for many-core system utilization
  - by support of Intel and Cray
  - HPC Seminar (Summer + Winter) by CCS
  - International HPC School
- Programming support
  - currently, the policy is the same as HA-PACS
    ⇨ we basically hope users & projects to write and tune their code with their own effort
    ⇨ MCRP strongly recommends to make a team of computational scientists and computer scientists


Summary

- COMA (PACS-IX) is a new cluster in CCS deployed in the end of March 2014, based on Intel Xeon Phi (MIC) accelerating devices
- 393 nodes, 786 MICs (Intel Xeon Phi 7110P) for 1 PFLOPS of peak performance
- With advanced many-core architecture, accelerated computing with high-density, high-performance and low-power is available
- Three types of operation model: CPU, MIC, mixed
- Program: CCS Multidisciplinary Collaborative Research, general use (not free)
- Operation starts at 15th April, 2014