Division of
High Performance Computing Systems
(High Performance Computing Systems Group)

Taisuke Boku

Division Leader, HPC Division

Center for Computational Sciences

University of Tsukuba

HPC Division Members

- Faculty Staffs
  - Taisuke Boku (Leader, Professor)
  - Mitsuhisa Sato (Professor)
  - Yuetsu Kodama (Professor)
  - Daisuke Takahashi (Professor)
  - Osamu Tatebe (Associate Professor)
  - Hideyuki Kawashima (Lecturer) *(moved from Computational Informatics Div.)
  - Hiroto Tadano (Assistant Professor)
  - Yutaka Ishikawa (Visiting Professor, University of Tokyo)
  - (Toshihiro Hanawa, Associate Professor, ~2013/12)
- Researchers
  - Research Fellows: Masahiro Tanaka, Hiroaki Umeda, Hideo Nuga, Mohamed Amin Jabri, Kazuya Matsumoto
  - Collaborative Fellows: Moritoshi Yasunaga, Koichi Wada, Tetsuya Sakurai, Yoshiki Yamaguchi

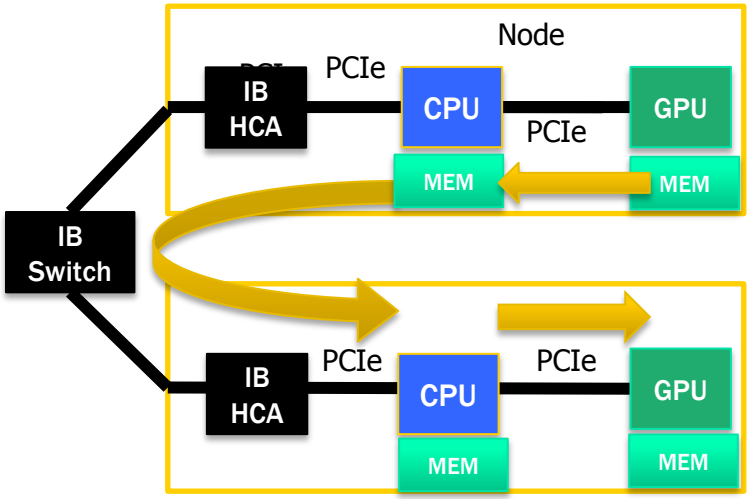
Major Research Activities

- HPC System Hardware
  - TCA/PEACH2: high performance direct communication between accelerators
- HPC System Software
  - XcalableMP: PGAS language for high level parallel programming on distributed memory system
  - XcalableMP-dev: accelerating device extension of XcalbaleMP
  - Gfarm: wide area distributed file system
  - Energy Effective System Design
- High Performance Numerical Algorithms/Libraries
  - FFT
  - Block Krylov subspace methods
- Application Code Development
  - RS-DFT, Nuclear Fusion, Large Eddy Simulation

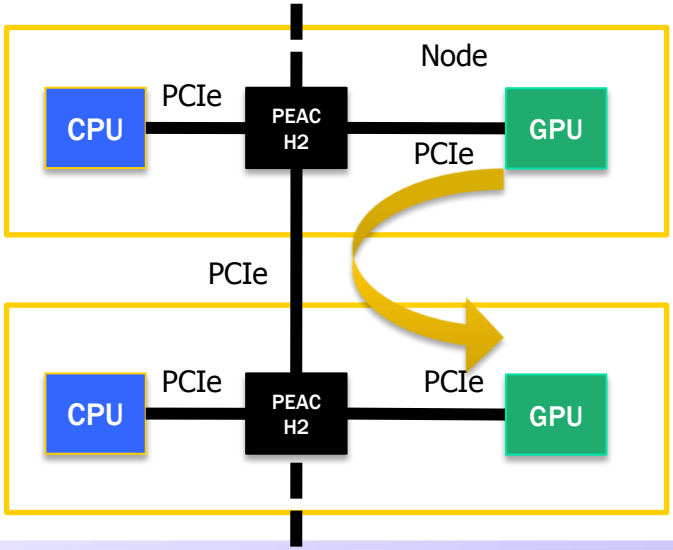
TCA
Tightly Coupled Accelerators
(main contribution by T. Boku, Y. Kodama, T. Hanawa)


TCA (Tightly Coupled Accelerators) Architecture

- **True GPU-direct**
  - current GPU clusters require 3-hop communication (3-5 times memory copy)
  - For strong scaling, inter-GPU direct communication protocol is needed for lower latency and higher throughput



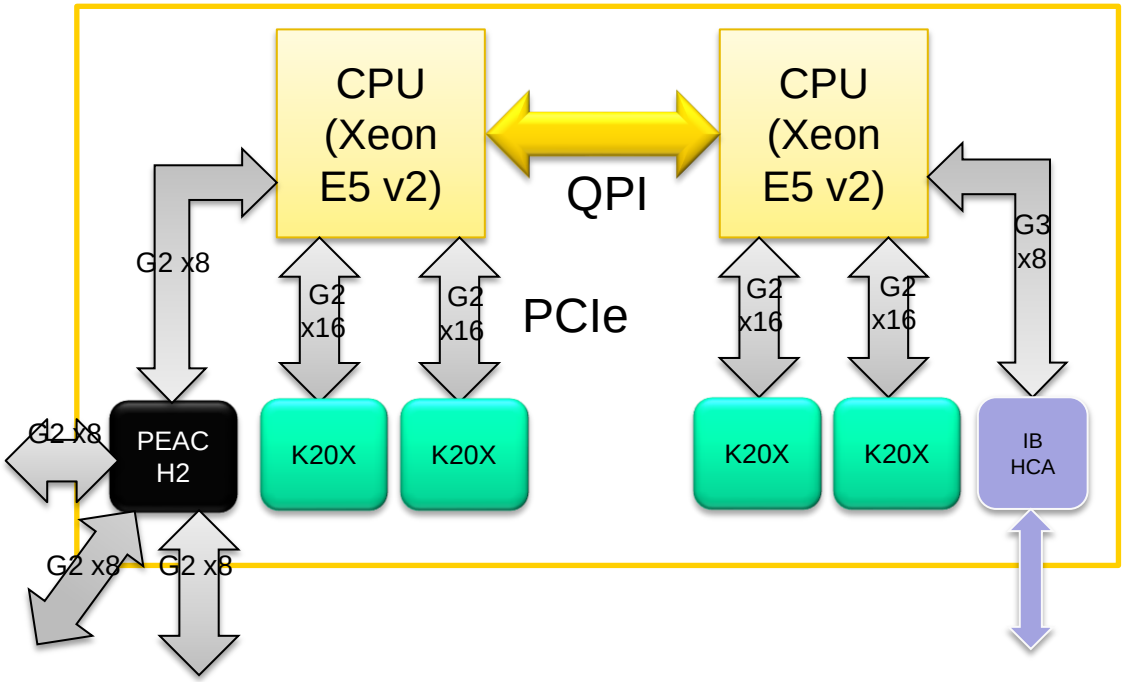
- **Enhanced version of PEACH**
  ⇒ ***PEACH2***
  - x4 lanes -> x8 lanes
  - hardwired on main data path and PCIe interface fabric



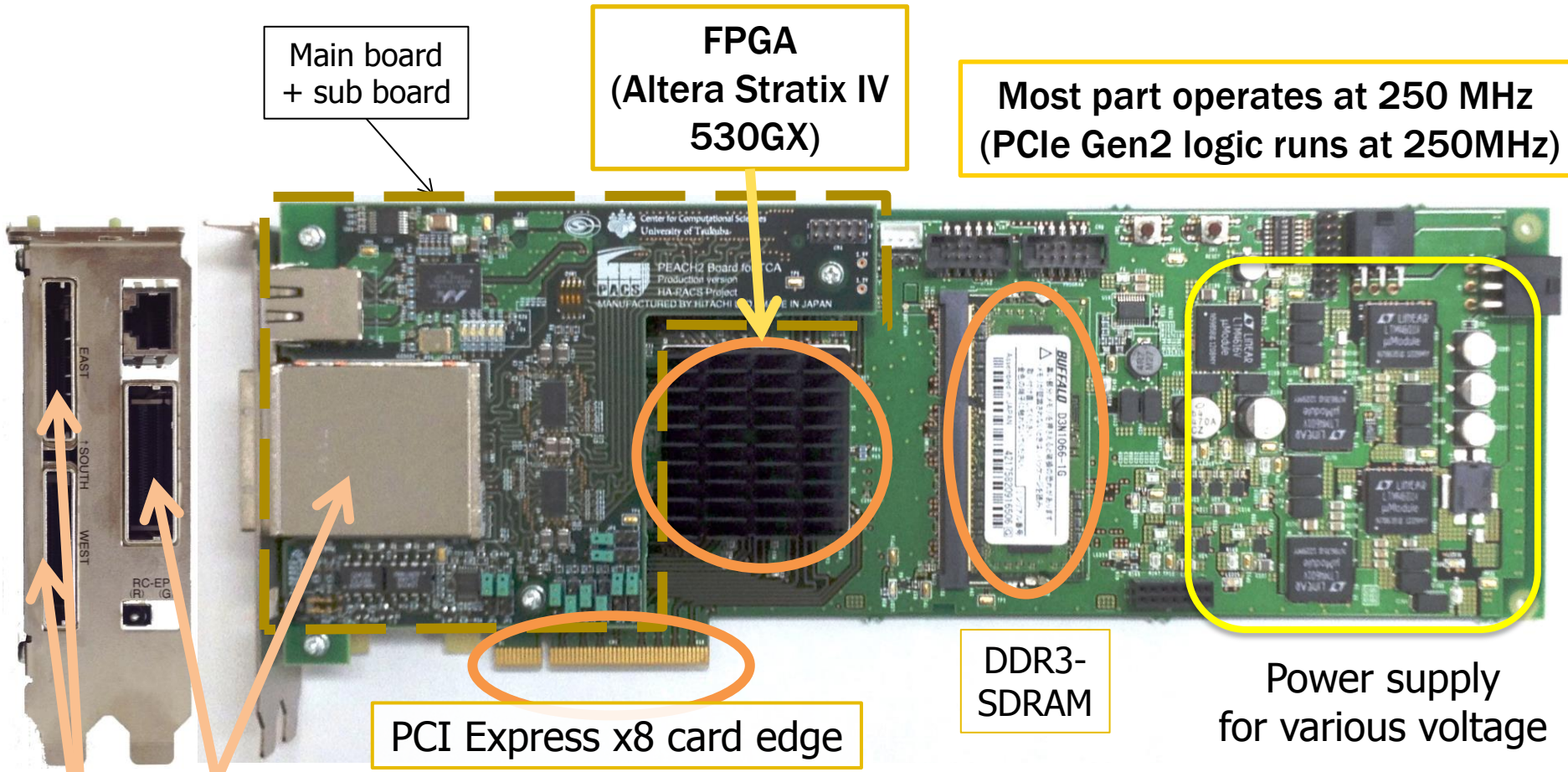
TCA testbed node structure

- CPU can uniformly access to GPUs.
- PEACH2 can access every GPUs
  - Kepler architecture + CUDA 5.0 “GPUDirect Support for RDMA”
  - Performance over QPI is quite bad.
    => support only for two GPUs on the same socket
- Connect among 3 nodes

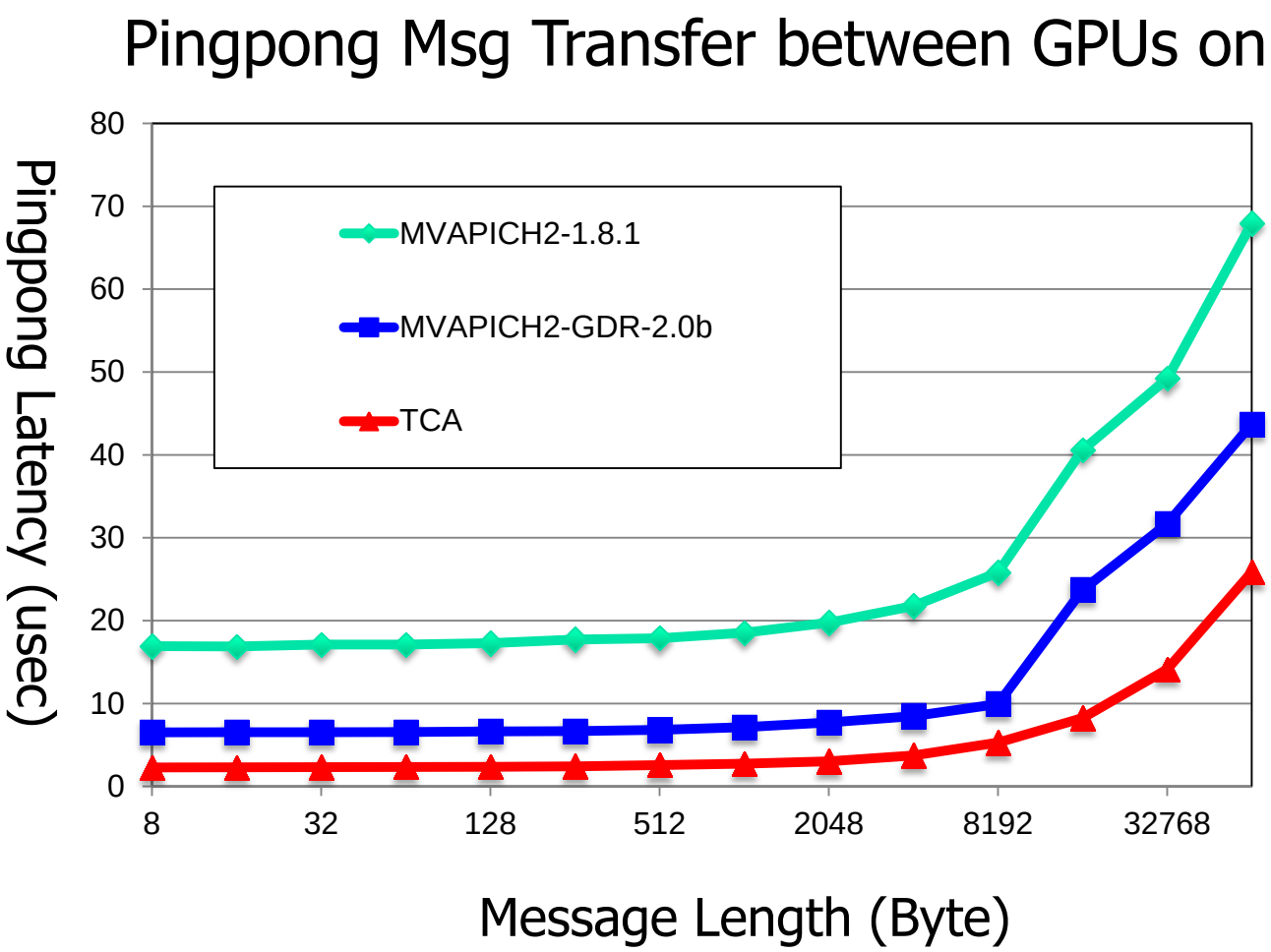
- This configuration is similar to HA-PACS base cluster except PEACH2.
  - All the PCIe lanes (80 lanes) embedded in CPUs are used.



PEACH2 board



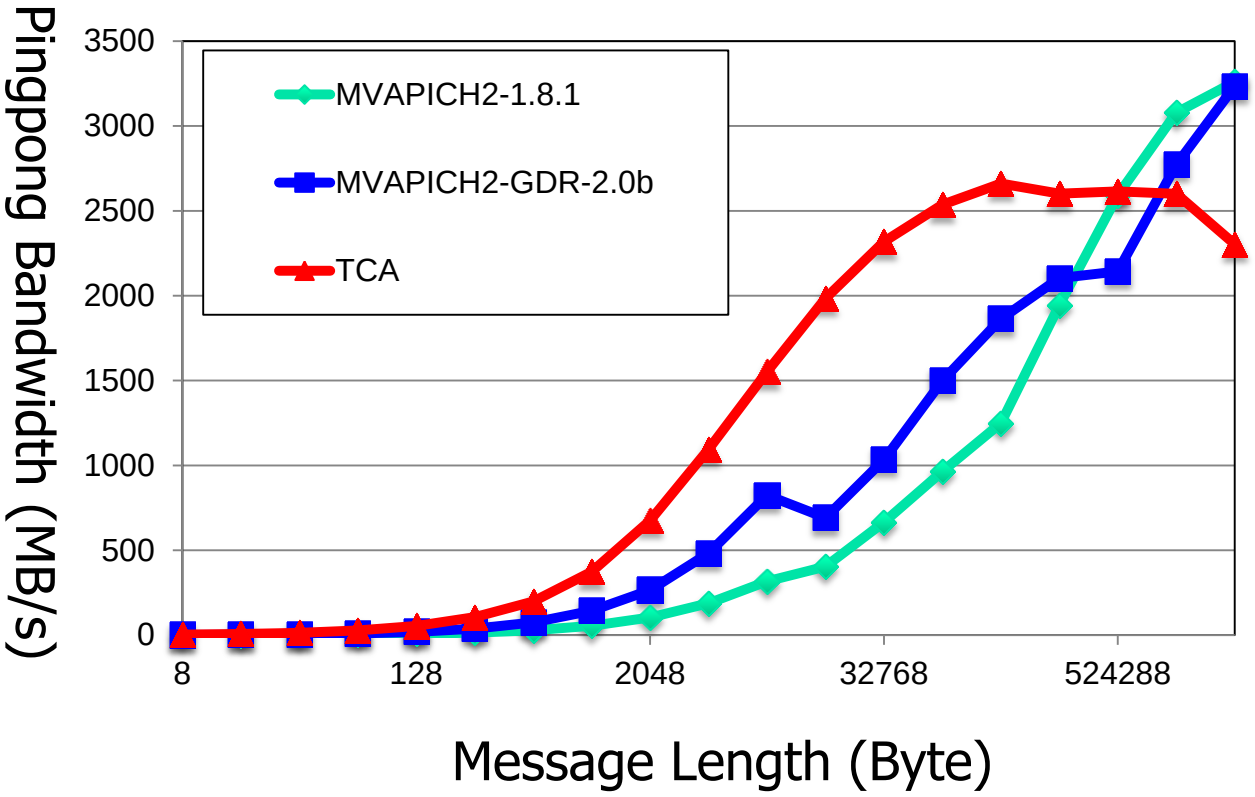
PEACH2 Performance (Lantecy)



Short Msg Latency
MVAPICH2: 17 usec
MVAPICH2/GDR: 6.5 usec
TCA: 2.3 usec

PEACH2 Performance (Bandwidth)

Pingpong Msg Transfer between GPUs on different nodes



PCIe gen2 x8 connection
= 3.8GB/s theoretical peak
max. b/w of TCA is 2.6GB/s
crossover = 512KB

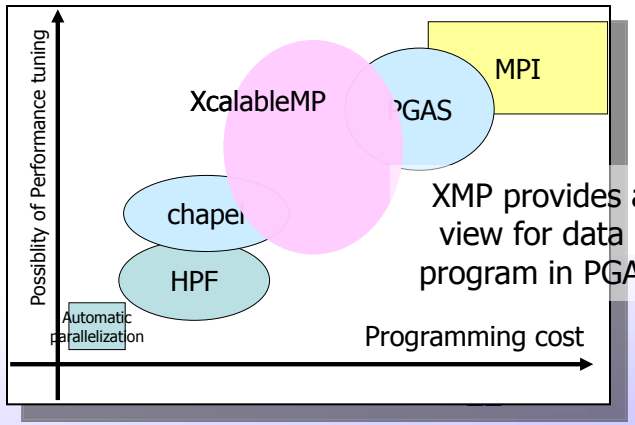
XcalbaleMP (XMP)
PGAS language for high level
parallel programming
(main contribution by M. Sato)

XMP-dev
accelerating device extension of XMP
(main contribution by M. Sato, T. Boku)


XcalableMP(XMP) http://www.xcalablemp.org

- What's XcalableMP (XMP for short)?
  - A PGAS programming model and language for distributed memory , proposed by **XMP Spec WG**
  - XMP Spec WG is a special interest group to design and draft the specification of XcalableMP language. It is now organized under **PC Cluster Consortium**, Japan. Mainly active in Japan, but open for everybody.

- Project status (as of Nov. 2013)
  - XMP Spec **Version 1.2** is available at XMP site. new features: mixed OpenMP and OpenACC , libraries for collective communications.
  - Reference implementation by U. Tsukuba and Riken AICS: **Version 0.7 (C and Fortran90)** is available for PC clusters, Cray XT and K computer. Source-to-Source compiler to code with the runtime on top of MPI and GasNet.



- Language Features
  - Directive-based language extensions for Fortran and C for PGAS model
  - Global view programming with global-view distributed data structures for data parallelism
    - SPMD execution model as MPI
    - pragmas for data distribution of global array.
    - Work mapping constructs to map works and iteration with affinity to data explicitly.
    - Rich communication and sync directives such as "gmove" and "shadow".
    - Many concepts are inherited from HPF
  - Co-array feature of CAF is adopted as a part of the language spec for local view programming (also defined in C).

Code example

```
int array[YMAX][XMAX];

#pragma xmp nodes p(4)
#pragma xmp template t(YMAX)
#pragma xmp distribute t(block) on p
#pragma xmp align array[i][*] to t(i)

main(){
  int i, j, res;
  res = 0;

#pragma xmp loop on t(i)  reduction(+:res)
  for(i = 0; i < 10; i++)
    for(j = 0; j < 10; j++){
      array[i][j] = func(i, j);
      res += array[i][j];
    }
}
```

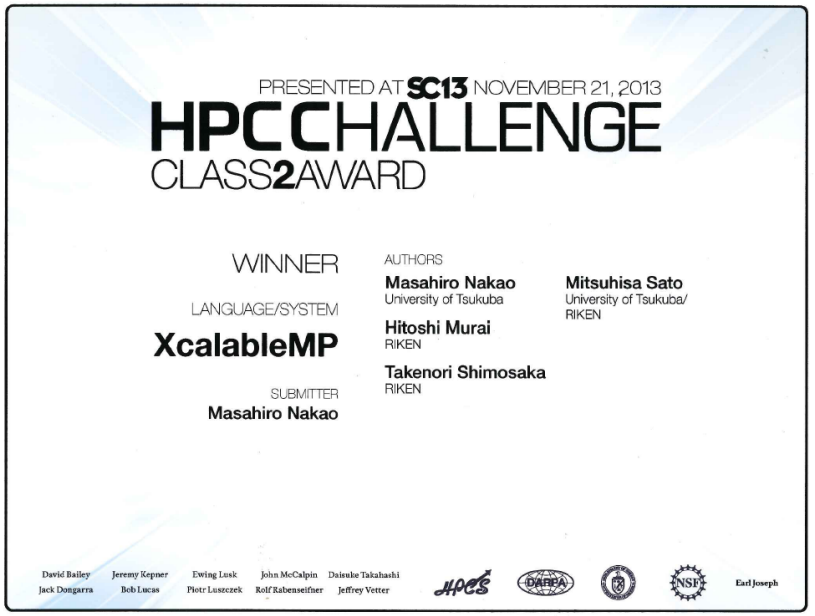
data distribution

add to the serial code : incremental parallelization

work sharing and data synchronization

HPCC Class2 competition

- XMP (U. Tsukuba and RIKEN AICS team) won HPCC Class2 Awards in SC13.
  - Implementation and performance evaluation on the K computer
- HPC Challenge Benchmarks:
  - HPL, FFT, RandomAccess, Stream
  - Class 1 for Performance
  - Class 2 for Productivity of Prog. Language (Performance and Elegance)

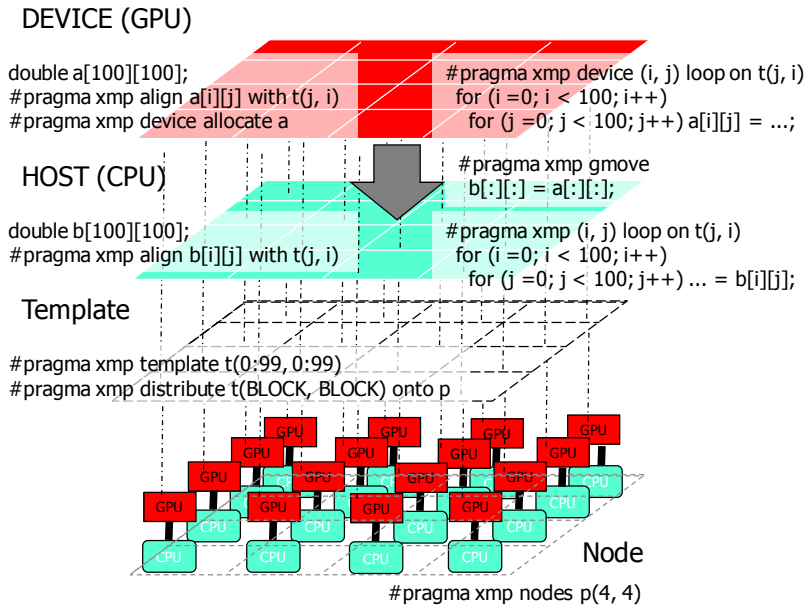


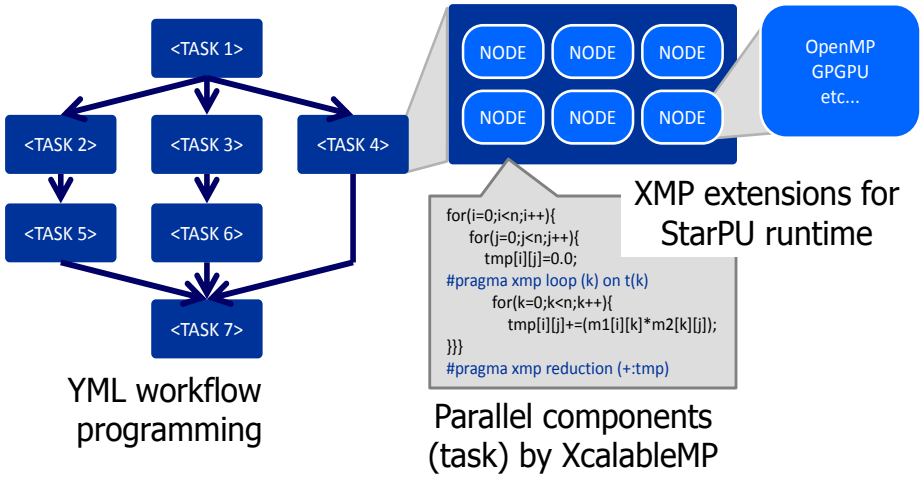
Benchmark	# Nodes	Performance	SLOC
HPL	16,384	933.8 TFlops (44.5% of peak)	306
RandomAccess	16,384	162.6 GUPs	250
FFT	36,864	50.1 TFlops (1.1% of peak)	239 + 283 + 1892
STREAM	16,384	481.8 TB/s	66
HIMENO	**82,944**	1.3 PFlops (12.7% of peak)	137

Full compute nodes

Researches and projects related to XMP

- XMP extension for GPU: XMP-dev and XMP/OpenACC integration (JST CREST)
- FP2C (Framework for Post-Petascale Computing) : the multilevel programming as a solution for post-petascale system (FP3C Japan-French project)
  - Integration with YML flow lang.
  - XMP/StarPU integration scheduling CPU/GPU
- Porting to other platform
  - NEC SX, IBM BG/Q
- Optimization for Intel Xeon Phi
- Dynamic Tasking with XMP





Gfarm
Wide Area Large Scale Distributed File System
(main contribution by O. Tatebe)


Gfarm file system

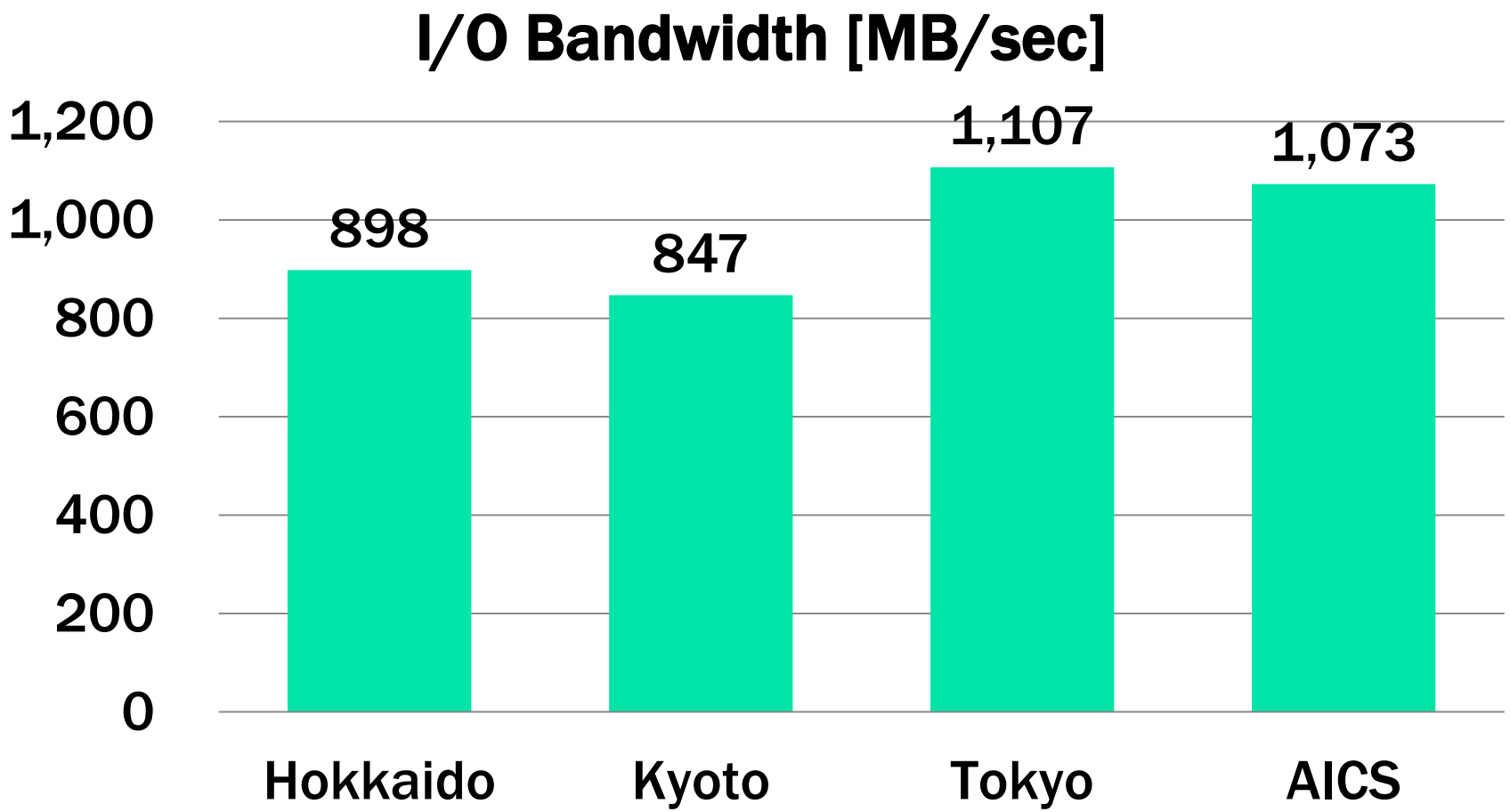
- Award-winning file system since 2000
  - Distributed infrastructure award in SC03
  - Most Innovative Use of Storage In Support of Science Award in SC05
  - Winner – Large Systems in HPC Storage Challenge in SC06
- Open Source distributed file system
  - http://sf.net/projects/gfarm/
- Supported by NPO OSS Tsukuba Support Center
- Features
  - Scaled-out performance in wide area
  - Data access locality, file replica
  - No single point of failure
    - Automatic file replica creation in case of storage failure
    - Hot stand-by MDS
- HPCI Shared Storage, Japan Lattice Data Grid, NICT Science Cloud

ossTsukuba

HPCI Shared Storage

- HPCI – High Performance Computing Infrastructure
  - “K”, Hokkaido, Tohoku, Tsukuba, Tokyo, Titech, Nagoya, Kyoto, Osaka, Kyushu, RIKEN, JAMSTEC, AIST
- A 20PB single distributed file system consisting East and West sites
- Grid Security Infrastructure (GSI) for user ID
- Parallel file replication among sites
- Parallel file staging to/from each center

Initial Performance Result



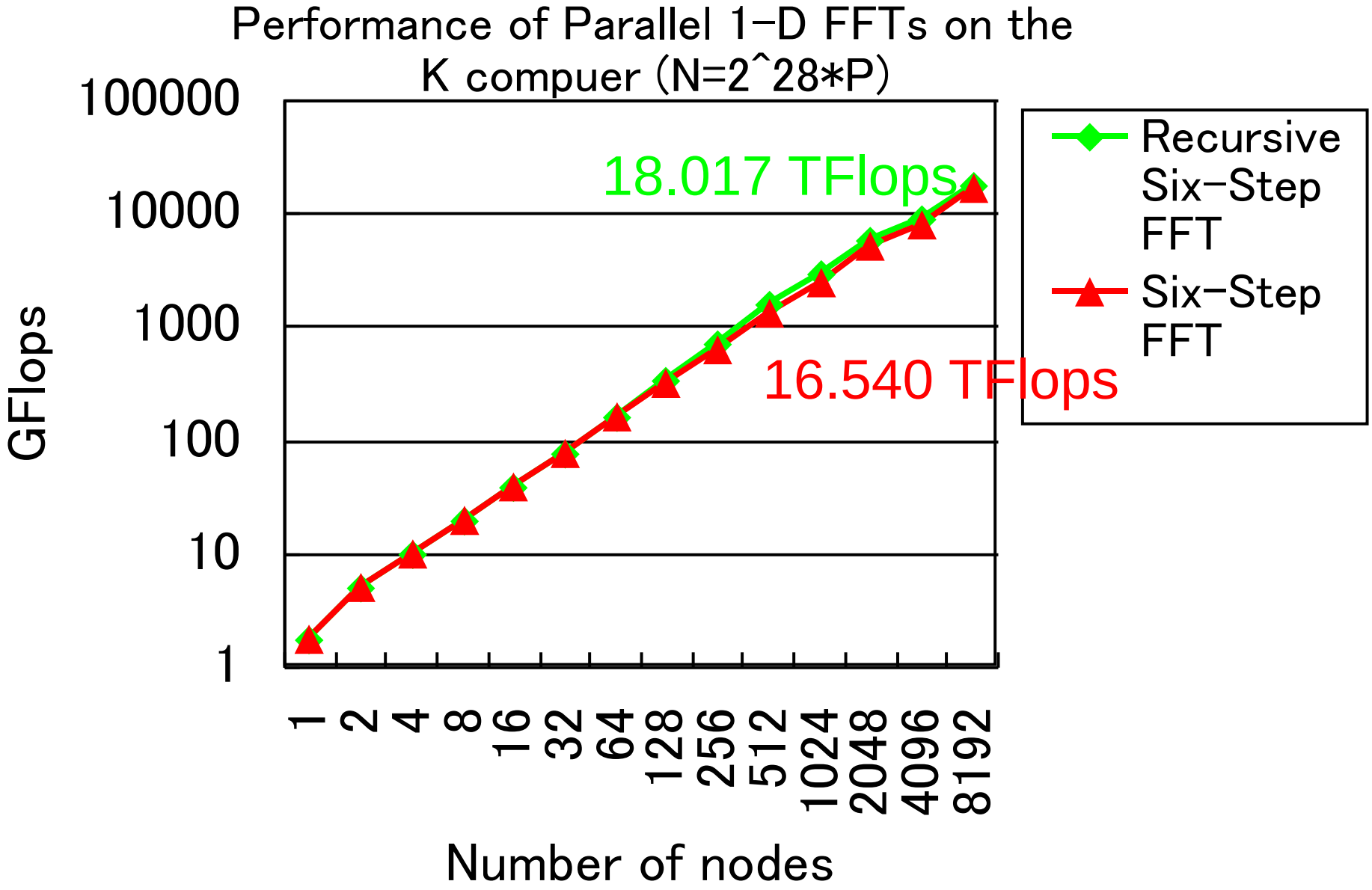
File copy performance of 300x of 1GB files

FFT and Parallel Numerical Libraries

(main contribution by D. Takahashi)

FFT and Parallel Numerical Libraries

- Overview of Research Activity
- Fast Fourier Transform (FFT)
  - FFTE: A High-Performance FFT Library
  - Recursive Six-Step FFT Algorithm
  - Performance Results of Parallel 1-D FFT on K computer
- Parallel Numerical Libraries: High Precision Arithmetic Operations
  - Triple and Quadruple Precision BLAS on GPUs
  - CUMP: The CUDA Multiple Precision Arithmetic Library


Performance of Parallel 1-D FFTs on the K compuer (N=2^28*P)
GFlops
100000
10000
1000
100
10
1
18.017 TFlops
16.540 TFlops
Recursive Six-Step FFT
Six-Step FFT
1
2
4
8
16
32
64
128
256
512
1024
2048
4096
8192
Number of nodes

Performance Summary

- We briefly introduced the FFTE library and performance results of parallel 1-D FFT on the K computer.
- Global FFT on the K computer (82,944 nodes) achieved first place (205.9 TFlops) in the 2012 HPC Challenge Class 1 Awards.
- High precision arithmetic operations will become increasingly necessary for emerging Exa-scale computing era.

Krylov Subspace Method

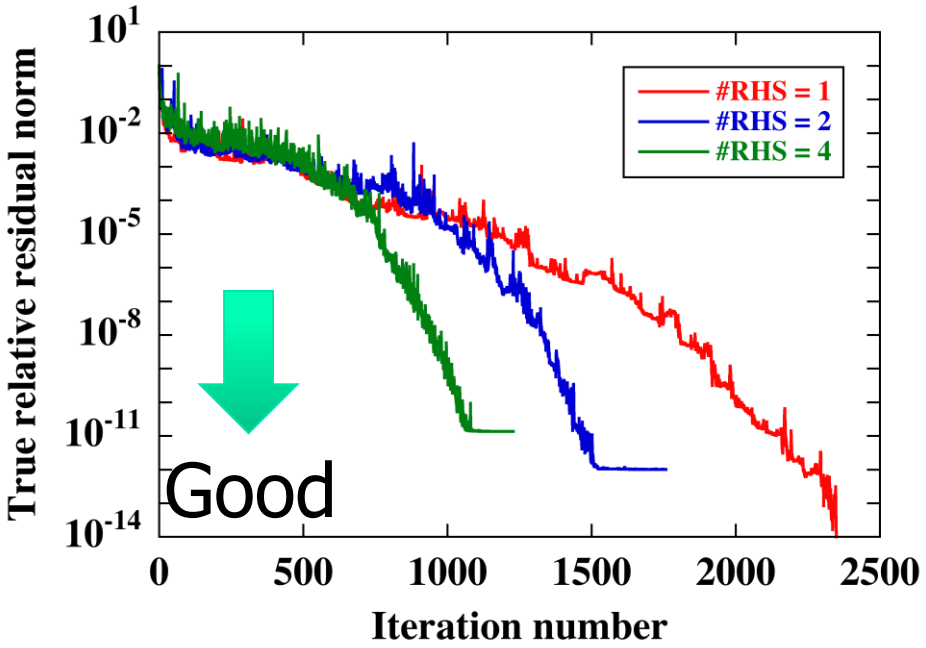
(main contribution by H. Tadano)

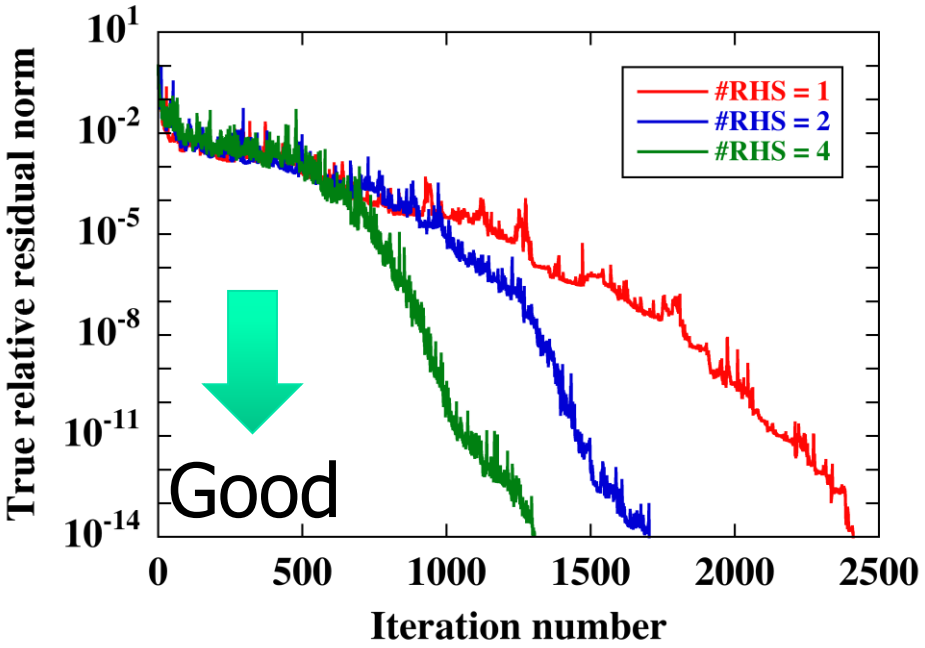
Development of stable and high accuracy solvers for linear system with multiple right-hand sides

- In this study, we consider solving linear systems with multiple right-hand sides.
- Block Krylov algorithms are efficient solvers for these linear systems in terms of the number of iterations. However, the accuracy of the obtained approximate solution often deteriorates when the number of right-hand sides is large.
- We have developed stable and high accuracy Block Krylov algorithms for solving linear systems with multiple right-hand sides.

Test problem

Linear systems derived from lattice quantum chromodynamics (QCD) calculation. Problem size: 1,572,864.
Number of right-hand sides: 1, 2, and 4.





(a) Block BiCGSTAB (Conventional method) (b) Block BiCGGR (Proposed method)

True relative residual history of Block Krylov subspace methods.
If this value is sufficiently small, the accuracy of the approximate solution is good.

External Budget and International Collaboration

- Major external budget
  - JST-CREST: “Low-Power and Dependable Parallel Processing Platform with Embedded Technology” (PI: M. Sato), 2006-2011
  - JST/CREST “System Software for Post Petascale Data Intensive Science” (PI: O. Tatebe), 2010-2015
  - JST/CREST “Unified Environment of Accelerated Computing and Communication towards Post-Petascale Era” (PI: T. Boku), 2012-2017
  - JST-ANR Japan-French collaboration: “FP3C project” (PI from Japan: M. Sato), 2011-2013
  - The G8RCI ECS (co-PI from Japan: M. Sato), 2011-2013
  - The G8RCI NuFuSE (PI from Japan: T. Boku), 2011-2013
- Major International/Domestic Collaboration Partners
  - NCAR, U. Tennessee, Princeton U. (USA), INRIA, CNRS, CEA (France), JSC, Juelich SC (Germany), KIAM (Russia)
  - RIKEN AICS, U. Tokyo, Kyoto U., Kyushu U., NII, Osaka U., KEK, AISTNagoya U., Tohoku U., etc.
  - Fujitsu, Renesas Electronics

Other roles in CCS

- **Administrative Committee Member for Computer Resource Management**
  - Most of the faculty staffs of HPC Division are the member of committee, and Division Leader is the chair to be responsible for management of all the computation resources including supercomputers
- **Supercomputer Procurement**
  - Most of the faculty staffs are involved in the procurement committee of CCS supercomputers as the central members for technical issues
- **Dual-Degree Program**
  - Students of Dual-Degree Program from application domains (major degree for PhD course) are usually supervised by HPC Division's staffs on Master Degree

Summary

- We are performing wide variety of research activities on HPC system hardware/software and numerical libraries on large scale parallel processing systems
- Hardware platform varies on PC clusters, K Computer, GPU, FPGA, etc.
- Collaborative work with most of Divisions of CCS for performance tuning, code development and code scaling