CCS-EPCC-WS July 05 2013

Feasibility Study on Future HPC Infrastructure

-- Towards Exascale Accelerated Computing --

Mitsuhisa Sato

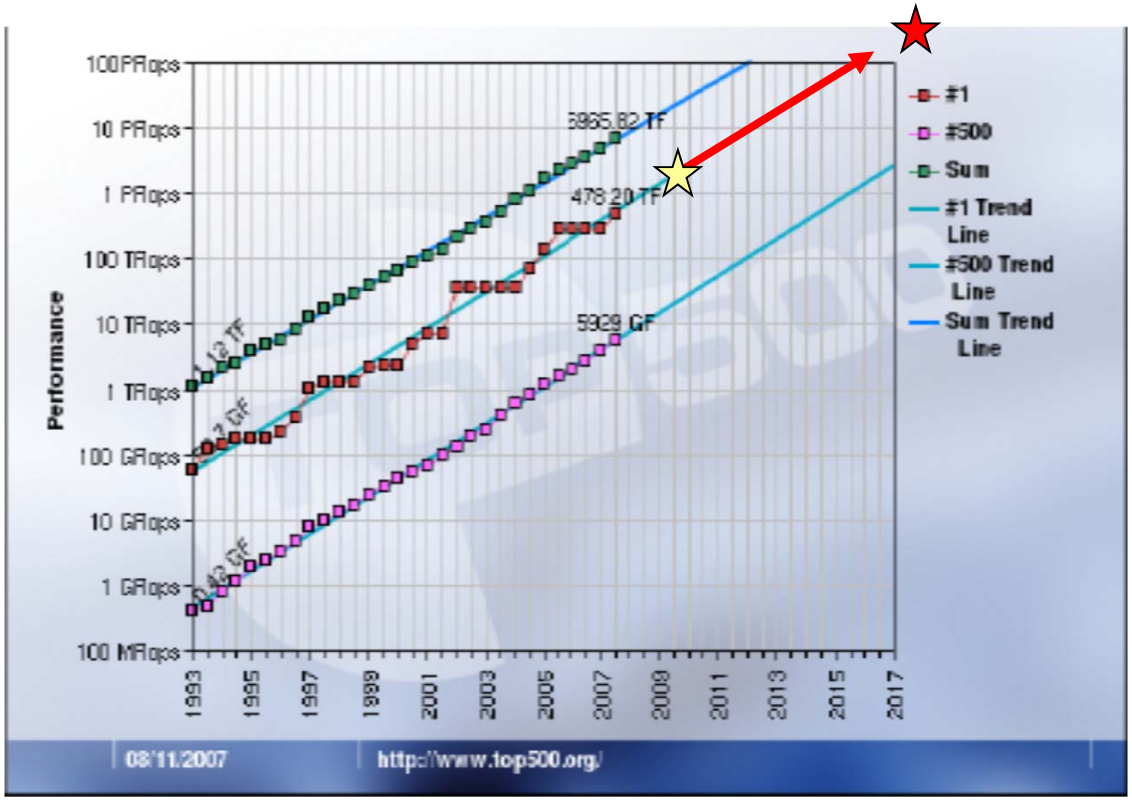
Professor, Center for Computational Sciences, University of Tsukuba /
Team Leader of Programming Environment Research Team, AICS, Riken

Outline

- Issues for Exascale computing
  - Why Accelerated computing?
  - CCS Research efforts for exascale computing
    - HA-PACS project
    - XcalableMP and XMP-dev extension for GPU Clusters
- HPCI-FS projects for Japanese post-petascale computing
  - "Study on exascale heterogeneous systems with accelerators"

Background: "Post-petascale computing“, toward exascale computing

- State of the art: Petascale computing infrastructure
  - US: Titan（27PF, 2012）,sequoia （>20PF,2012）
  - Japan: The K computer (>10PF, 2011), Tsubame 2.0
  - EU: PRACE machines (>5 PF, 2012-2013)
  - China: Tianhe-2
  - #cores 10^6
  - power >10 MW
- What's the next of "Petascale"?
  - Projection
  (and prediction)
  by Top500



What's "post-petascale" computing

- Post-petascale ⊃ "exaflops"
  - Several Possibilities and Alternatives for the next of Petascale, on the road to "exascale"

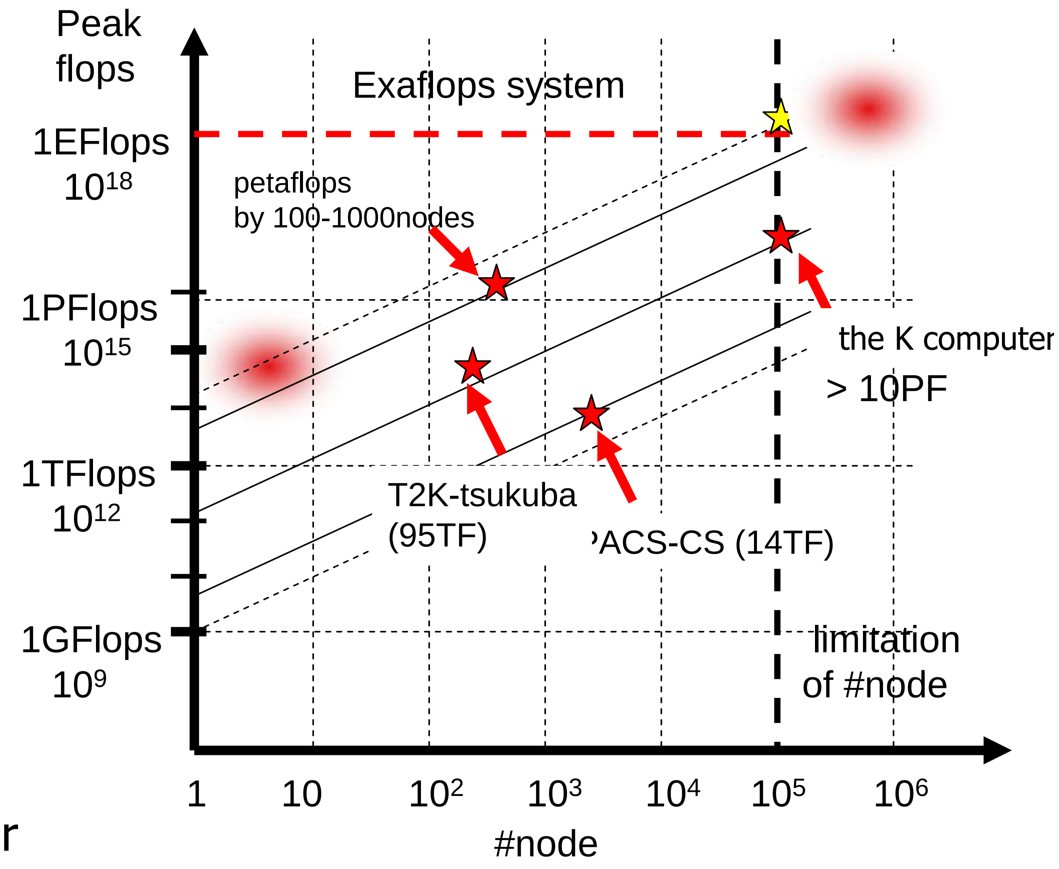
- Exascale=Extreme Scale ≠ "exaflops"
  - Embedded terascale (hand-held, 10-100W)
  - Departmental petascale (1-2 racks, 10-100kW)
  - (Inter)national exascale (100 racks, 25-50MW)

- Challenges
  - strong scaling = find 1000 × more parallelism in applications
  - fault tolerance = new algorithms + validation/verification
  - energy efficiency = new programming model(s), eg minimise data movement, intelligent powering
  - Novel hardware and programming, algorithms = GPGPUs, heterogeneous chips
  - massive (potentially corrupted) data and storage = new I/O models

Issues for exascale computing

- Two important aspects of post-petascale computing
  - Power limitation
    - < 20-30 MW
  - Strong-scaling
    - < 10^6 nodes, for FT
    - > 10TFlops/node
    - accelerator, many-cores

- Solution: Accelerated Computing
  - by GPGPU
  - by Application-specific Accelerator
  - by ... future acceleration device ...



simple projection of #nodes and peak flops

Research efforts for exascale computing in CCS

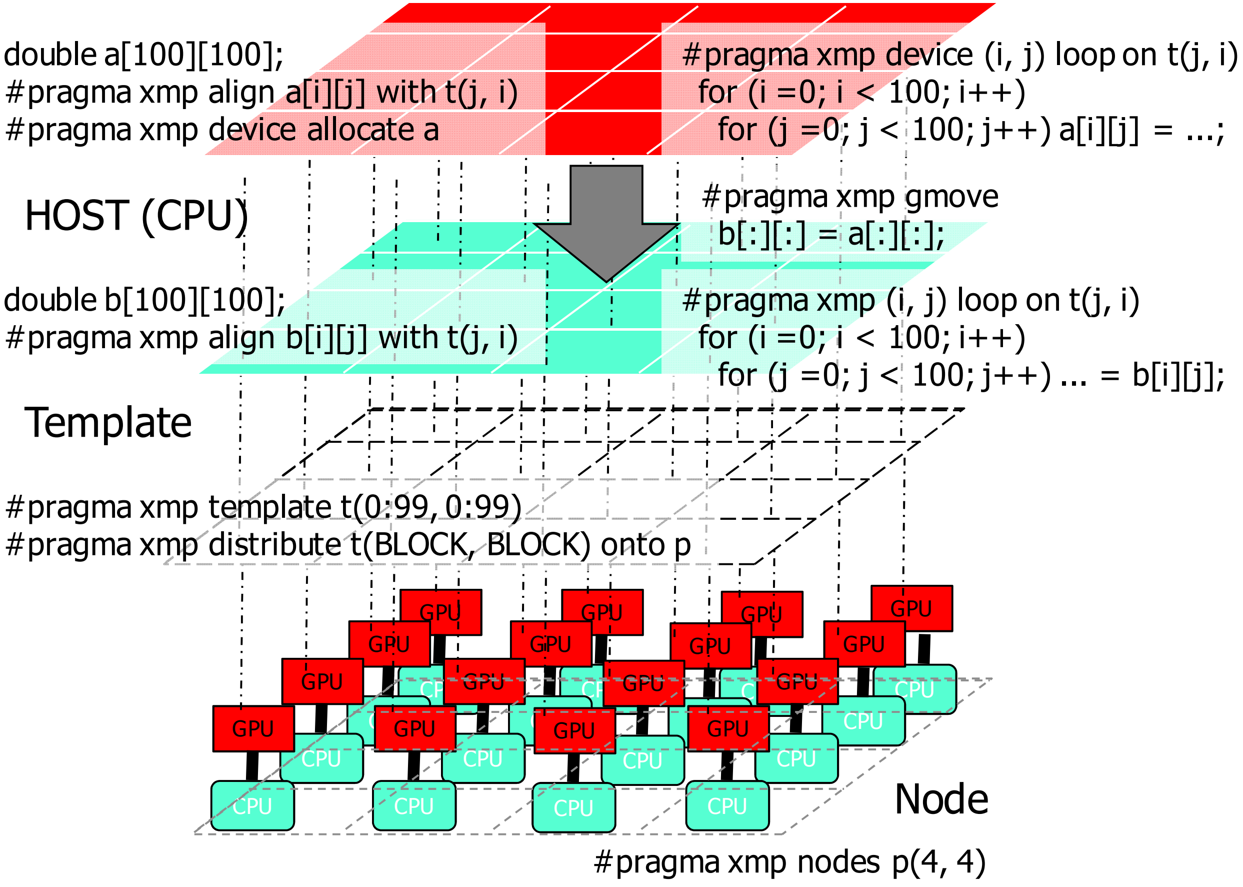
- HA-PACS project
  - HA-PACS (Highly Accelerated Parallel Advanced system for Computational Sciences)
  - “Advanced research and education on computational sciences driven by exascale computing technology”, funded by MEXT, Apr. 2011 – Mar. 2014, 3-year
  - TCA: Tightly Coupled Accelerator, Direct connection between accelerators (GPUs)
- Programming issue
  - XcalableMP and XMP-dev extension for GPU Clusters

XMP-dev: XcalableMP acceleration device extension

- Offloading a set of distributed array and operation to a cluster of GPU

- Hide complicated communication by reflect operation on distributed array allocated on GPUs
- Laplace solver: 21.6 times speedup on Laplace 4GPU(Tesla C2050) in Laplace solver.

DEVICE (GPU)



```
double a[100][100];
#pragma xmp align a[i][j] with t(j, i)
#pragma xmp device allocate a
```

```
#pragma xmp device (i, j) loop on t(j, i)
 for (i =0; i < 100; i++)
  for (j =0; j < 100; j++) a[i][j] = ...;
```

```
#pragma xmp gmove
 b[:][:] = a[:][:];
```

```
double b[100][100];
#pragma xmp align b[i][j] with t(j, i)
```

```
#pragma xmp (i, j) loop on t(j, i)
 for (i =0; i < 100; i++)
  for (j =0; j < 100; j++) ... = b[i][j];
```

```
#pragma xmp template t(0:99, 0:99)
#pragma xmp distribute t(BLOCK, BLOCK) onto p
```

```
#pragma xmp nodes p(4, 4)
```

```
#pragma xmp device replicate(u, uu)
{
#pragma xmp device replicate_sync in (u)
 for (k = 0; k < ITER; k++) {
#pragma xmp device reflect (u)
#pragma xmp device loop (x, y) on t(x, y) thre
   for (y = 1; y < N-1; y++)
    for (x = 1; x < N-1; x++)
      uu[y][x] = (u[y-1][x] + u[y+1][x] +
                u[y][x-1] + u[y][x+1]) / 4.0;
#pragma xmp device loop (x, y) on t(x, y) thre
   for (y = 1; y < N-1; y++)
    for (x = 1; x < N-1; x++)
      u[y][x] = uu[y][x];
 }
#pragma xmp device replicate_sync out (u)
} // #pragma xmp device replicate
```

The SDHPC white paper and Japanese “Feasibility Study" project

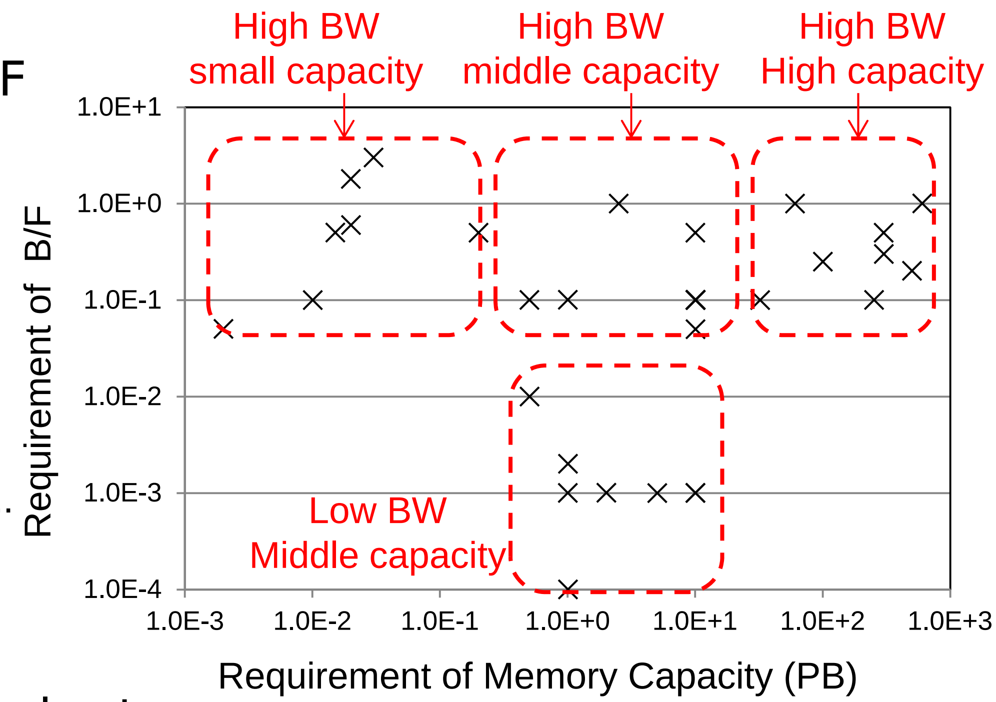
- WGs ware orgainzed for drafting the white paper for Strategic Direction/Development of HPC in JAPAN by young Japanese researchers with advisers (seniors)
- Contents
  - Science roadmap until 2020 and List of application for 2020’s
  - Four types of hardware architectures identified and performance projection in 2018 estimated from the present technology trend
  - Necessity of further research and development to realize the science roadmap

- For “Feasibility Study" project, 4 research teams were accepted
  - Application study team leaded by RIKEN AICS (Tomita)
  - System study team leaded by U Tokyo (Ishikawa)
    - Next-generation “General-Purpose” Supercomputer
  - System study team leaded by U Tsukuba (Sato)
    - Study on exascale heterogeneous systems with accelerators
  - System study team leaded by Tohoku U (Kobayashi)
- Projects were started from July 2012 (1.5 year) …

System requirement analysis for Target sciences

(From SDHPC white paper)

- System performance
  - FLOPS: 800 – 2500PFLOPS
  - Memory capacity: 10TB – 500PB
  - Memory bandwidth: 0.001 – 1.0 B/F
  - Example applications
    - Small capacity requirement
      - MD, Climate, Space physics, …
    - Small BW requirement
      - Quantum chemistry, …
    - High capacity/BW requirement
      - Incompressibility fluid dynamics, …

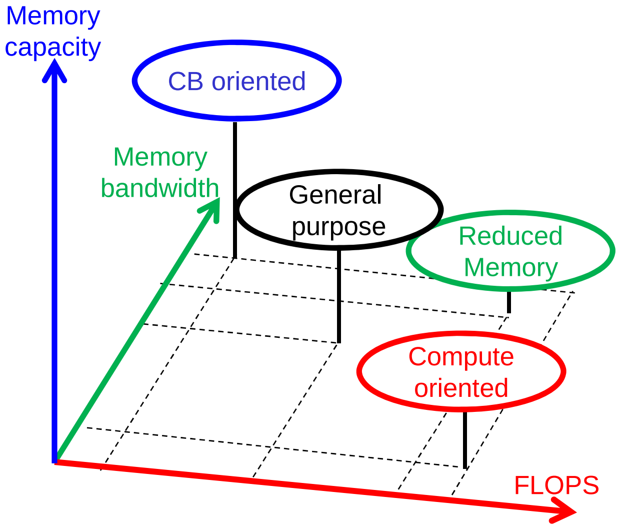


- Interconnection Network
  - Not enough analysis has been carried out
  - Some applications need >1us latency and large bisection BW
- Storage
  - There is not so big demand

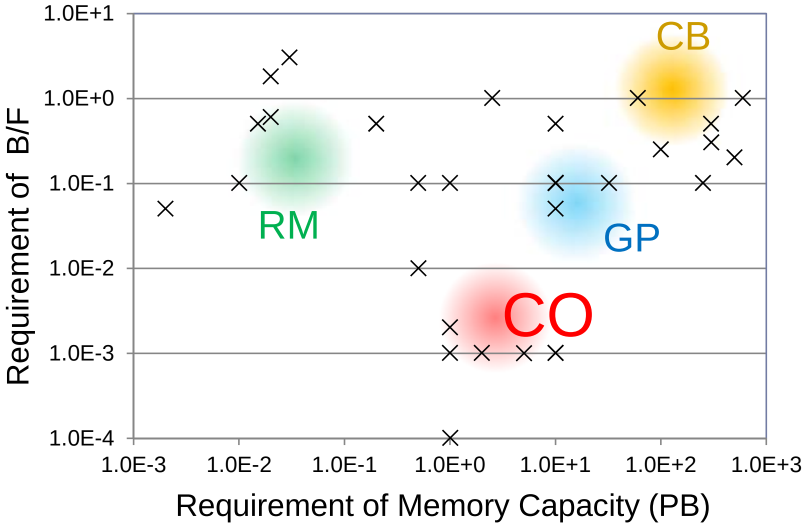
(From SDHPC white paper)

Alternatives of ExaScale Architecture

- Four types of architectures are identified for exascale:
  - General Purpose (GP)
    - Ordinary CPU-based MPPs
    - e.g.) K-Computer, GPU, Blue Gene, x86-based PC-clusters
  - Capacity-Bandwidth oriented (CB)
    - With expensive memory-I/F rather than computing capability
    - e.g.) Vector machines
  - Reduced Memory (RM)
    - With embedded (main) memory
    - e.g.) SoC, MD-GRAPE4, Anton
  - Compute Oriented (CO)
    - Many processing units
    - e.g.) ClearSpeed, GRAPE-DR, GPU?

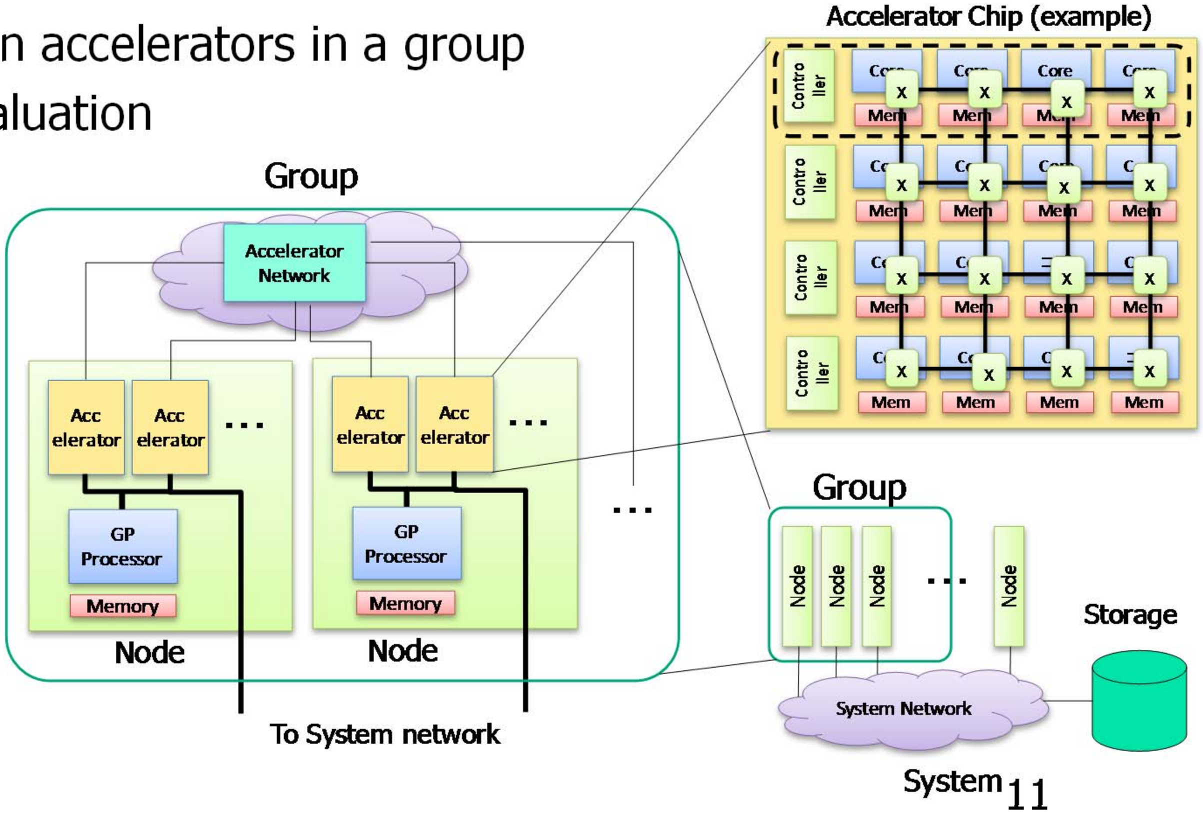


Mapping of Architectures



Study on exascale heterogeneous systems with accelerators (U Tsukuba proposal)

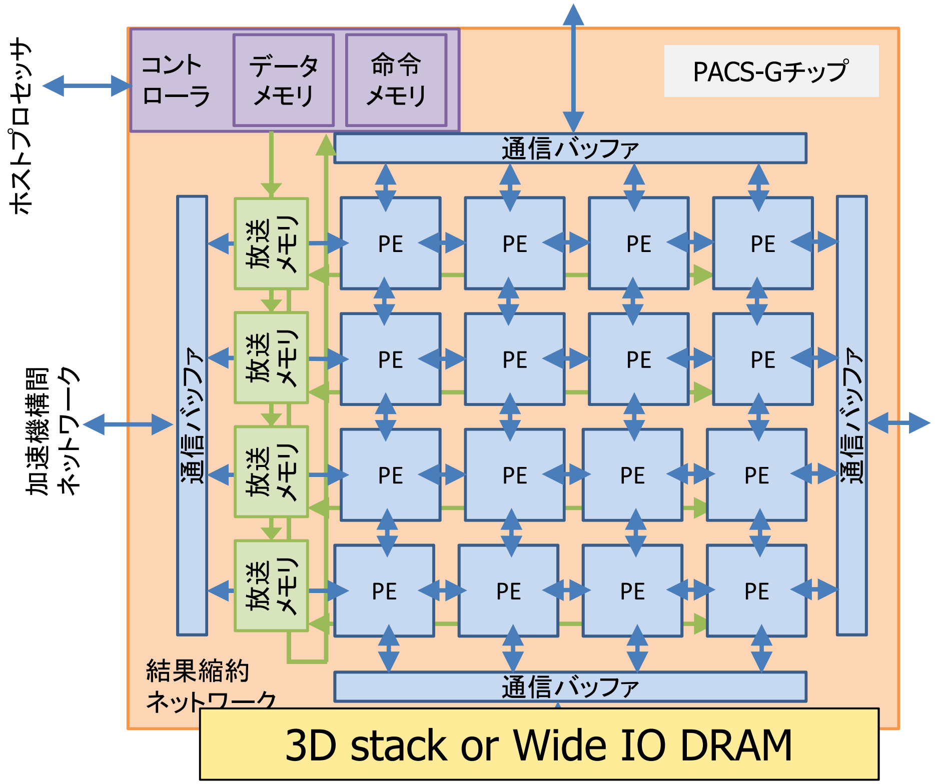
- Two keys for exascale computing
  - Power and strong-scaling
- We study “exascale” heterogeneous systems with accelerators of many-cores. We are interested in:
  - Architecture of accelerators, core and memory architecture
  - Special-purpose functions
  - Direct connection between accelerators in a group
  - Power estimation and evaluation
  - Programming model and computational science applications
  - Requirement for general-purpose system
  - etc …



PACS-G: a straw man architecture

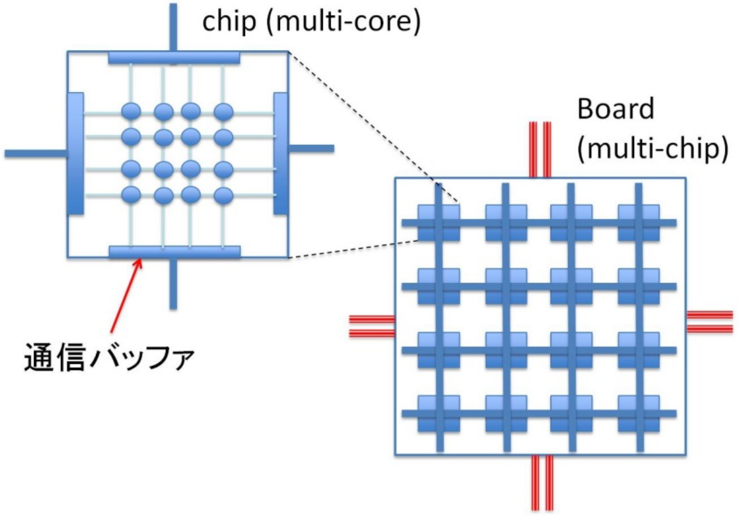
- SIMD architecture, for compute oriented apps (N-body, MD), and stencil apps.
- 4096 cores (64x64), 2FMA@1GHz, 4GFlops x 4096 = 16TFlops/chip
- 2D mesh (+ broardcast/reduction) on-chip network for stencil apps.
- We expect 14nm technology at the range of year 2018-2020,
  Chip dai size: 20mm x 20mm
- Mainly working on on-chip memory (size 512 MB/chip, 128KB/core),
  and, with module memory by
  3D-stack/wide IO DRAM
  memory (via 2.5D TSV),
  bandwidth 1000GB/s,
  size 16-32GB/chip
- No external memory (DIM/DDR)

- 250 W/chips expected
- 64K chips for 1 EFLOPS (at peak)



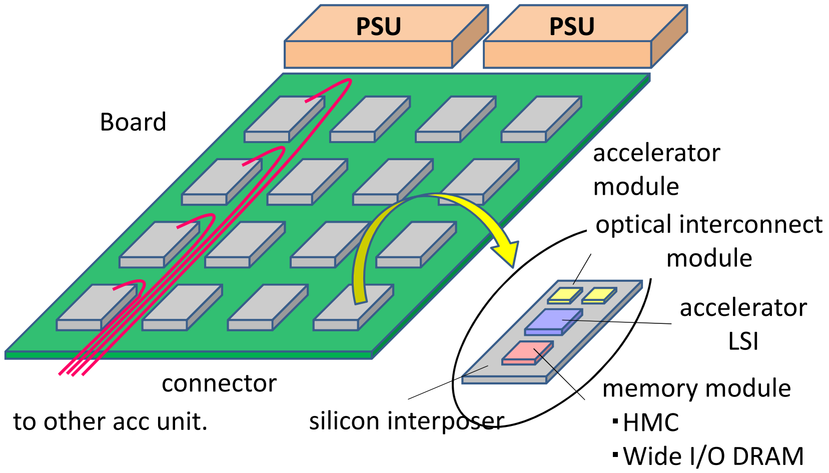
PACS-G: a straw man architecture

- A group of 1024～2048 chips are connected via accelerator network (inter-chip network)
- 25 – 50Gpbs/link for inter-chip: If we extend 2-D mesh network to the (2D-mesh) external network in a group, we need 200～400GB/s (= 32 ch. x 25～50Gbps x 2(bi-direction))
- For 50Gpbs data transfer, we may need direct optical interconnect from chip.
- I/O Interface to Host: PCI Express Gen 4 x16 (not enough!!!)



interconnect between chips (2D mesh)

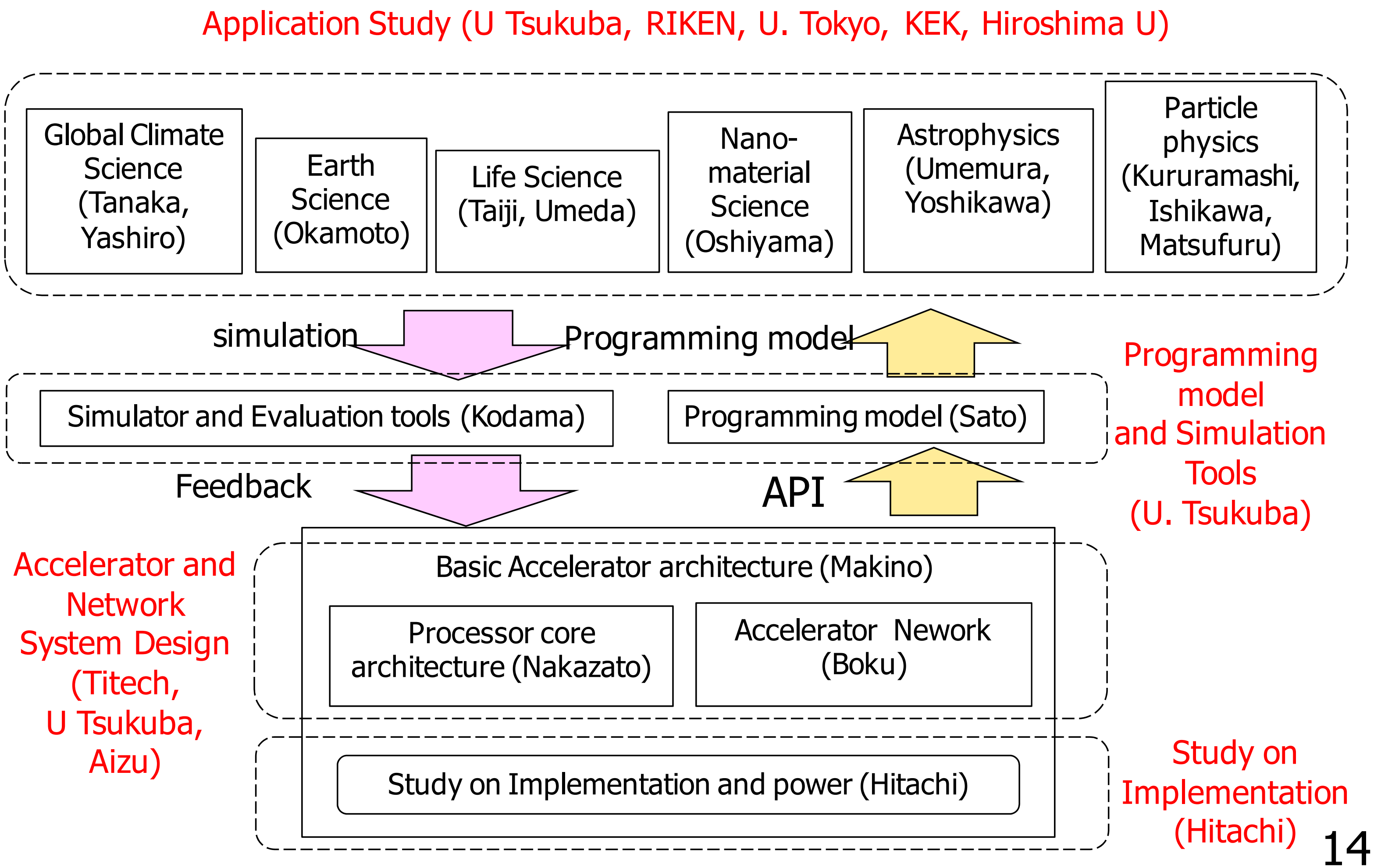
- Programming model: XcalableMP + OpenACC
  - Use OpenACC to specify offloaded fragment of code and data movement
  - To align data and computation to core, we use the concept "template" of XcalableMP (virtual index space). We can generate code for each core.
  - (And data parallel lang. like C*)



An example of implementation (for 1U rack)

Project organization

- Joint project with Titech (Makino), Aizu U (Nakazato), RIKEN (Taiji), U Tokyo, KEK, Hiroshima U, and Hitachi as a super computer company
- Target apps: QCD in particle physics, tree N-body, HMD in Astrophysics, MD in life sci., FDM of earthquake, FMO in chemistry, NICAM in climate sci.



Application Study (U Tsukuba, RIKEN, U. Tokyo, KEK, Hiroshima U)
Global Climate Science (Tanaka, Yashiro)
Earth Science (Okamoto)
Life Science (Taiji, Umeda)
Nano-material Science (Oshiyama)
Astrophysics (Umemura, Yoshikawa)
Particle physics (Kururamashi, Ishikawa, Matsufuru)
simulation
Programming model
Programming model and Simulation Tools (U. Tsukuba)
Simulator and Evaluation tools (Kodama)
Programming model (Sato)
Feedback
API
Accelerator and Network System Design (Titech, U Tsukuba, Aizu)
Basic Accelerator architecture (Makino)
Processor core architecture (Nakazato)
Accelerator Nework (Boku)
Study on Implementation and power (Hitachi)
Study on Implementation (Hitachi)

Current status and plan

- We are now working on performance estimation by co-design process
  - 2012 (done): QCD, N-body, MD, HMD
  - 2013: earth quake sim, NICAM (climate), FMO (chemistry)
    - When all data fits on on-chip memory, ratio B/F is 4 B/F, total mem size 1TB/group
    - When data fits into module memory, ratio B/F is 0.05B/F, total mem size 32TB/group
- Also, developing simulators (clock-level/instruction level) for more precious and quantitative performance evaluation
- Compiler development (XMP and OpenACC)
- (Re-)Design and investigation of network topology
  - 2D mesh is sufficient? or, other alternative?
- Code development for apps using Host and Acc, including I/O
- Precious and more detail estimation of power consumptions