

Challenges for Parallel Programming Models and Languages of post- petascale and exascale computing

Post-K and post-T2K project

Mitsuhisa Sato Team Leader of Architecture Development Team

Exascale supercomputer project

RIKEN Advance Institute of Computational Science (AICS)

2015/Oct/1st

Outline

- **Background: Japanese supercomputing infrastructure, HPCI**
 - Post-T2K project: a collaboration with U. Tokyo and U. Tsukuba.
- **FLAGSHIP 2020 project**
 - to develop the next Japanese flagship computer system, “post-K”
 - “co-design” effort to design the system
- **Challenges for Parallel Programming Models and Languages for post-petascale and exascale computing**

AICS and Supercomputer Centers in Japanese Universities

AICS, RIKEN :
K computer (10 Pfflops, 4PB)
Available in 2012



Kyoto Univ.
T2K Open Supercomputer
(61.2 Tflops, 13 TB)



Osaka Univ. :
SX-9 (16Tflops, 10TB)
SX-8R (5.3Tflops, 3.3TB)
PCCluster (23.3Tflops, 2.9TB)



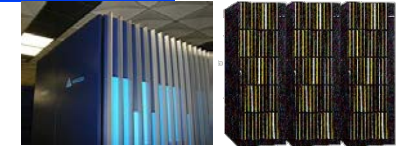
Kyushu Univ. :
PC Cluster (55Tflops, 18.8TB)
SR16000 L2 (25.3Tflops, 5.5TB)
PC Cluster (18.4Tflops, 3TB)



Nagoya Univ. :
FX1(30.72Tflops, 24TB)
HX600(25.6Tflops, 10TB)
M9000(3.84Tflops, 3TB)



Hokkaido Univ. :
SR11000/K1(5.4Tflops, 5TB)
PC Cluster (0.5Tflops, 0.64TB)



Tohoku Univ. :
NEC SX-9(29.4Tflops, 18TB)
NEC Express5800 (1.74Tflops, 3TB)



Univ. of Tsukuba :
T2K Open Supercomputer
95.4Tflops, 20TB

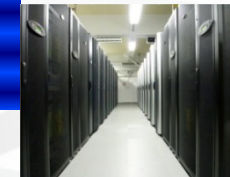


Univ. of Tokyo :
T2K Open Supercomputer
(140 Tflops, 31.25TB)

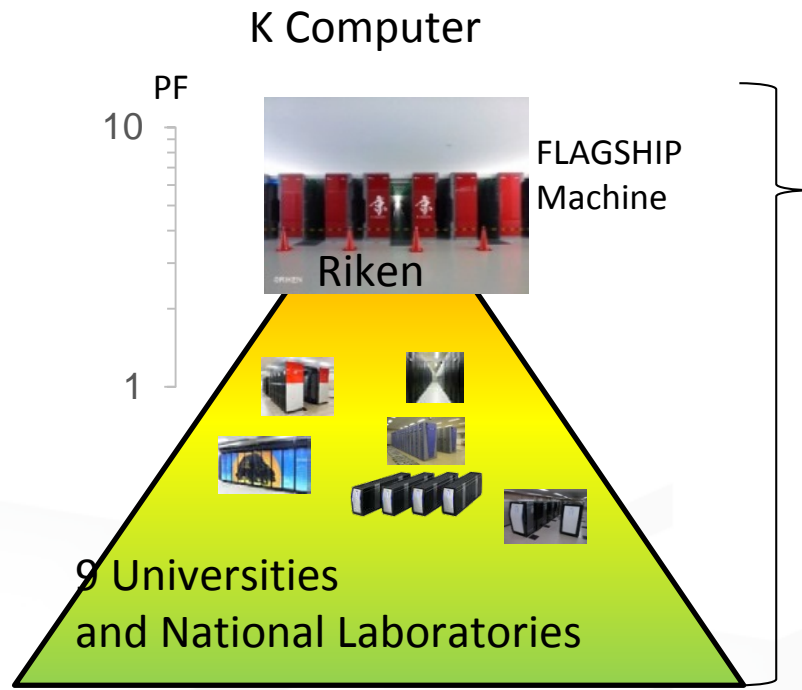


A 1 Pflops machine without accelerator will be installed by the end of 2011

Tokyo Institute of Technology :
Tsubame 2
(2.4 Pflops, 100TB)



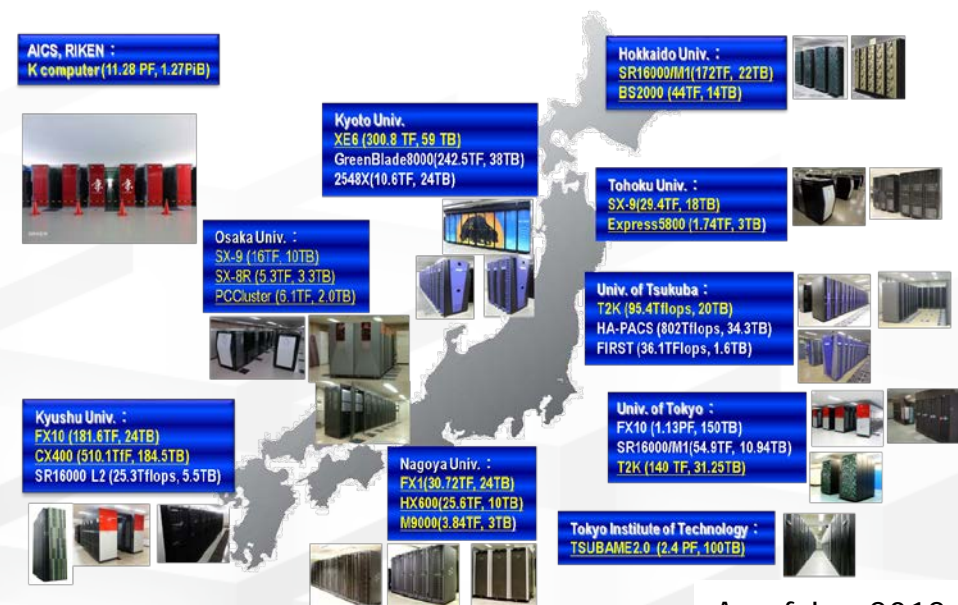
Supercomputers in Japan



HPCI (High Performance Computing Infrastructure) is formed from those machines, called leading machines

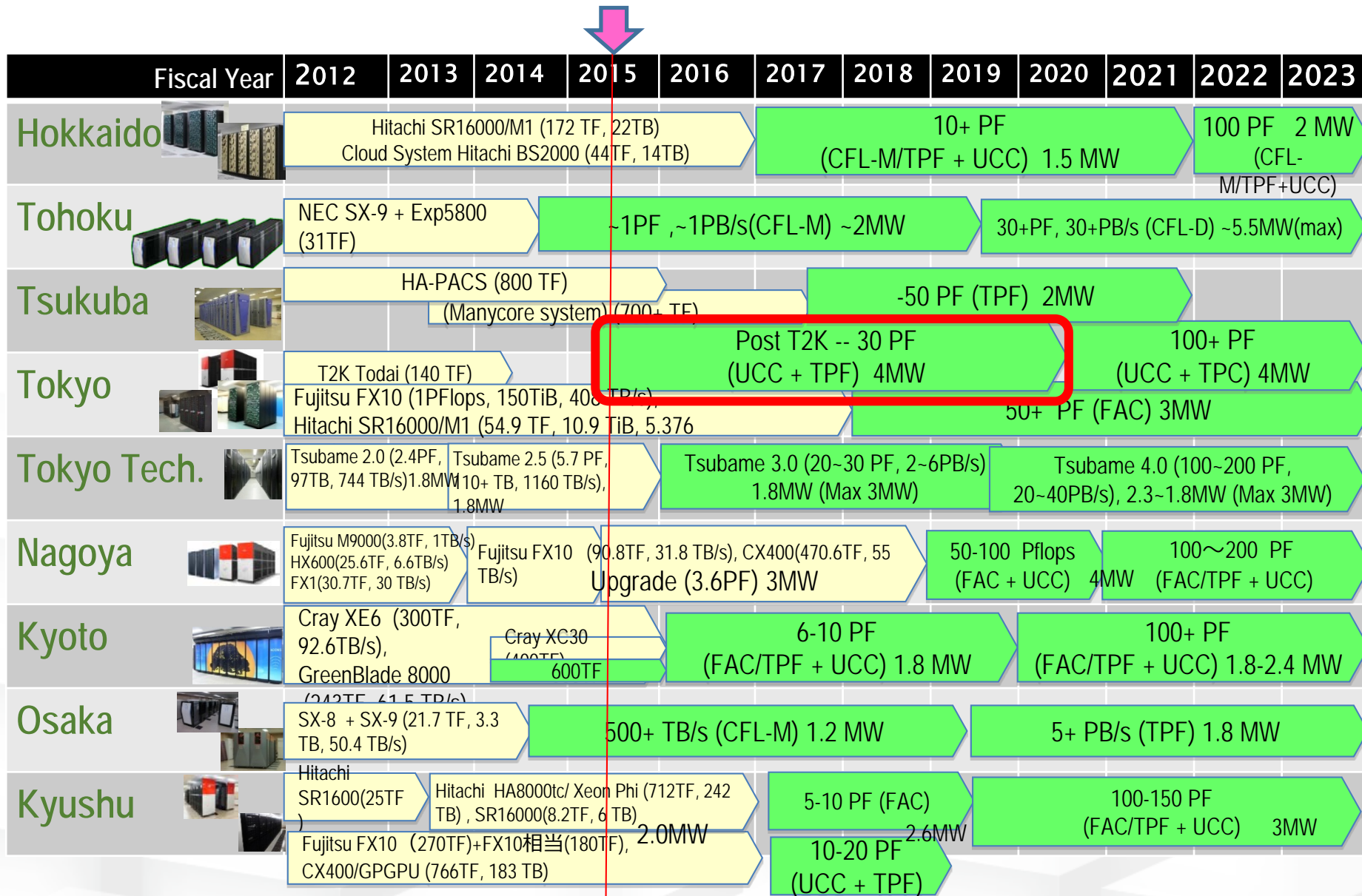
Features: Single sign-on
Shared storage (Distributed file system)

- ❑ Each supercomputer center has one, two or more supercomputers.
- ❑ Each supercomputer center replaces their machines every 4.5 to 6 years.



As of Jun 2012

Supercomputer Centers operated at Japanese Universities and Plan/schedule



JCAHPC and Post-T2K project

- The post T2K project, a project following T2K Open-supercomputer Alliance, is aiming to build and install a large-scale manycore cluster system to provide services for computational science researchers in Japan (not limited to Japan).
- This is a joint effort by Univ. Tokyo and Univ. Tsukuba, in collaboration with Kyoto Univ.

■ The two universities agreed to established a virtual organization, **Joint-Center for Advanced High Performance Computing (JCAHPC)** to develop and procure, run the system.

T2K Open Supercomputer Alliance (2007~)

- T2K :Tsukuba, Tokyo and Kyoto
- Primary aiming at design of common specification of new supercomputers.
- Promotion of collaborative work on research, education, grid operation, ..., for inter-disciplinary computational (& computer) science.
- *Open* hardware architecture with commodity devices & technologies.
- *Open* software stack with open-source middleware & tools.
- *Open* to user's needs not only in FP & HPC field but also IT field.

Kyoto Univ.

416 nodes (61.2TF) / 13TB
 Linpack Result:
 Rpeak = 61.2TF (416 nodes)
 Rmax = 50.5TF (82.5%)

Univ. Tokyo

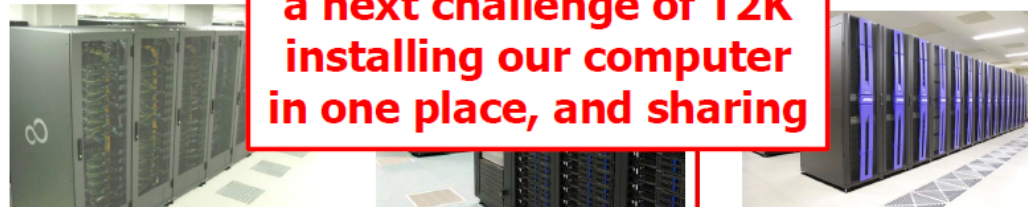
952 nodes (140.1TF) / 31TB

Univ. Tsukuba

648 nodes (95.4TF) / 20TB

Linpack Result:
 92.0TF (625 nodes)
 76.5TF (83.15%)

post T2K project is a next challenge of T2K installing our computer in one place, and sharing



Mission ① Research and Development of large-scale HPC system

- Design of large-scale HPC system by adopting advanced and timely (commodity) technologies.
 - "Co-design" of the system with apps.
 - A key point of "development" of modern HPC system is using advanced commodities and configuring them in "optimal" way.
 - Manycore processor is a "hot" and "advanced" commodity for building high performance systems.
 - ⇒ to produce a draft of Specification of the system for the procurement.

- As a co-design effort, research and development several software incl. OS, programing lang., math libs are being carried out.
 - Operating system for manycore - McKernel
 - Programing lang. for manycore - XcalableMP
 - Others, ... and developed software are to be deployed in the system.

- Collaboration with other universities (incl. U. Kyoto) and PC Cluster Consortium, Japan.

Items for co-design

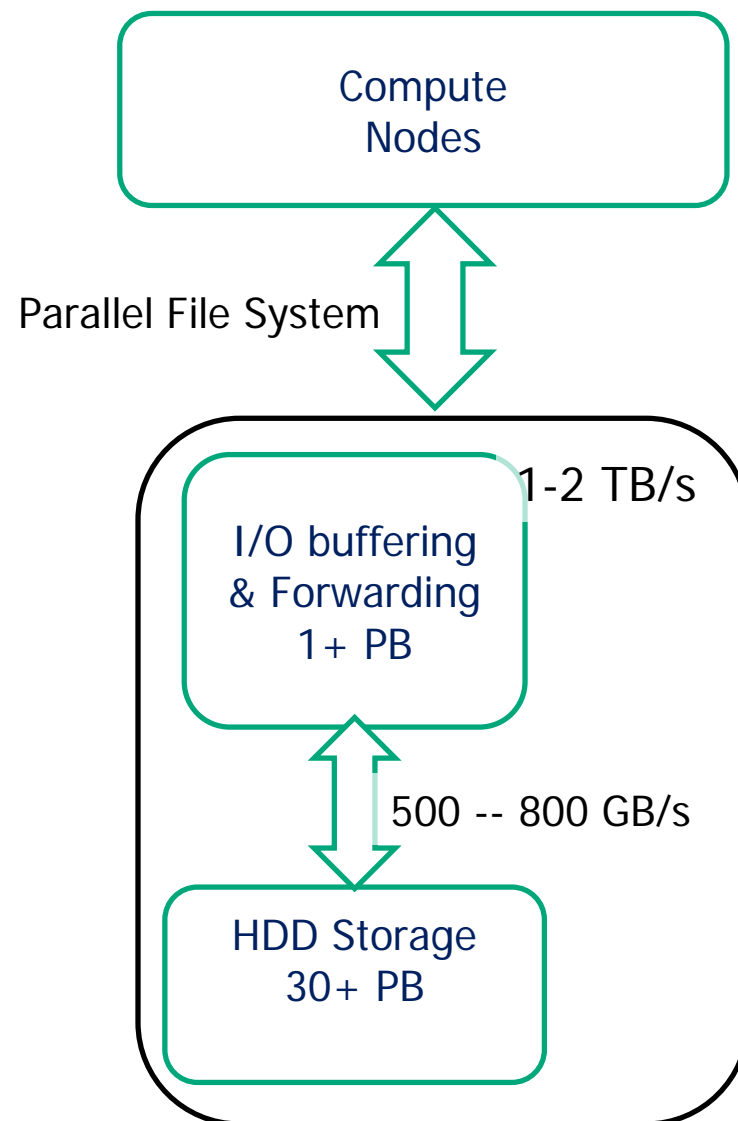
- constraints
 - budget, power (<4MW), space
- node processor
 - commodity
 - choice of architecture (#core, manycore)
 - memory size
 - ...
- Network
 - Network topology (e.g. IB for Fat-tree)
 - hierarchical structure for partitioning
- Storage
 - Global storage / local storage
 - How to access from node, staging data ...
- What are benchmark programs. How to benchmark.
- Programing model
 - MPI/OpenMP/PGAS (XcalableMP)
- Operating System
 - Linux and **McKernel**
- Job scheduler
 - Network topology aware scheduling
 - Accounting to enable management by each university
 - Provision (for power saving, exchange OS, ...)

Draft Specification of PostT2K Hardware

- Total Performance
 - Peak FP performance: ~ 30 PF
 - Memory: ~ 900 TB

- Node
 - Peak floating point performance: 3+ TF
 - Memory: ~ 115 GB

- Network
 - Fat Tree, 100Gbps



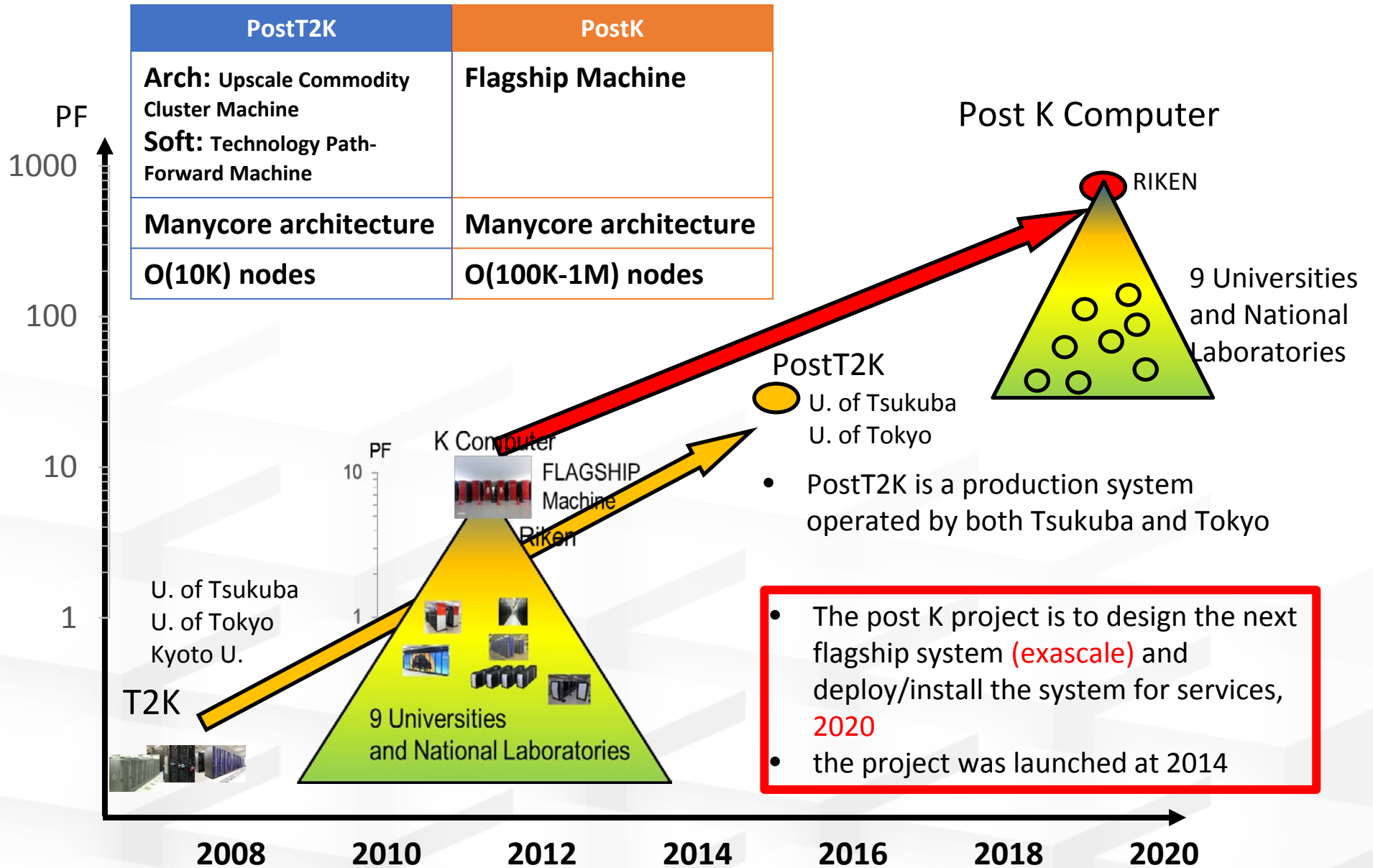
Mission ① Research and Development of large-scale HPC system

- Design of large-scale HPC system by adopting advanced and timely (commodity) technologies.
 - "Co-design" of the system with apps.
 - A key point of "development" of modern HPC system is using advanced commodities and configuring them in "optimal" way.
 - Manycore processor is a "hot" and "advanced" commodity for building high performance systems.
 - ⇒ to produce a draft of Specification of the system for the procurement.

- As a co-design effort, research and development several software incl. OS, programing lang., math libs are being carried out.
 - Operating system for manycore - McKernel
 - Programing lang. for manycore - XcalableMP
 - Others, ... and **developed software are to be deployed in the system.**

- Collaboration with other universities (incl. U. Kyoto) and PC Cluster Consortium, Japan.

Towards the Next Flagship Machine



FLAGSHIP 2020 Project

- **Missions**

- Building the Japanese national flagship supercomputer, Post K, and
- Developing wide range of HPC applications, running on Post K, in order to solve social and science issues in our country.

- **Planned Budget**

- 110 Billion JPY (about 0.91 Billion USD at the rate 120 JPY/\$)
- including research, development (NRE) and acquisition/deploy, and application development

- **Post K Computer: System and Software**

- RIKEN AICS is in charge of development
- Fujitsu is selected as a vendor partner
- Started from 2014

| CY | 2014 | | | | 2015 | | | | 2016 | | | | 2017 | | | | 2018 | | | | 2019 | | | | 2020 | | | |
|----|--------------|----|----|----|---------------------------|----|----|----|------|----|----|----|---|----|----|----|-----------|----|----|----|------|----|----|----|------|----|----|----|
| | Q1 | Q2 | Q3 | Q4 | Q1 | Q2 | Q3 | Q4 | Q1 | Q2 | Q3 | Q4 | Q1 | Q2 | Q3 | Q4 | Q1 | Q2 | Q3 | Q4 | Q1 | Q2 | Q3 | Q4 | Q1 | Q2 | Q3 | Q4 |
| | Basic Design | | | | Design and Implementation | | | | | | | | Manufacturing, Installation, and Tuning | | | | Operation | | | | | | | | | | | |

Current status of the post-K project

- **The procurement for the development of the post-K computer system was done.**
 - Fujitsu was selected as the vendor partner.

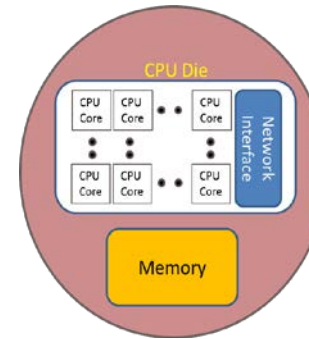
- **In the specification of RFP:**
 - Constraints are:
 - Power capacity (about 30MW)
 - Space for system installation (in Kobe AICS building)
 - Budget (money) for development (NRE) and production.
 - ... some degree of compatibility to the current K computer.

- **We are now finishing the “basic design” of the system with the vendor partner.**

- **The system should be designed to maximize the performance of applications in each computational science field.**
 - "Co-design" is a keyword!

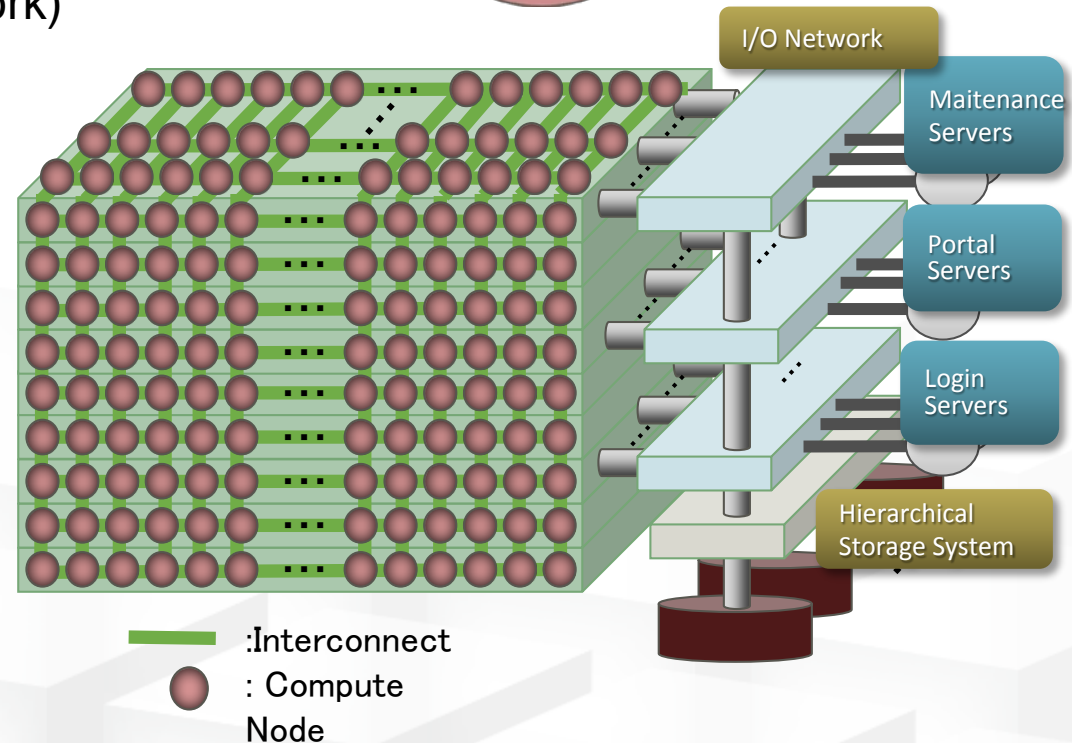
Post K Computer

- ✓ CPU
 - Many-core with Interconnect interface integrated on chip
 - Power Knob feature for saving power
- ✓ Interconnect
 - TOFU (mesh/torus network)



Co-design may include:

- Compute Node Features
 - Core architecture, FP performance
 - Memory hierarchy, control, capacity, and bandwidth
- Network Performance
- I/O Performance



- **“Co-design” in Wikipedia**

- **“Co-design** or **codesign** is a product, service, or organization development process where design professionals empower, encourage, and guide users to develop solutions for themselves.”

... ..

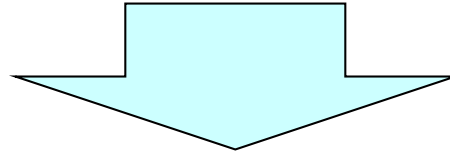
- “The phrase co-design is also used in reference to the simultaneous development of interrelated software and hardware systems. The term co-design has become popular in mobile phone development, where the two perspectives of hardware and software design are brought into a co-design process”

- **The co-design of HPC must optimize and maximize the benefits to cover many applications as possible.**

- different from "co-design" in embedded systems. For example, in embedded field, co-design sometimes includes "specialization" for a particular applications.
- On the other hands, in HPC, the system will be shared by many applications.

Why “co-design” is needed in very high-end HPC and exascale?

- In modern very high-end parallel system, more performance can be delivered (even upto “exascale”) by increasing the number of nodes, but ...



- **We need to design the system by trade-off between “energy/power” and “cost” and performance**
 - to satisfy constraints of “energy/power” and “cost”
 - to maximize the performance.

We need to design the system by taking characteristics of applications in account

⇒ “codesign” in HPC

- **The elements of "co-design" in our post-K project may include**
 - Note that we are going to design processor/network and system with the selected partner vender.
 - Different from supercomputer acquisition in universities.

Co-design elements in HPC systems

● Hardware/architecture

- Node architecture (#core, #SIMD, etc...)
- cache (size and bandwidth)
- network (topologies, latency and bandwidth)
- memory technologies (HBM and HMC, ...)
- specialized hardware
- #nodes
- Storage, file systems
- ... system configurations

■ System software

- Operating system for many core architecture
- communication library (low level layer, MPI, PGAS)
- Programming model and languages
- DSL, ...

■ Algorithm and math lib

- Dense and Sparse solver
- Eigen solver
- ... Domain-specific lib and framework

■ And, Applications!

What applications does our co-design target for?



- **SPIRE (Strategic Programs for Innovative Research) Program for the K computer**
 - The projects were organized around 2011.

- **For the post-K system,**
 - The committee (from academia and industry) was organized by our government to identify "priority research area" (9) and "frontier research area"(5) to be exploited by the post-K system.

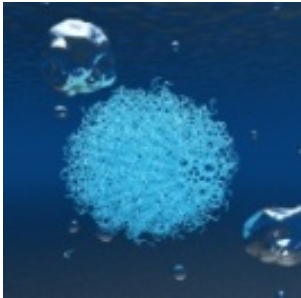
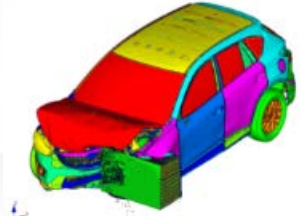
 - The call for project proposals for these "priority research area" and "frontier research area" has been issued.

 - The projects for "priority research area" were accepted for the design of target apps and the co-design of the post-K system.

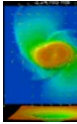
9 social and scientific priority issues (1/3)

| Category | Priority issues |
|---|---|
| <p data-bbox="109 287 395 519">Achievement of a society that provides health and longevity</p>  | <p data-bbox="437 287 1773 362"><u>① Innovative drug discovery infrastructure through functional control of biomolecular systems</u></p> <p data-bbox="437 379 1922 505">Develop ultra-high speed molecular simulations to achieve not only functional inhibition but also functional control of many biomolecules including factors that cause side-effects, in order to discover safe and highly effective drugs.</p> <p data-bbox="437 544 1902 619"><u>② Integrated computational life science to support personalized and preventive medicine</u></p> <p data-bbox="437 636 1897 805">Exploit large-scale analysis of healthcare and medical “Big Data” and biomedical simulations (heart, brain and nervous system etc.) on the basis of optimal models obtained using these data, in order to support medicine tailored to each individual and preventive medicine that can extend healthy life expectancy.</p> |
| <p data-bbox="109 846 339 1025">Disaster prevention and global climate</p>  | <p data-bbox="437 846 1777 922"><u>③ Development of integrated simulation systems for hazard and disaster induced by earthquake and tsunami</u></p> <p data-bbox="437 939 1929 1108">Develop an integrated simulation system for hazard and disaster which are induced by earthquake and tsunami and are not estimated based on past experiences, by improving and strengthening a package of related analysis methods. The system is to be implemented in disaster management systems of the Cabinet Office and local governments, etc.</p> <p data-bbox="437 1146 1918 1222"><u>④ Advancement of meteorological and global environmental predictions utilizing observational “Big Data”</u></p> <p data-bbox="437 1239 1922 1408">Build an infrastructure for a system that employs model calculations incorporating observational “Big Data” to accurately predict localized torrential rain, tornados, typhoons etc. and that also monitors and projects impacts of environmental changes due to human activity, in order to contribute to environmental policy, disaster prevention and health measures.</p> |

9 social and scientific priority issues (2/3)

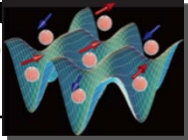

| Category | Priority issues |
|--|--|
| <p>Energy problem</p>  | <p>⑤ Development of new fundamental technologies for high-efficiency energy creation, conversion/storage and use</p> <p>Perform full-system simulations at the molecular level for complicated real-world complex systems to explain the entire process of high-efficiency energy creation, conversion/storage and use in coordination with experimentation, in order to develop new fundamental technologies to resolve energy-related problem.</p> <hr/> <p>⑥ Accelerated Development of Innovative Clean Energy Systems</p> <p>Subject the complex physical phenomena that form the core of energy systems to first-principles analysis to predict their occurrence and explicate their comprehensive behavior for accelerating the practical application of innovative and clean energy systems that have ultra-high efficiency and low environmental impact.</p> |
| <p>Enhancement of industrial competitiveness</p>  | <p>⑦ Creation of new functional devices and high-performance materials to support next-generation industries</p> <p>Accelerate the development of electronics technologies, structural materials, functional chemical products etc. that have great international competitiveness, through coordination with large-scale massively parallel computing and the analysis of “Big Data” and data from measurement and experimentation, in order to create devices and materials to support next-generation industries.</p> <hr/> <p>⑧ Development of Innovative Design and Production Processes that Lead the Way for the Manufacturing Industry in the Near Future</p> <p>Conduct research and development for innovative design techniques, where the product concept is quantitatively assessed at the initial stage and optimization is performed. By implementing innovative manufacturing processes that reduce costs and by performing ultra-high speed integration simulations, both of which form the core of the research and development efforts, high value-added product development can be achieved.</p> |

9 social and scientific priority issues (3/3)

| Category | Priority issues |
|--|---|
| <p data-bbox="126 429 410 564">Development of basic science</p>  | <p data-bbox="468 429 1725 464">⑨ Elucidation of the fundamental laws and evolution of the universe</p> <p data-bbox="468 479 1929 639">Realize precise calculations of the phenomena over wide range of scales from elementary particles to the universe. Combining with the data from large-scale experiments and observations, they play crucial roles to address the remaining problems in the history of the universe that extend across particle, nuclear and astro physics.</p> |

4 exploratory Challenges

Research organizations for 4 challenges have not been selected

| Exploratory challenges | |
|---|---|
| <p>⑩ Frontiers of basic science: challenge to the limits</p> <p>At the frontiers of basic science where researchers pursue the limits and extreme conditions, efforts will be made to resolve difficult problem and challenges that have been answered neither by experiments, observations nor even by individual achievements of computational science using the Post-K Computer. Co-creation of new science and interdisciplinary collaboration using the "Post-K computer" is called for.</p> |  |
| <p>⑪ Construction of models for interaction among multiple socioeconomic</p> <p>To give policy and measures the agility to deal with various problems produced in our complex and rapidly changing modern society, research and development of systems for determination, analysis and prediction will be conducted, taking into account the effect of the mutual influence of individual elements of social activities such as transport and the economy.</p> | |
| <p>⑫ Elucidation of the birth of exoplanets (Second Earths) and the environmental variations of planets in the solar system</p> <p>Through multidisciplinary approach under the collaboration of computational sciences (in the fields of astrophysics, geophysical/planetary science, meteorology, and molecular science), we achieve large-scale calculations, which can be directly confronted to observations and experiments, and explore the origin of terrestrial planets, the environment of the solar system, and interstellar molecular science.</p> | |
| <p>⑬ Elucidation of how neural networks realize thinking and its application to artificial intelligence</p> <p>By integrating big data produced by innovative brain science technologies, large-scale multi-level models of the brain are constructed and through large-scale simulations using the "Post-K computer," the brain's mechanism of thinking by neural networks is reproduced and applied to artificial intelligence.</p> |  |

Selected target apps from each area for “codesign”

| | Target Application | |
|---|--------------------|---|
| | Program | Brief description |
| ① | GENESIS | MD for proteins |
| ② | Genomon | Genome processing (Genome alignment) |
| ③ | GAMERA | Earthquake simulator (FEM in unstructured & structured grid) |
| ④ | NICAM+LETK | Weather prediction system using Big data (structured grid stencil & ensemble Kalman filter) |
| ⑤ | NTChem | molecular electronic (structure calculation) |
| ⑥ | FFB | Large Eddy Simulation (unstructured grid) |
| ⑦ | RSDFT | an ab-initio program (density functional theory) |
| ⑧ | Adventure | Computational Mechanics System for Large Scale Analysis and Design (unstructured grid) |
| ⑨ | CCS-QCD | Lattice QCD simulation (structured grid Monte Carlo) |

A projection: Pre-exa, exa, post-exa

| | Pre-exa | exascale | Post-exa |
|---------------------------------|-----------------------------------|---------------|--------------|
| System performance (PF) | 50~500 | 500~5,000 | 1,000~10,000 |
| node performance (TF) | 1~10 | 5~50 | 10~100 |
| #number of node (K) | 5~500 | 10~1,000 | 10~1,000 |
| Performance/ power(GF/W) | 2~20 | 20~200? | 400? |
| Memory bandwidth and technology | 0.5~1TB/s (HBM) 150GB/s (DDR4) | 1~4TB/s (HBM) | ??? |

- Node performance must increase! Because the system scale is limited by space and power.
- Memory performance will be limited. So, the cap between B/F will be getting worse.
- Improvement of performance/power will be difficult and limited.

Challenges of Programming Languages/models for exascale computing

- **Scalability, Locality and scalable Algorithms in system-wide**
- **Strong Scaling in node**
- **Workflow and Fault-Resilience**
- **(Power-aware)**

“MPI+X” for exascale?

- **X is OpenMP!**
- **“MPI+Open” is now a standard programming for high-end systems.**
 - I’d like to celebrate that OpenMP became “standard” in HPC programming
- **Questions:**
 - “MPI+OpenMP” is still a main programming model for exa-scale?

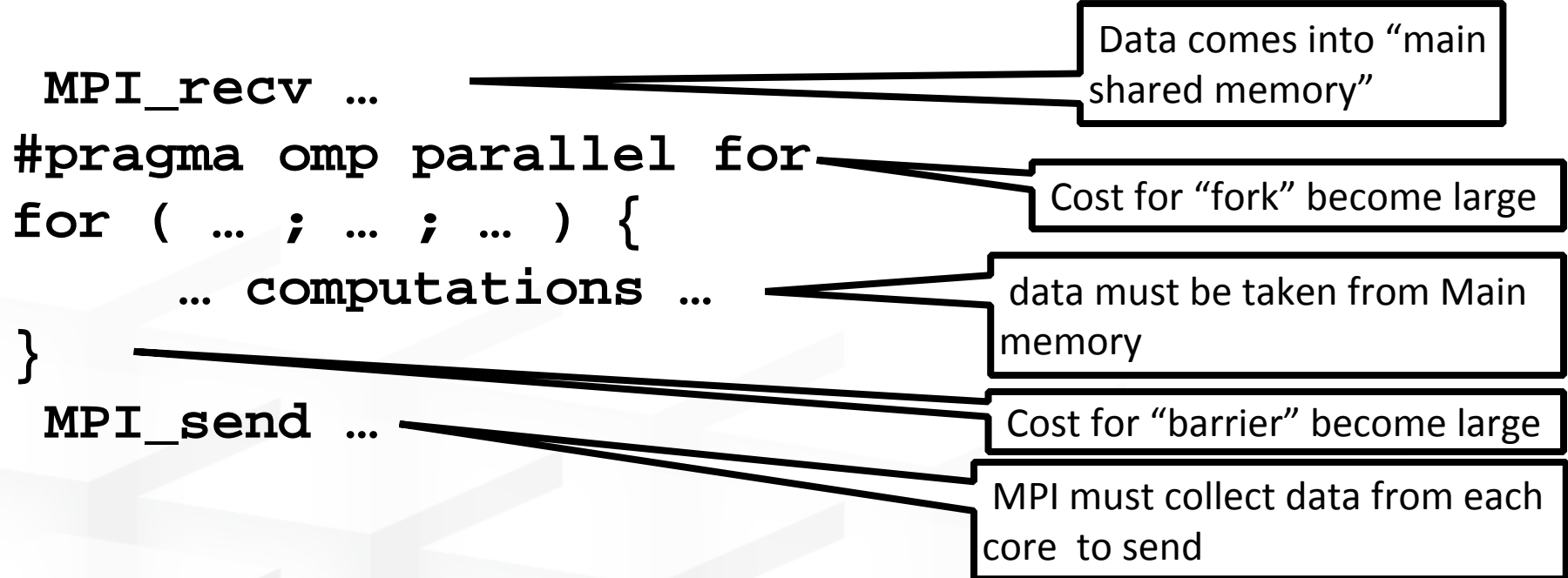
Question

- What happens when executing code using all cores in manycore processors like this ?

```

MPI_recv ...
#pragma omp parallel for
for ( ... ; ... ; ... ) {
    ... computations ...
}
MPI_send ...

```



The diagram illustrates the performance challenges of using MPI and OpenMP on manycore processors. Callouts point to specific parts of the code:

- Data comes into "main shared memory"**: Points to the `MPI_recv` call.
- Cost for "fork" become large**: Points to the `#pragma omp parallel for` directive.
- data must be taken from Main memory**: Points to the `... computations ...` block.
- Cost for "barrier" become large**: Points to the closing brace of the `for` loop.
- MPI must collect data from each core to send**: Points to the `MPI_send` call.

- What are solutions?
 - MPI+OpenMP runs on divided small "NUMA domains" rather than all cores?

Barrier in Xeon Phi

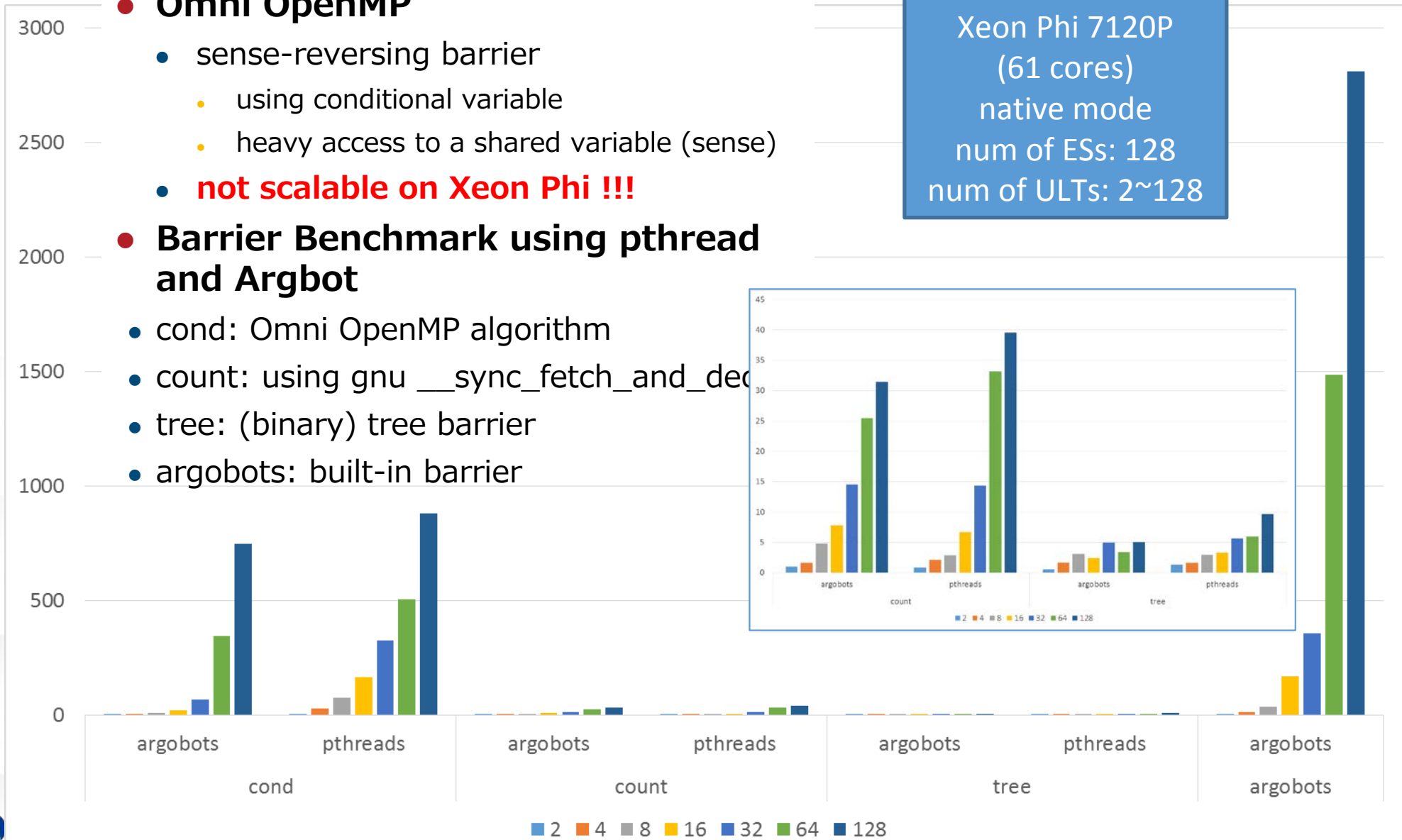
- **Omni OpenMP**

- sense-reversing barrier
 - using conditional variable
 - heavy access to a shared variable (sense)
- **not scalable on Xeon Phi !!!**

- **Barrier Benchmark using pthread and Argbot**

- cond: Omni OpenMP algorithm
- count: using gnu `__sync_fetch_and_dec`
- tree: (binary) tree barrier
- argobots: built-in barrier

Xeon Phi 7120P
(61 cores)
native mode
num of ESs: 128
num of ULTs: 2~128



Multitasking model

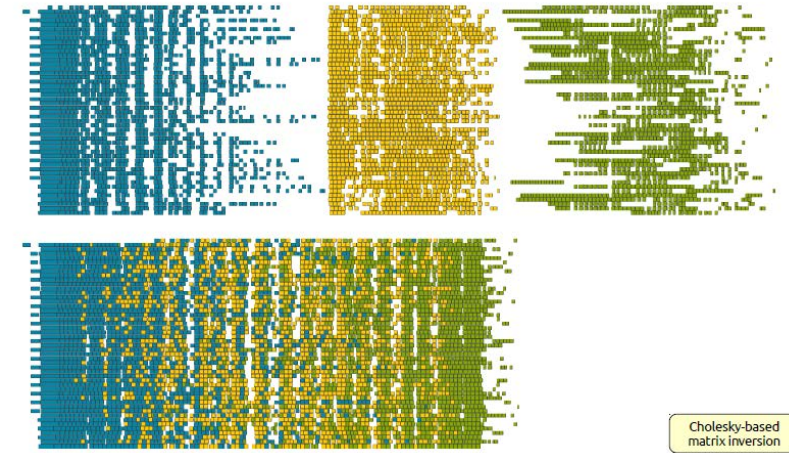
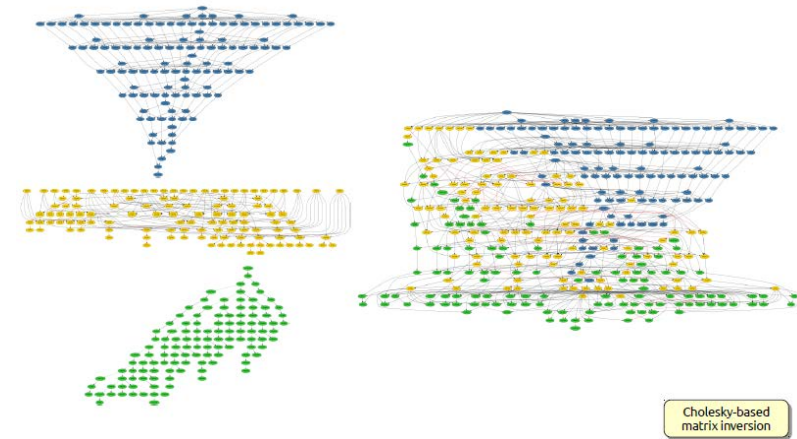
- **Multitasking/Multithreaded execution: many “tasks” are generated/executed and communicates with each others by data dependency.**

- OpenMP task directive, OmpSS, PLASMA/QUARK, StarPU, ..
- Thread-to-thread synchronization /communications rather than barrier

- **Advantages**

- Remove barrier which is costly in large scale manycore system.
- Overlap of computations and computation is done naturally.
- New communication fabric such as Intel OPA (OmniPath Architecture) may support core-to-core communication that allows data to come to core directly.

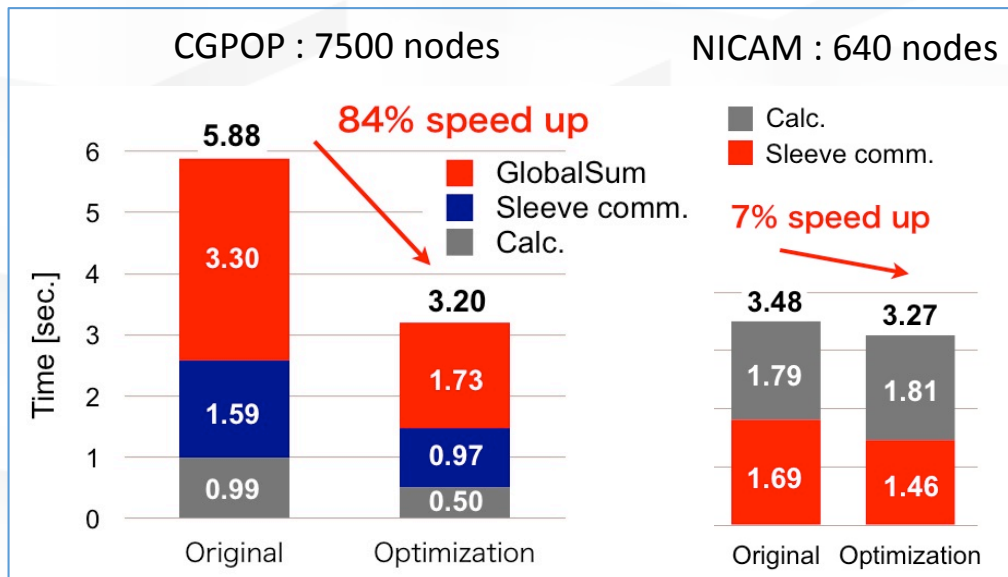
- **New algorithms must be designed to use multitasking**



From PLASMA/QUARK slides by ICL, U. Tennessee

PGAS (Partitioned Global Address Space) models

- Light-weight one-sided communication and low overhead synchronization semantics.
- PAGES concept is adopted in Coarray Fortran, UPC, X10, XMP.
 - XMP adopts notion Coarray not only Fortran but also “C”, as “local view” as well as “global view” of data parallelism.
- Advantages and comments
 - Easy and intuitive to describe, not only one side-comm, but also strided comm.
 - Recent networks such as Cray and Fujitsu Tofu support remote DMA operation which strongly support efficient one-sided communication.
 - Other collective communication library (can be MPI) are required.



Case study of XMP on K computer
CGPOP, NICAM: Climate code

5-7 % speed up is obtained by replacing
MPI with Coarray

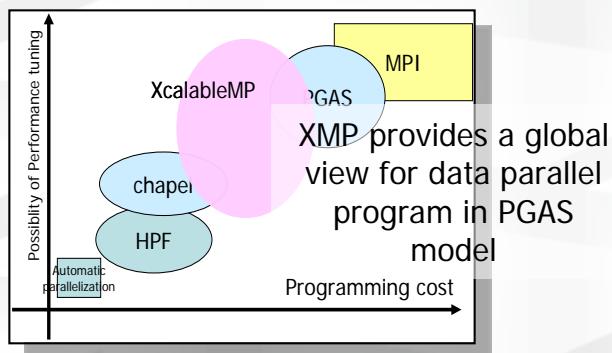
• What's XcalableMP (XMP for short)?

- 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. 2014)

- 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 0.9 (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.

• HPC class 2 Winner 2013. 2014



■ 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)
```

data distribution

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

add to the serial code : incremental parallelization

```
#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];
  }
}
```

work sharing and data synchronization

XcalableMP as evolutionary approach

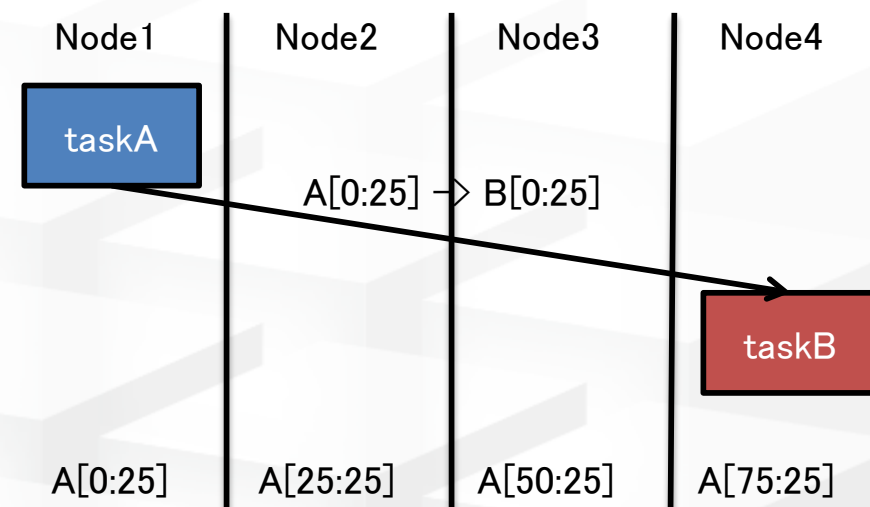
- **We focus on migration from existing codes.**
 - Directive-based approach to enable parallelization by adding directives/pragma.
 - Also, should be from MPI code. Coarray may replace MPI.
- **Learn from the past**
 - Global View for data-parallel apps. Japanese community had experience of HPF for Global-view model.
- **Specification designed by community**
 - Spec WG is organized under the PC Cluster Consortium, Japan
- **Design based on PGAS model and Coarray (From CAF)**
 - PGAS is an emerging programming model for exascale!
- **Used as a research vehicle for programming lang/model research.**
 - XMP 2.0 for multitasking.
 - Extension to accelerator (XACC)

XcalableMP 2.0

- **Specification v 1.2:**
 - Support for Multicore: hybrid XMP and OpenMP is defined.
 - Dynamic allocation of distributed array
- **A set of spec in version 1 is now “converged”. New functions should be discussed for version 2.**
- **Main topics for XcalableMP 2.0: Support for manycore**
 - Multitasking with integrations of PGAS model
 - Synchronization models for dataflow/multitasking executions
 - Proposal: tasklet directive
 - Similar to OpenMP task directive
 - Including inter-node communication on PGAS

```

int A[100], B[25];
#pragma xmp nodes P()
#pragma xmp template T(0:99)
#pragma xmp distribute T(block) onto P
#pragma xmp align A[i] with T(i)
/ ... /
#pragma xmp tasklet out(A[0:25], T(75:99))
taskA();
#pragma xmp tasklet in(B, T(0:24)) out(A[75:25])
taskB();
#pragma xmp taskletwait
  
```



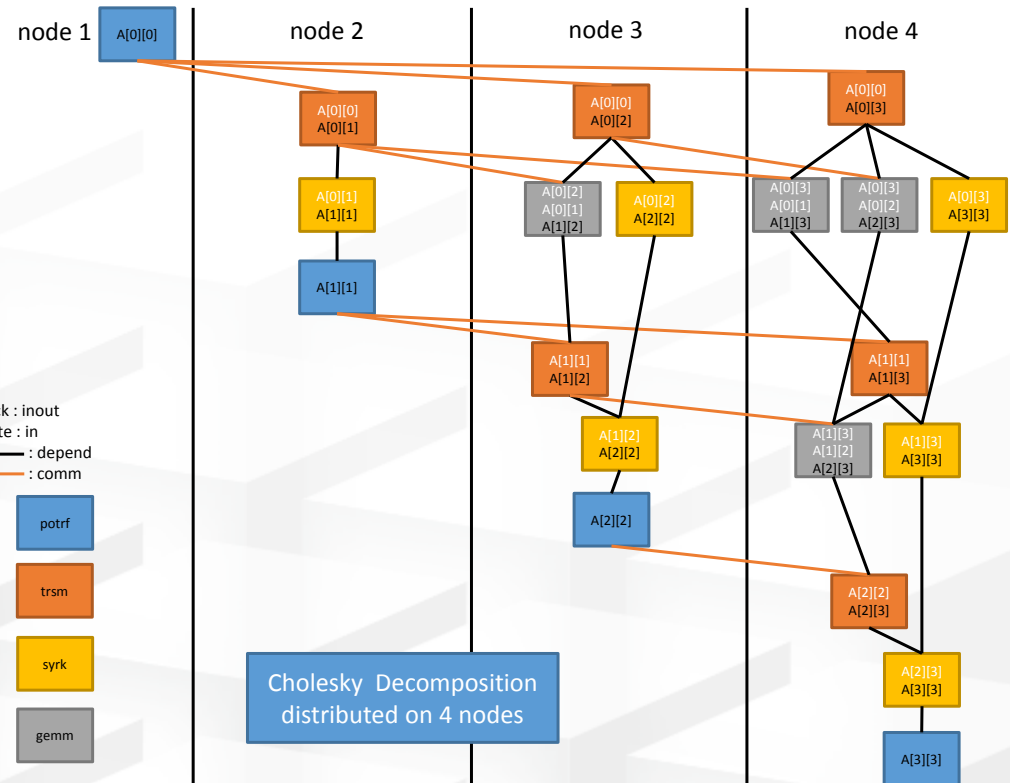
Proposal of Tasklet directive

- The detail spec of the directive is under discussion in spec-WG
- Currently, we are working on prototype implementations and preliminary evaluations
- Example: Cholesky Decomposition

```
double A[nt][nt][ts*ts], B[ts*ts], C[nt][ts*ts];
#pragma xmp node P(*)
#pragma xmp template T(0:nt-1)
#pragma xmp distribute T(cyclic) onto P
#pragma xmp align A[*][i][*] with T(i)
```

```
for (int k = 0; k < nt; k++) {
  #pragma xmp tasklet inout(A[k][k], T(k+1:nt-1))
  omp_potrf (A[k][k], ts, ts);

  for (int i = k + 1; i < nt; i++) {
    #pragma xmp tasklet in(B, T(k)) inout(A[k][i], T(i+1:nt-1))
    omp_trsm (B, A[k][i], ts, ts);
  }
  for (int i = k + 1; i < nt; i++) {
    for (int j = k + 1; j < i; j++) {
      #pragma xmp tasklet in(A[k][i]) in(C[j], T(j)) inout(A[j][i])
      omp_gemm (A[k][i], C[j], A[j][i], ts, ts);
    }
    #pragma xmp tasklet in(A[k][i]) inout(A[i][i])
    omp_syrk (A[k][i], A[i][i], ts, ts);
  }
}
#pragma xmp taskletwait
```



Strong Scaling in node

- **Two approaches:**
 - SIMD for core in manycore processors
 - Accelerator such as GPUs
- **Programming for SIMD**
 - Vectorization by directives or automatic compiler technology
 - Limited bandwidth of memory and NoC
 - Complex memory system: Fast-memory (MD-DRAM, HBM, HMC) and DDR , VMRAM
- **Programming for GPUs**
 - Parallelization by OpenACC/OpenMP 4.0. Still immature but getting matured soon
 - Fast memory (HMB) and fast link (NV-Link): similar problem of complex memory system in manycore.
 - Programming model to be shared by manycore and accelerator for high productivity.

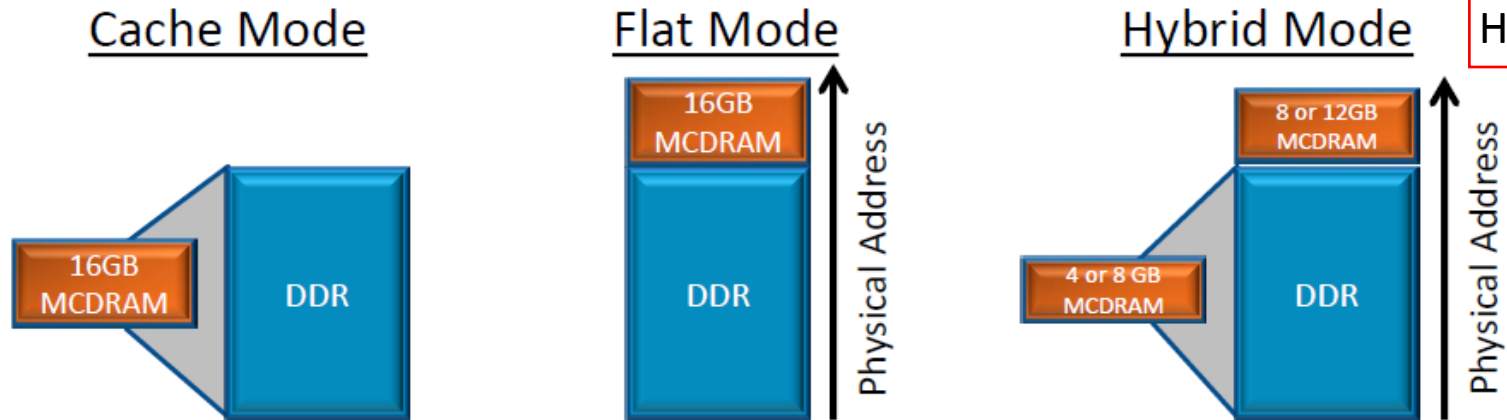
How to use MC-DRAM in KNL?

- **New Xeon Phi (KNL) has fast memory called MC-DRAM.**
 - KNL performance: < 5 TF (Theoretical Peak)
 - DDR4: 100~200 GB/s, MC-DRAM: 0.5 TB/s
 - How to use?

Memory Modes

Three Modes. Selected at boot

From Intel Slide presented at HotChips 2015



- SW-Transparent, Mem-side cache
- Direct mapped. 64B lines.
- Tags part of line
- Covers whole DDR range

- MCDRAM as regular memory
- SW-Managed
- Same address space

- Part cache, Part memory
- 25% or 50% cache
- Benefits of both

XcalableACC(ACC) = XcalableMP+OpenACC

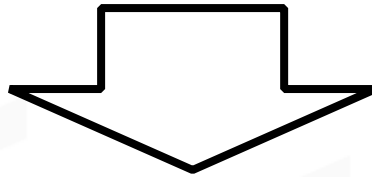
● Extension of XcalableMP for GPU

- A project of U. Tsukuba led by Prof. Taiuske Boku
- “vertical” integration of XcalableMP and OpenACC
 - Data distribution for both host and GPU by XcalableMP
 - Offloading computations in a set of nodes by OpenACC
- Proposed as unified parallel programming model for many-core architecture & accelerator
 - GPU, Intel Xeon Phi
 - OpenACC supports many architectures

Source Code Example: NPB CG

```
#pragma xmp nodes p(NUM_COLS, NUM_ROWS)
#pragma xmp template t(0:NA-1,0:NA-1)
#pragma xmp distribute t(block, block) onto p
#pragma xmp align w[i] with t(*,i)
#pragma xmp align q[i] with t(i,*)
double a[NZ];
int rowstr[NA+1], colidx[NZ];
...
#pragma acc data copy(p,q,r,w,rowstr[0:NA+1]¥
                    , a[0:NZ], colidx[0:NZ])
{
    ...
    #pragma xmp loop on t(*,j)
    #pragma acc parallel loop gang
    for(j=0; j < NA; j++){
        double sum = 0.0;
        #pragma acc loop vector reduction(+:sum)
        for (k = rowstr[j]; k < rowstr[j+1]; k++)
            sum = sum + a[k]*p[colidx[k]];
        w[j] = sum;
    }
    #pragma xmp reduction(+:w) on p(:,*) acc
    #pragma xmp gmove acc
    q[:] = w[:];
    ...
} //end acc data
```

- Petascale system was targeting some of “capability” computing.
- In exascale system, it become important to execute huge number of medium-grain jobs for parameter-search type applications.

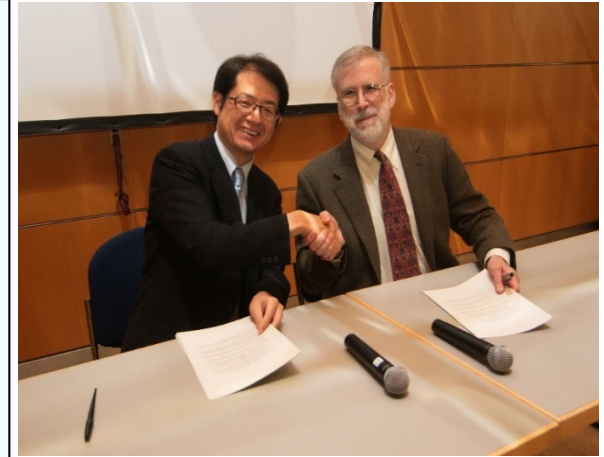


Workflow to control and collect/process data is important, also for “big-data” apps.

International Collaboration between DOE and MEXT

PROJECT ARRANGEMENT
UNDER THE IMPLEMENTING ARRANGEMENT
BETWEEN
THE MINISTRY OF EDUCATION, CULTURE, SPORTS, SCIENCE AND TECHNOLOGY OF JAPAN
AND
THE DEPARTMENT OF ENERGY OF THE UNITED STATES OF AMERICA
CONCERNING COOPERATION IN RESEARCH AND DEVELOPMENT IN ENERGY AND RELATED
FIELDS

CONCERNING COMPUTER SCIENCE AND SOFTWARE RELATED TO CURRENT AND FUTURE
HIGH PERFORMANCE COMPUTING FOR OPEN SCIENTIFIC RESEARCH



Yoshio Kawaguchi (MEXT, Japan)
and William Harrod (DOE, USA)

Purpose: Work together where it is mutually beneficial to expand the HPC ecosystem and improve system capability

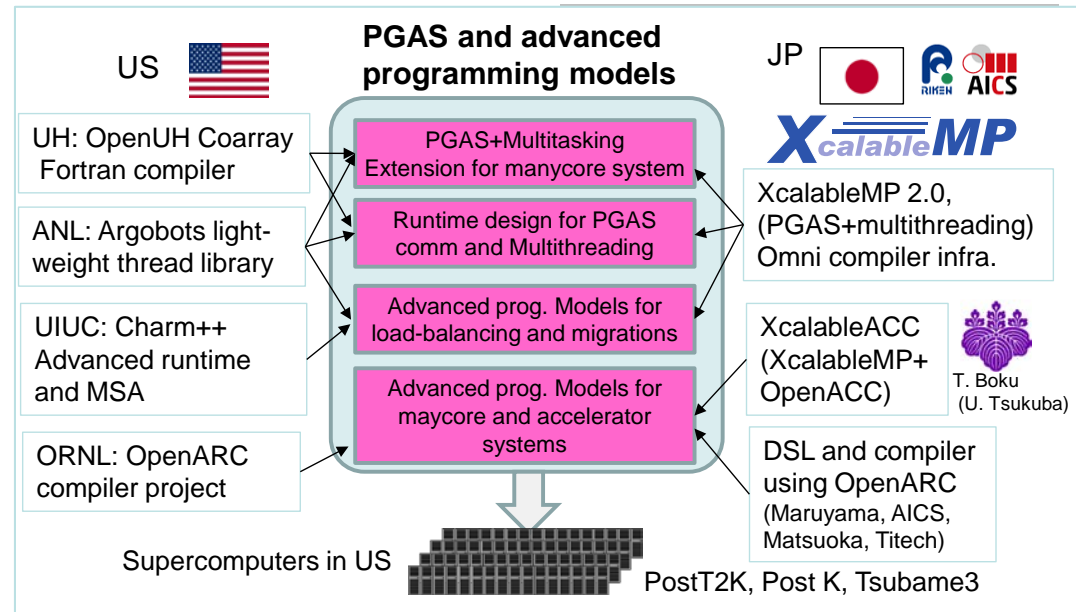
- Each country will develop their own path for next generation platforms
- Countries will collaborate where it is mutually beneficial
- Joint Activities
 - Pre-standardization interface coordination
 - Collection and publication of open data
 - Collaborative development of open source software
 - Evaluation and analysis of benchmarks and architectures
 - Standardization of mature technologies

Technical Areas of Cooperation

- Kernel System Programming Interface
- Low-level Communication Layer
- Task and Thread Management to Support Massive Concurrency
- Power Management and Optimization
- Data Staging and Input/Output (I/O) Bottlenecks
- File System and I/O Management
- Improving System and Application Resilience to Chip Failures and other Faults
- Mini-Applications for Exascale Component-Based Performance Modelling

PGAS and Advanced programming models for exascale systems

- Coordinators
 - US: P. Beckman (ANL), JP: M. Sato (RIKEN)
- Leaders
 - US: L. Kale (UIUC), B Chapman (U Huston), J. Vetter (ORNL), P. Balaji (ANL)
 - JP: M Sato (RIKEN)
- Collaborators
 - S. Seo (ANL), D Bernholdt (ORNL), D. Eachempati(UH)
 - H. Murai (RIKEN), J. Lee (RIKEN), N. Maruyama (RIKEN), T. Boku (U. Tsukuba)
- Collaboration topics
 - Extension of PGAS (Partitioned Global Address Space) model with language constructs of multithreading for manycore-based exascale systems
 - Runtime design for PGAS communication and multithreading
 - Advanced programming models to support both manycore-based and accelerator-based exascale system for high productivity.
 - Advanced programming models for dynamic load-balancing and migration in exascale systems
- How to collaborate
 - Twice meetings per year
 - Student / young researchers exchange, sharing codes
 - Funding:
 - US: ARGO, X-stack(XPRESS), X-stack(Vancouver, ARES)
 - JP: FLAGSHIP 2020, PP-CREST (JP)



- Deliverables
 - Concepts for PGAS and multithreading integration for manycore-based exascale systems.
 - Concepts for advanced programming model to be shared by both manycore and accelerators-based systems.
 - Pre-standardization of Application Programming Interface for multithreading (based on Argobots) and PGAS
- Recent activities and plans
 - AICS teams visited UH, UIUC and ANL for discussions.
 - Start using Argobots for Omni OpenMP compiler and produced preliminary results on intel Xeon Phi.
 - AICS invited Post-doc from UH for collaborations on PGAS
 - ORNL visited AICS to have a meeting for the collaboration
 - JP (AICS, Tsukuba) will send Post-doc and students to ANL and UH, ORNL
 - JP and ORNL will have a meeting in JP or US how to collaborate.

Concluding remarks

- **FLAGSHIP 2020 project**

- To develop the next Japanese flagship computer system, post-K
- The basic architecture design and target application performances will be decided by 2015 3Q
- “Co-design” effort will be continued (application design for architecture)

- **XcalableMP is our research vehicle for programming language/model research.**

- XMP 2.0 for multitasking for many-core-based system.
- Extension to accelerator (XACC)

- **Schedule**

CY

| 2014 | | | | 2015 | | | | 2016 | | | | 2017 | | | | 2018 | | | | 2019 | | | | 2020 | | | |
|--------------|----|----|----|------|----|----|----|---------------------------|----|----|----|------|----|----|----|---|----|----|----|------|----|----|----|-----------|----|----|----|
| Q1 | Q2 | Q3 | Q4 | Q1 | Q2 | Q3 | Q4 | Q1 | Q2 | Q3 | Q4 | Q1 | Q2 | Q3 | Q4 | Q1 | Q2 | Q3 | Q4 | Q1 | Q2 | Q3 | Q4 | Q1 | Q2 | Q3 | Q4 |
| Basic Design | | | | | | | | Design and Implementation | | | | | | | | Manufacturing, Installation, and Tuning | | | | | | | | Operation | | | |