Post-K Computer Project Overview

contact : postk-info@riken.jp

RIKEN was selected to carry out development of the post-K computer—designed to be the successor of the K computer—under the Ministry of Education, Culture, Sports, Science, and Technology' s FLAGSHIP 2020 Project. We are currently engaged in the development work with the aim to launch the new computer around 2020.

Outline of the development of the post-K computer

● Top priority on problem-solving research

During development, highest priority will be given to creating a system capable of contributing to the solution of various scientific and societal issues. For this, the hardware and software will be developed in a coordinated way (Co-design), with the aim to make it usable in a variety of fields.

● World-leading performance

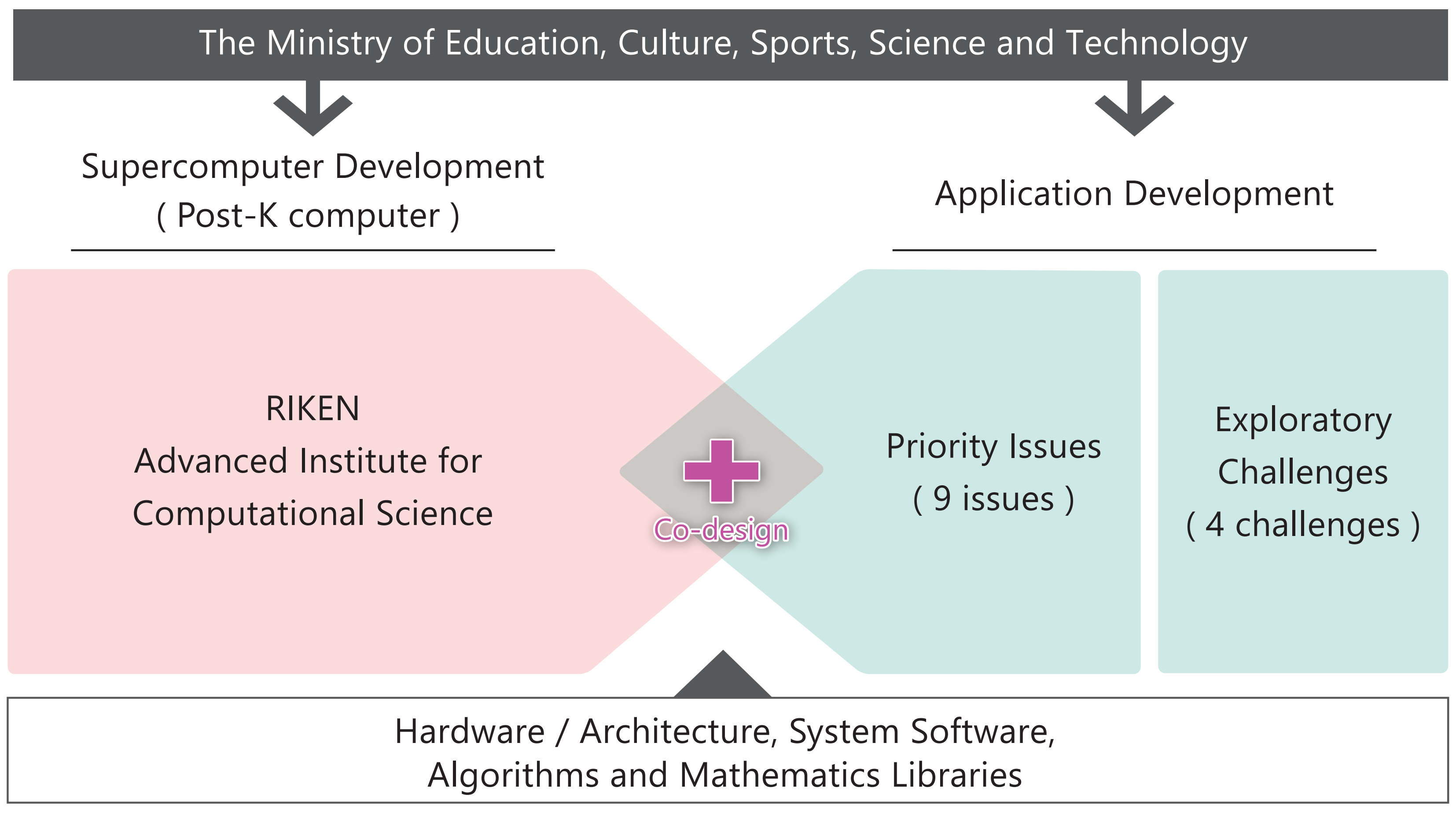
Create the most advanced general-use system in the world.

● Improve performance through international cooperation

While leveraging Japan' s strengths, cooperate internationally to achieve world-leading technologies of the highest quality and become the international standard.

● Continue the legacy of the K computer

Make the fullest use of the technologies, human resources, and applications of the K computer project for developing the post-K system.



The Ministry of Education, Culture, Sports, Science and Technology

Supercomputer Development (Post-K computer)

Application Development

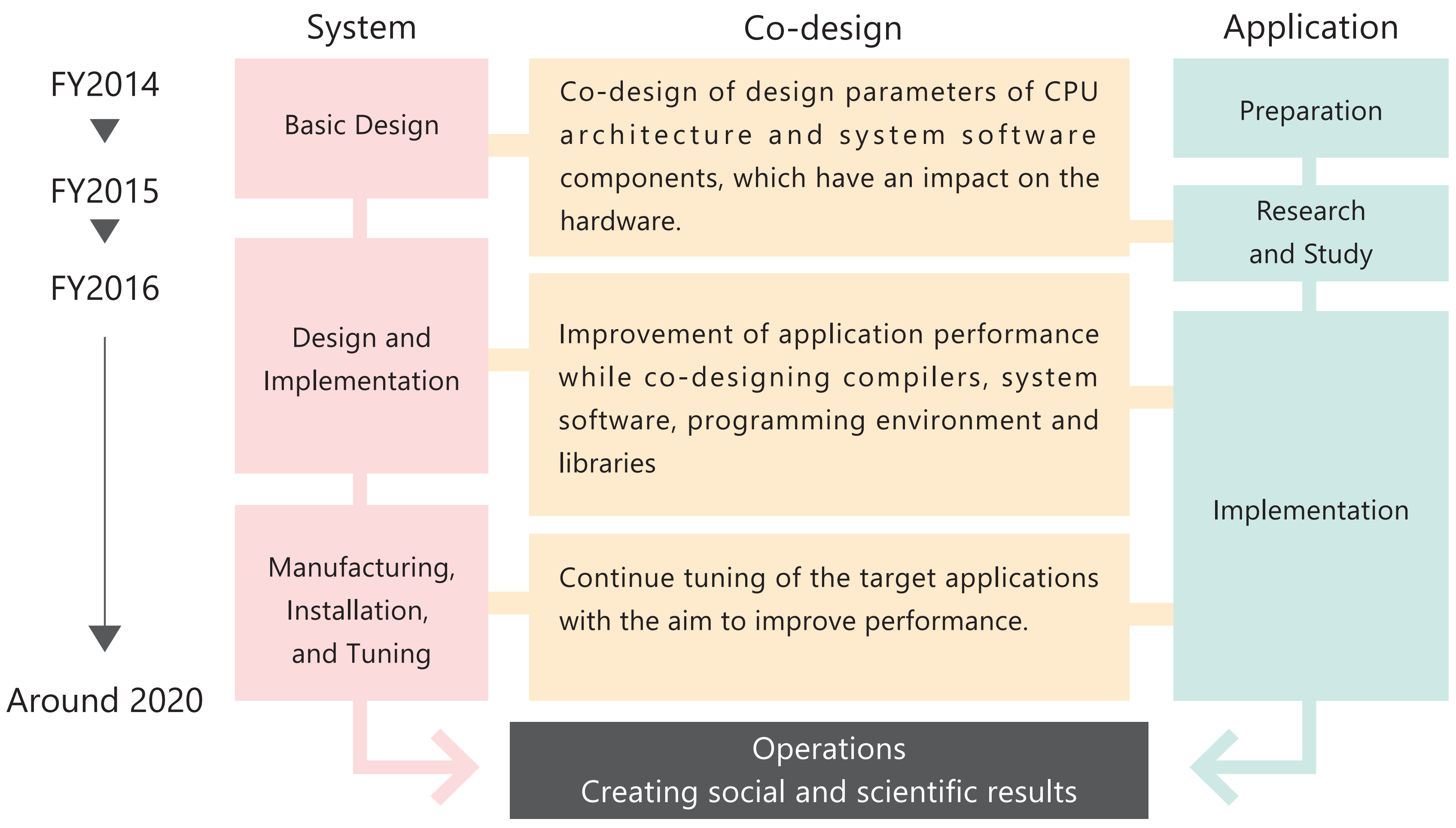
RIKEN Advanced Institute for Computational Science

Co-design

Priority Issues (9 issues)

Exploratory Challenges (4 challenges)

Hardware / Architecture, System Software, Algorithms and Mathematics Libraries



System

Co-design

Application

FY2014

FY2015

FY2016

Around 2020

Basic Design

Design and Implementation

Manufacturing, Installation, and Tuning

Co-design of design parameters of CPU architecture and system software components, which have an impact on the hardware.

Improvement of application performance while co-designing compilers, system software, programming environment and libraries

Continue tuning of the target applications with the aim to improve performance.

Preparation

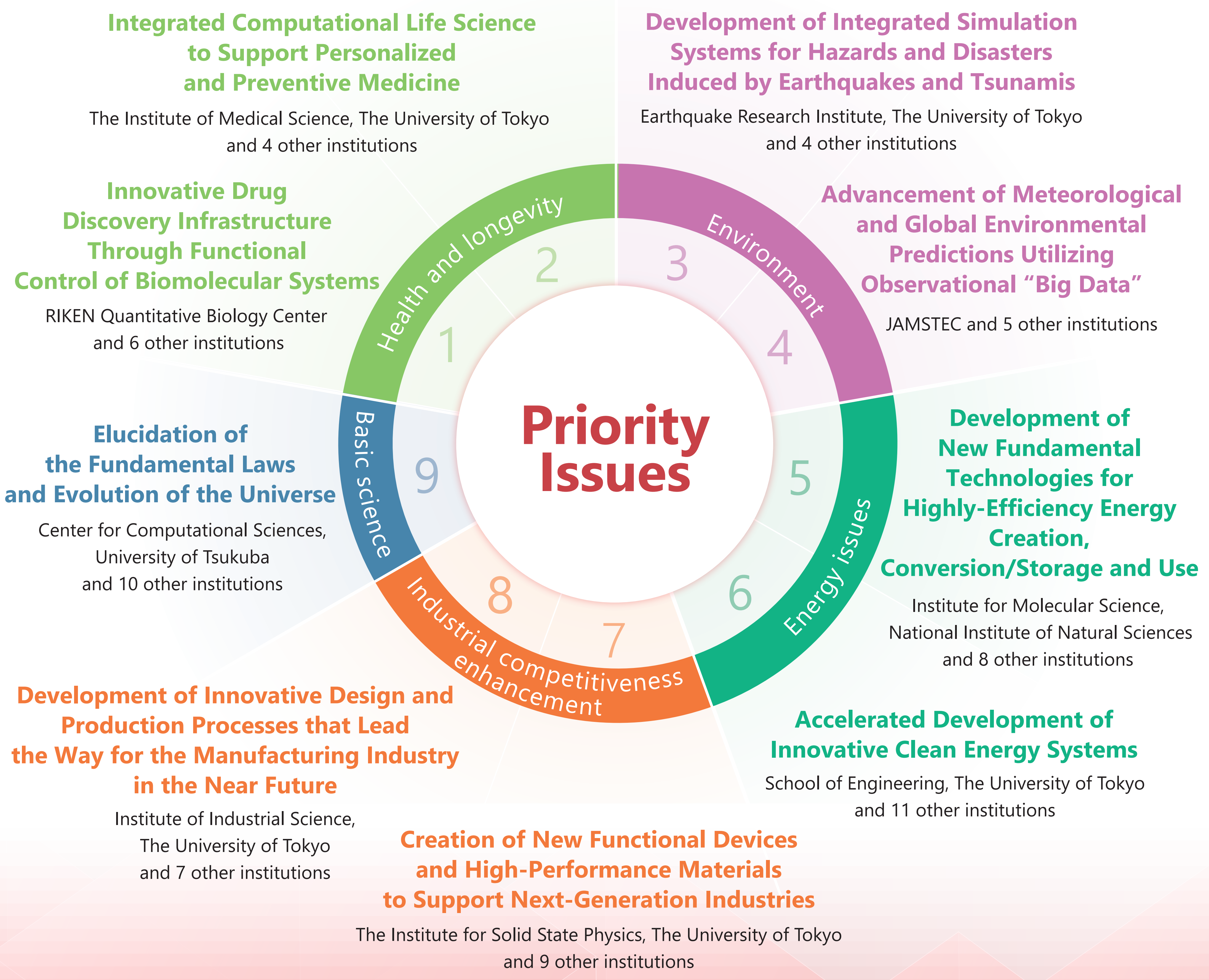
Research and Study

Implementation

Operations

Creating social and scientific results

Research Subjects of the post-K computer



Priority Issues

Health and longevity

Environment

Energy issues

Industrial competitiveness enhancement

Basic science

1

Innovative Drug Discovery Infrastructure Through Functional Control of Biomolecular Systems

RIKEN Quantitative Biology Center and 6 other institutions

2

Integrated Computational Life Science to Support Personalized and Preventive Medicine

The Institute of Medical Science, The University of Tokyo and 4 other institutions

3

Development of Integrated Simulation Systems for Hazards and Disasters Induced by Earthquakes and Tsunamis

Earthquake Research Institute, The University of Tokyo and 4 other institutions

4

Advancement of Meteorological and Global Environmental Predictions Utilizing Observational "Big Data"

JAMSTEC and 5 other institutions

5

Development of New Fundamental Technologies for Highly-Efficiency Energy Creation, Conversion/Storage and Use

Institute for Molecular Science, National Institute of Natural Sciences and 8 other institutions

6

Accelerated Development of Innovative Clean Energy Systems

School of Engineering, The University of Tokyo and 11 other institutions

7

Creation of New Functional Devices and High-Performance Materials to Support Next-Generation Industries

The Institute for Solid State Physics, The University of Tokyo and 9 other institutions

8

Development of Innovative Design and Production Processes that Lead the Way for the Manufacturing Industry in the Near Future

Institute of Industrial Science, The University of Tokyo and 7 other institutions

9

Elucidation of the Fundamental Laws and Evolution of the Universe

Center for Computational Sciences, University of Tsukuba and 10 other institutions

Priority Issues

Japanese research institutes and universities in charge of the priority issues started their work in 2015 and are actively involved in creating a bright new future.

Exploratory challenges

In addition to the priority issues, four exploratory challenges to be tackled with post-K computer have been selected. Their actualization will be examined through feasibility study.

- **Frontiers of Basic Science: Challenging the Limits**
- **Construction of Models for Interaction Among Multiple Socioeconomic Phenomena**
- **Elucidation of the Birth of Exoplanets [Second Earth] and the Environmental Variations of Planets in the Solar System**
- **Elucidation of How Neural Networks Realize Thinking and Its Application to Artificial Intelligence**

Overview of Post-K Architecture

contact : postk-info@riken.jp

System Architecture and System Software

CPU node

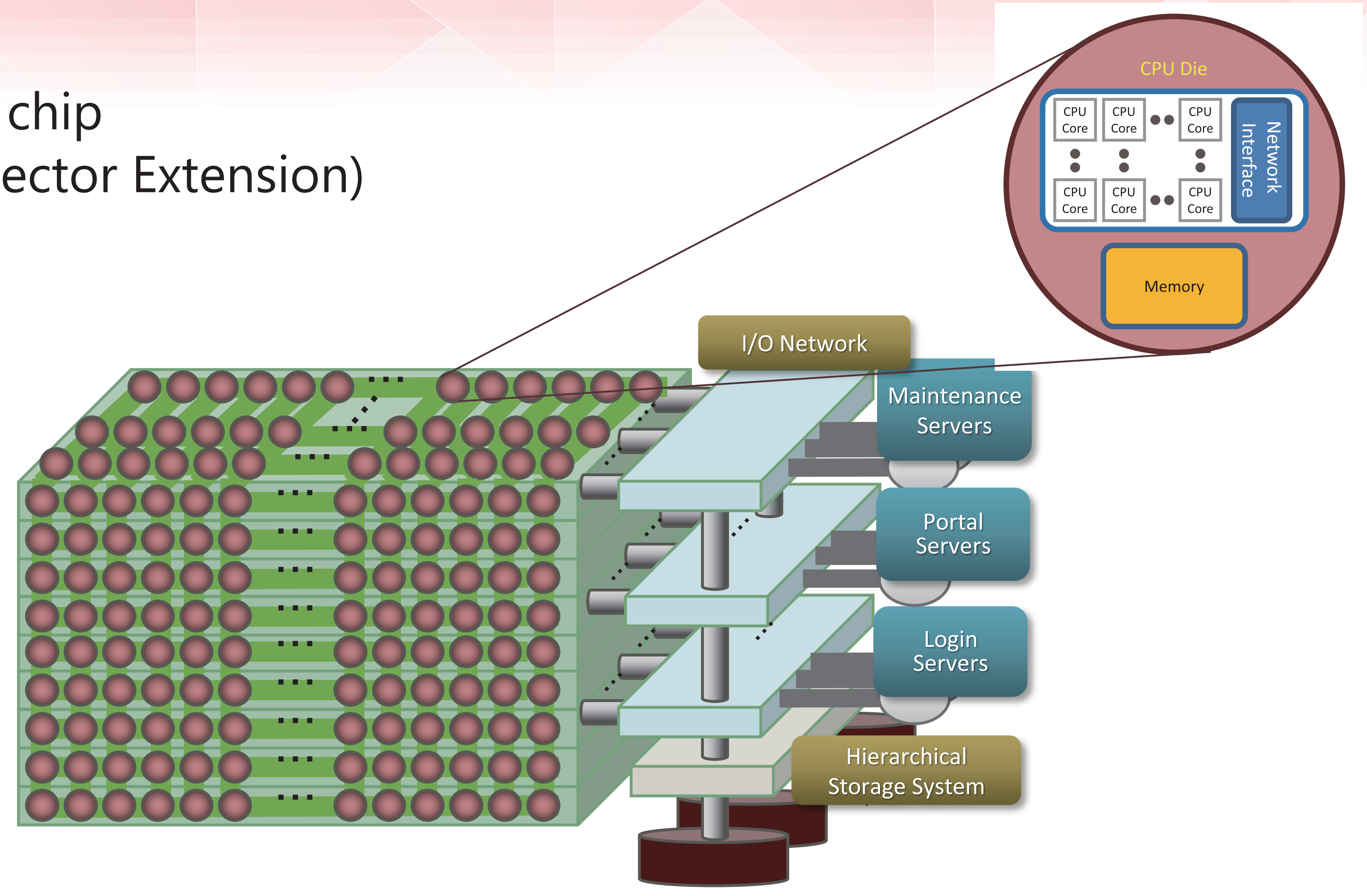
- Many-core processor with Interconnect interface integrated on chip
- A new Instruction Set Architecture: ARMv8 with SVE (Scalable Vector Extension)
- Power Knob features for saving power consumption

Interconnect

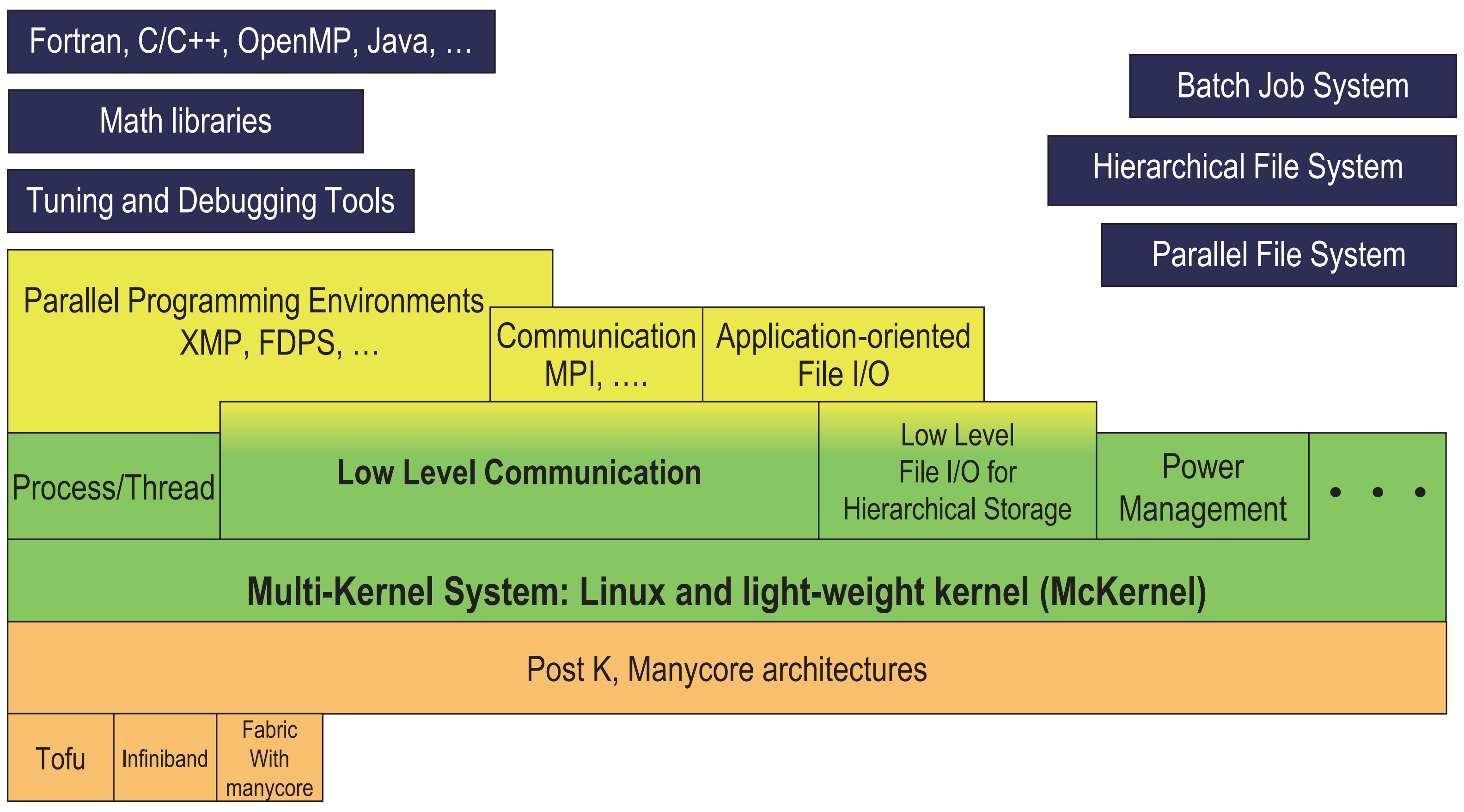
- A new version of TOFU (6D mesh/torus network)

3-level hierarchical storage system

- Silicon Disk
- Magnetic Disk
- Storage for archive



Architecture Overview



Fortran, C/C++, OpenMP, Java, …

Math libraries

Tuning and Debugging Tools

Batch Job System

Hierarchical File System

Parallel File System

Parallel Programming Environments XMP, FDPS, …

Communication MPI, ….

Application-oriented File I/O

Process/Thread

Low Level Communication

Low Level File I/O for Hierarchical Storage

Power Management

• • •

Multi-Kernel System: Linux and light-weight kernel (McKernel)

Post K, Manycore architectures

Tofu

Infiniband

Fabric With manycore

Overview of System software

System Software

- Multi-Kernel: Linux with Light-weight Kernel
- File I/O middleware for 3-level hierarchical storage system and application
- Application-oriented file I/O middleware
- MPI+OpenMP programming environment
- Highly productive programing language and libraries

ARM Scalable Vector Extension (SVE)

SVE is an extension developed specifically for vectorization of HPC scientific workloads. (FP64/FP32/FP16)

Features of SVE

- Scalable vector length (VL):   Increased parallelism while allowing implementation choice of VL
- VL agnostic (VLA) programming: Supports a programming paradigm of write-once, run-anywhere scalable vector code
- Gather-load & Scatter-store: Enables vectorization of complex data structures with non-linear access patterns
- Per-lane predication: Enables vectorization of complex, nested control code containing side effects and avoidance of loop heads and tails (particularly for VLA)
- Predicate-driven loop control and management: Reduces vectorization overhead relative to scalar code
- Scalarized intra-vector sub-loops: Supports vectorization of loops containing complex loop-carried dependencies

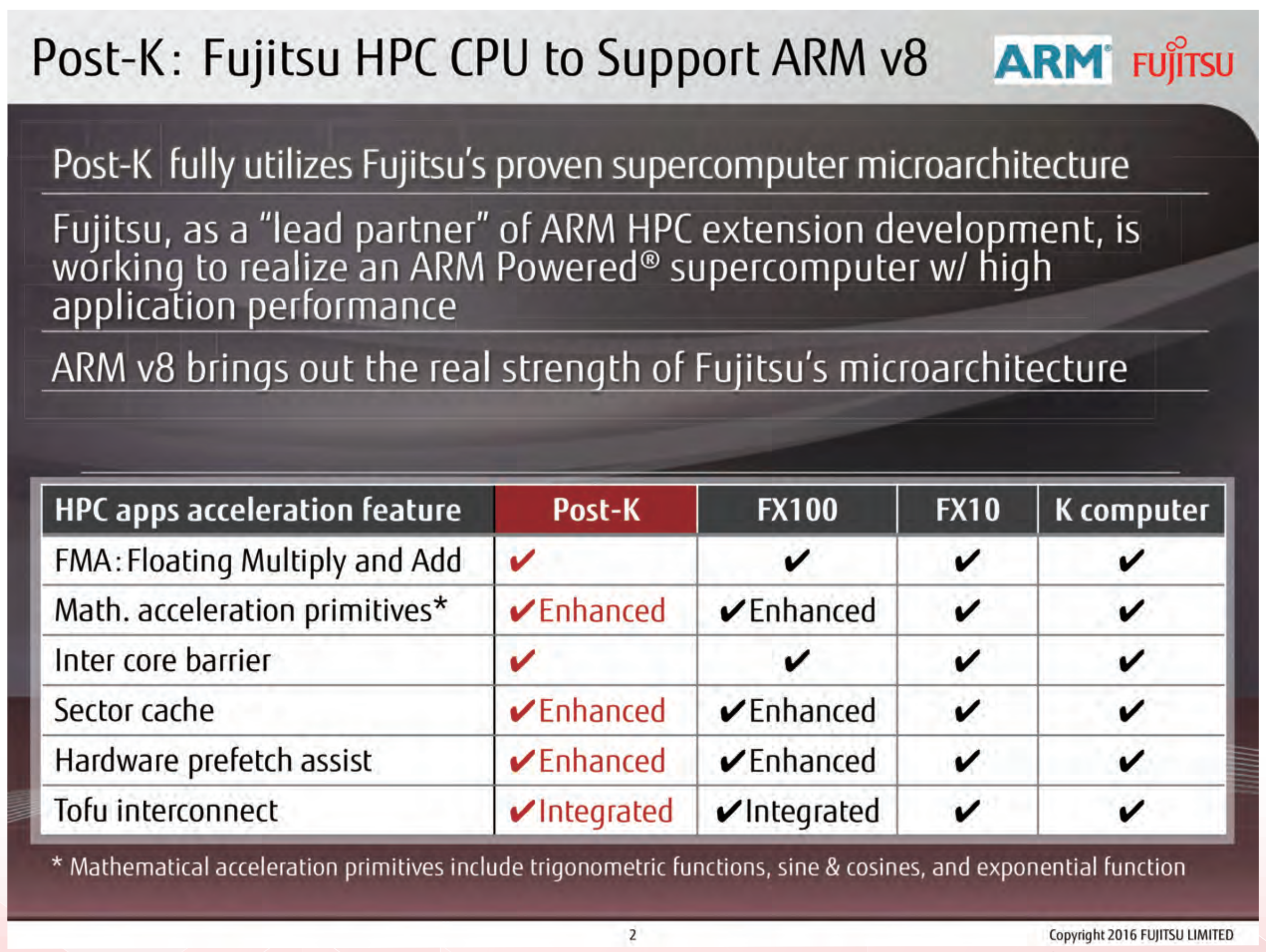
(Info from ARM:
https://community.arm.com/groups/processors/blog/2016/08/22/technology-update-the-scalable-vector-extension-sve-for-the-armv8-a-architecture)

The ARMv8-A SVE Supplement is now publically available.

(ARM(r) Architecture Reference Manual Supplement - The Scalable Vector Extension (SVE), for ARMv8-A:
https://developer.arm.com/products/architecture/a-profile/docs)

Research for ARM SVE

- Early assessment for ARM SVE spec.
- GEM5 processor simulator for ARM SVE
  — Development of GEM5 O3 Model for Post-K processor
- Evaluation and Testing of compilers for ARM SVE (with Kyoto Univ.)
  — ARM compiler (based on LLVM) (C and C++)   and Fujitsu compiler   (Fortran and C, C++)
- Compiler Research on SIMD-vector code-generation for ARM SVE



Post-K: Fujitsu HPC CPU to Support ARM v8

ARM FUJITSU

Post-K fully utilizes Fujitsu's proven supercomputer microarchitecture

Fujitsu, as a "lead partner" of ARM HPC extension development, is working to realize an ARM Powered® supercomputer w/ high application performance

ARM v8 brings out the real strength of Fujitsu's microarchitecture

HPC apps acceleration feature	Post-K	FX100	FX10	K computer
FMA: Floating Multiply and Add	✔	✔	✔	✔
Math. acceleration primitives*	✔Enhanced	✔Enhanced	✔	✔
Inter core barrier	✔	✔	✔	✔
Sector cache	✔Enhanced	✔Enhanced	✔	✔
Hardware prefetch assist	✔Enhanced	✔Enhanced	✔	✔
Tofu interconnect	✔Integrated	✔Integrated	✔	✔

* Mathematical acceleration primitives include trigonometric functions, sine & cosines, and exponential function

2

Copyright 2016 FUJITSU LIMITED

Announcement from Fujitsu
about the adoption of ARM v8 as post-K's ISA (at ISC 2016)
https://www.fujitsu.com/global/Images/moving-forward-the-next-step-in-fujitsu-supercomputing.pdf

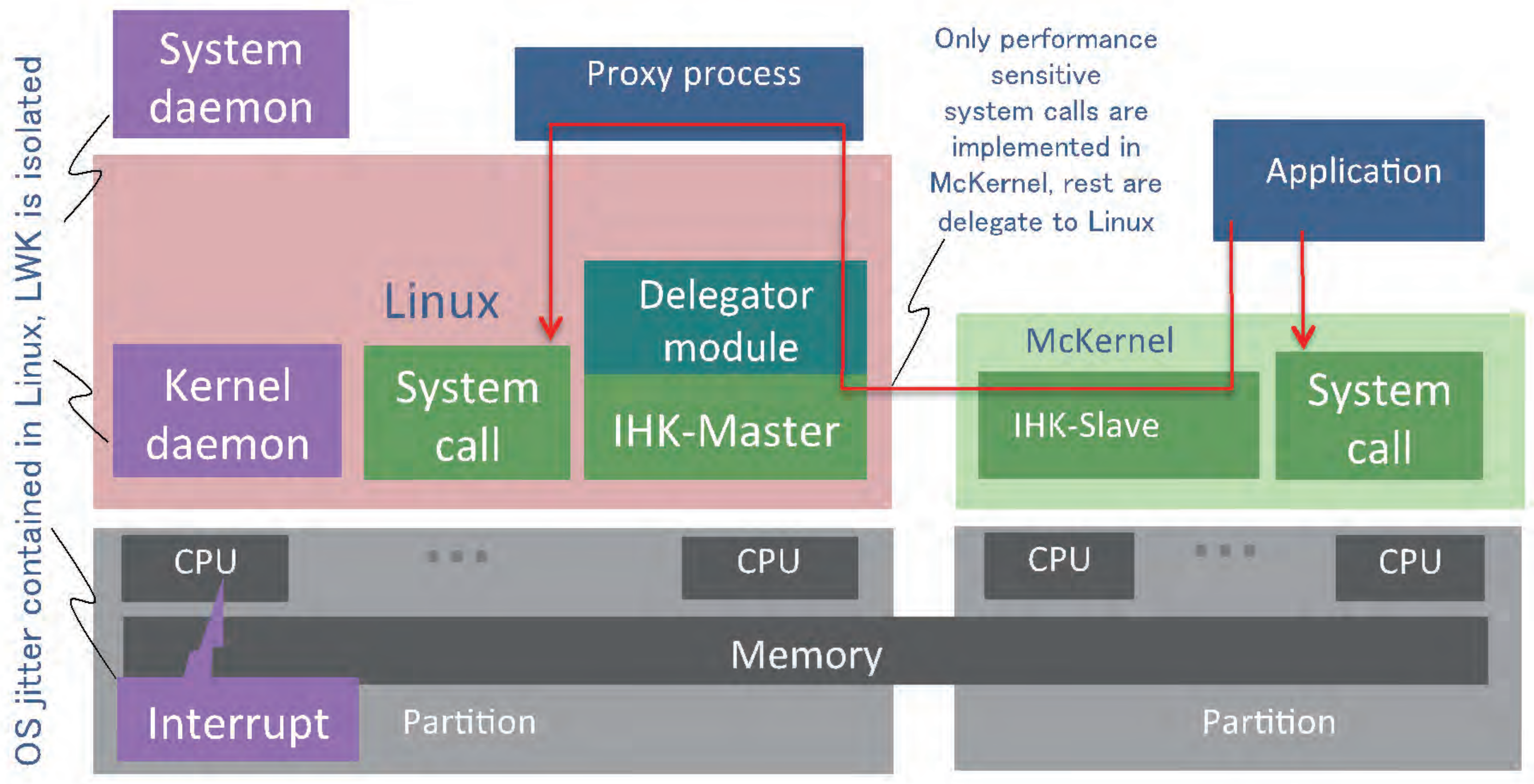
System Software Development Team
Architecture Development Team

System Software Development Team

Team Leader : Yutaka ISHIKAWA
contact : http://www.sys.aics.riken.jp/

The system software development team designs and develops system software for the post K supercomputer, focusing broadly on three topics.

A lightweight multi-kernel based operating system (called IHK/McKernel) that combines Linux with a lightweight kernel (LWK) to achieve the followings:
- Provide LWK scalability and full Linux API compatibility at the same time. Linux compatibility is retained by selectively offloading OS services from the LWK to Linux
- Provide kernel level specialization both for hardware and application specific needs



Interface for Heterogeneous Kernels (IHK) and McKernel provide a multi-kernel operating system (OS) that is tailored to high-end HPC but retains full Linux compatibility

Communication:
- RIKEN MPI (MPICH-based MPI library)
  - Support latest and even in-draft MPI standards
  - Provide scalable performance
- Broader communication library
  - Connect compute-nodes to off-site machines

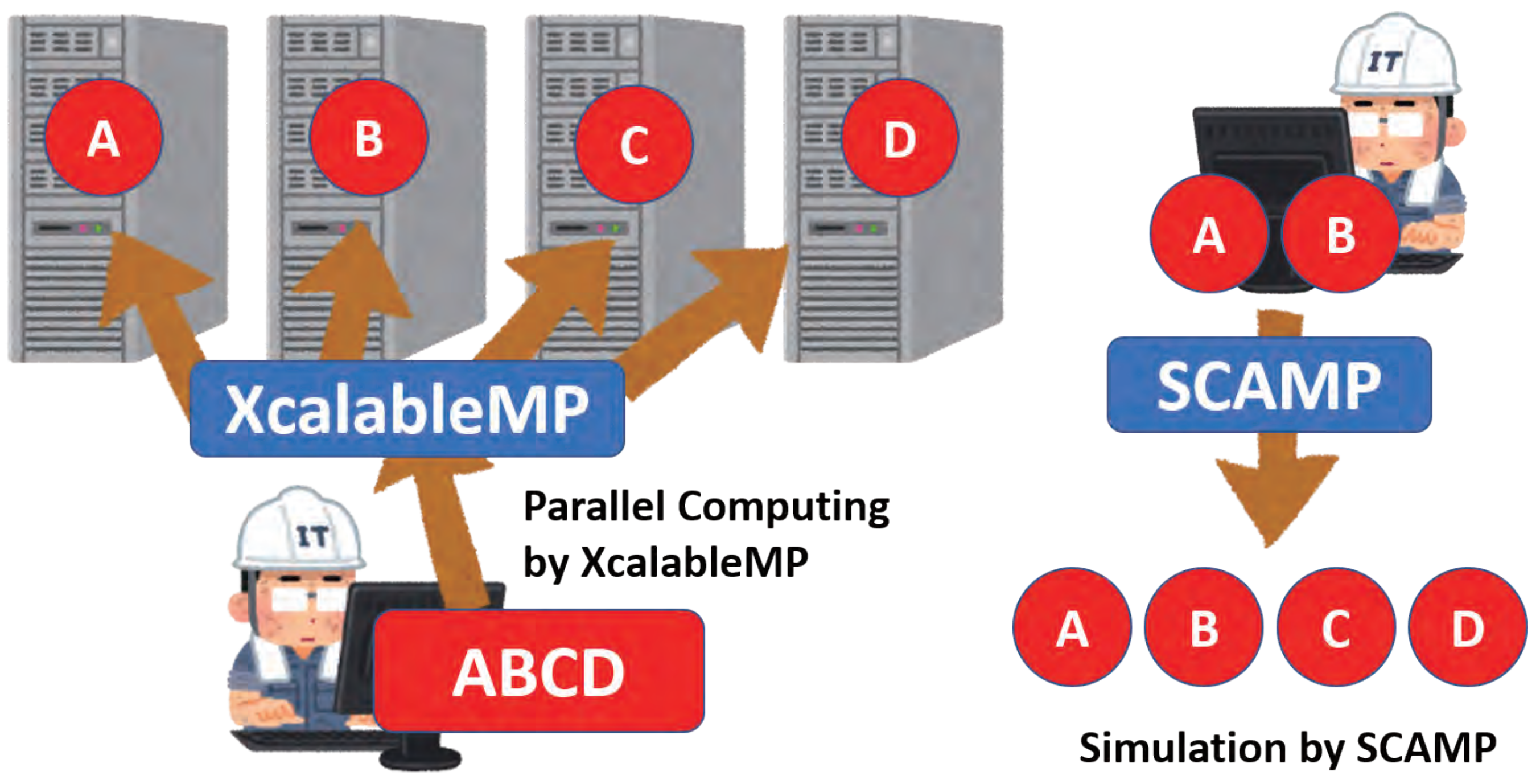
File I/O and Hierarchical Storage:
- Provide abstractions to efficiently deal with multi-level storage hierarchy
- Absorb bursty I/O traffic and provide scalable caching

Architecture Development Team

Team Leader : Mitsuhisa SATO contact : msato@riken.jp

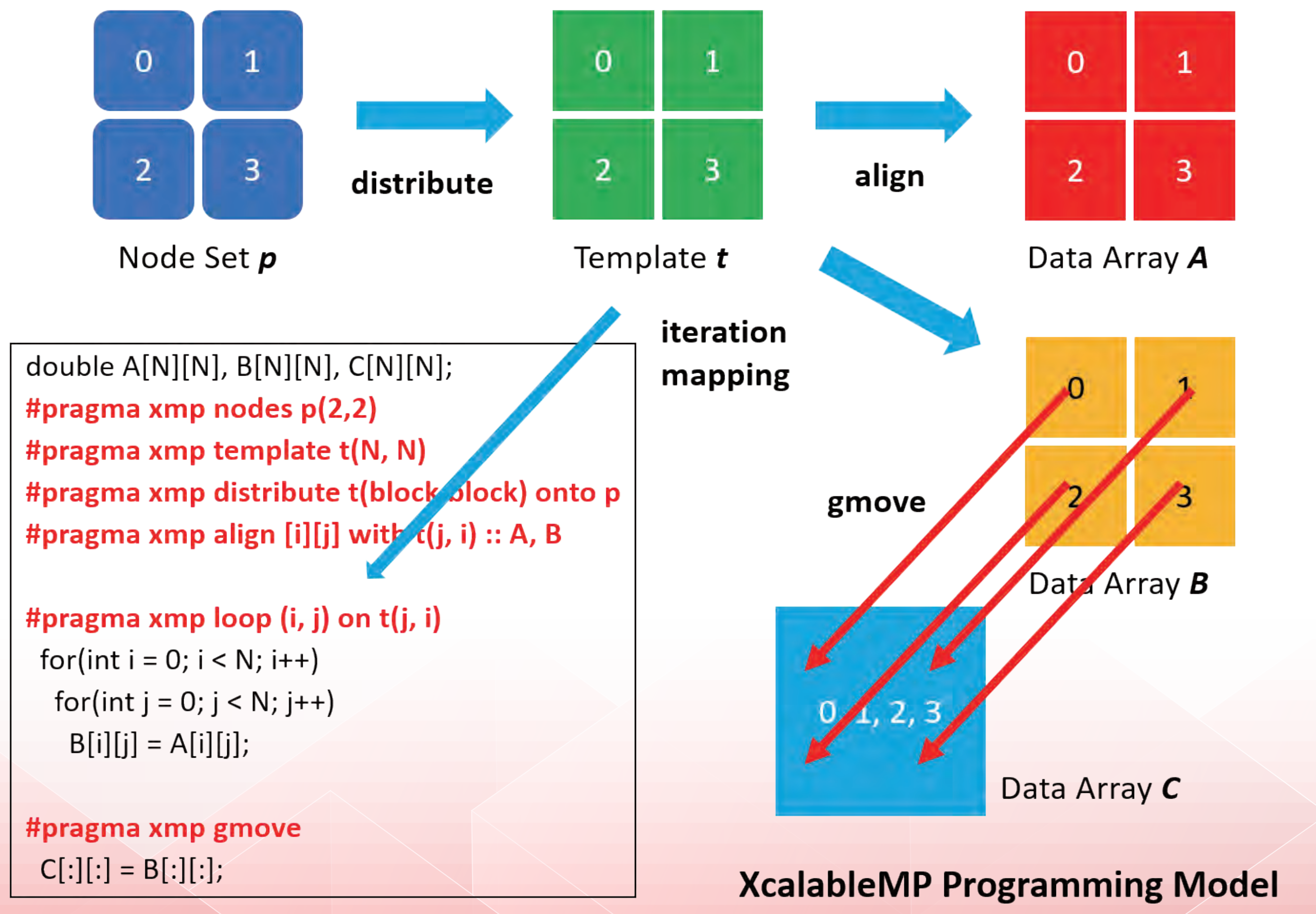
The architecture development team designs the architecture of the post K supercomputer and the programming environment in cooperation with our partner company, Fujitsu.

Co-design Tools
- CPU simulator for Post-K
  - based on gem5, capable of Out-of-Order execution
- MPI application replay tool
  - investigates parallel applications on a single node
- SCAMP (SCAlable Mpi Profiler)
  - simulates a large scale network from a small # of profiling results



XcalableMP
- PGAS programming language for cluster computing
  - OpenMP-like directives for data-parallelism
  - Coarray syntax for one-sided communication
- Omni compiler: http://www.xcalablemp.org/
- Extension for multitasking using lightweight threads

Compiler Development based on LLVM
- OpenMP extensions
  - task allocation for NUMA optimization
  - user interface for target specific SIMD programming
- SIMD vectorization for ARM SVE



```
double A[N][N], B[N][N], C[N][N];
#pragma xmp nodes p(2,2)
#pragma xmp template t(N, N)
#pragma xmp distribute t(block, block) onto p
#pragma xmp align [i][j] with t(j, i) :: A, B

#pragma xmp loop (i, j) on t(j, i)
  for(int i = 0; i < N; i++)
    for(int j = 0; j < N; j++)
      B[i][j] = A[i][j];

#pragma xmp gmove
  C[:][:] = B[:][:];
```

XcalableMP Programming Model

Application Development Team
Co-Design Team

Application Development Team

Team Leader : Hirofumi TOMITA contact : htomita@riken.jp

Co-design Based on Target Applications

The Application Development Team co-designs the applications and the Post-K system through optimization and sophistication of the Target Applications. The team works in cooperation with the developers of the Target Applications in Institutes and Universities throughout Japan.

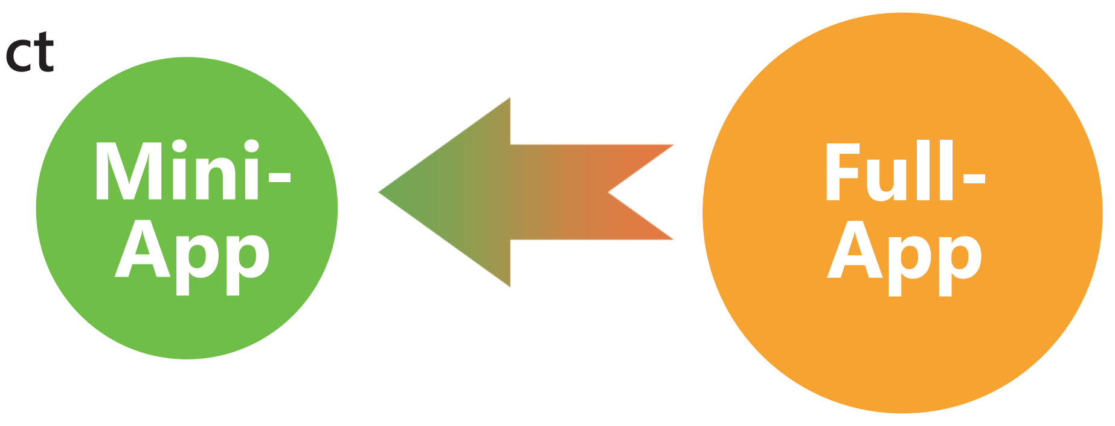
The "Target Applications" are selected from the nine social and science priority issues of Post-K project to represent the features of wide variety of applications.

	Application	Feature
I	GENESIS	Classical MD of biomolecules (Particle simulation)
II	Genomon	Genome processing (Genome alignment)
III	GAMERA	Earthquake simulator (FEM in structured & unstructured grid)
IV	NICAM+LETKF	Weather prediction system with Big Data (Structured grid stencil & ensemble Kalman filter)
V	NTChem	Molecular electronic structure calculation (Post HF)
VI	ADVENTURE	General-purpose computational mechanics system (3D FEM in unstructured grid)
VII	RSDFT	Electronic structure calculation with Density Functional Theory
VIII	FFB	Large eddy simulation (Unstructured grid)
IX	LQCD	Lattice QCD simulation (Structured grid Monte Carlo)

Development of Mini-Applications

The team develops "Mini-Applications", which is simplified by confining the calculation condition, but inherits the main feature of the full application. This can be employed as a benchmark for critical evaluation of future HPC systems including Post-K.

Simple and compact
1K ~ 10K lines
Communication,
file I/O
Open Source



Large and complex
10K+ ~ 100K+ lines
Various forms of
parallel execution,
communication, I/O
Source not
always available

https://fiber-miniapp.github.io/

Development of Application Infrastructures

The team develops general numerical libraries and domain-specific frameworks to maximize the performance of applications on the Post-K system.

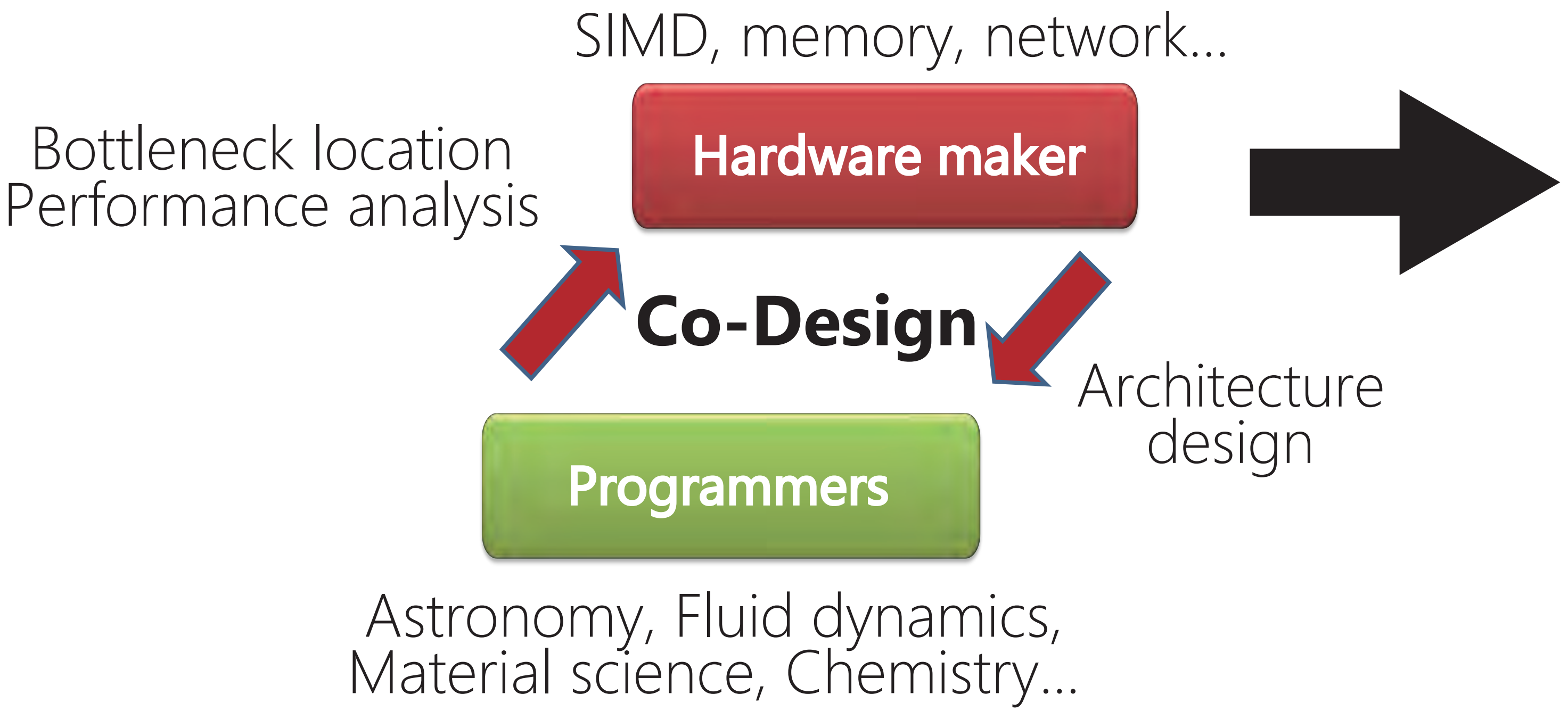
Project for Future HPC Application

The team leads the promotional activities of future HPC applications, by investigating the social and scientific challenges to be solved in the next 5-10 years, for future HPC projects.

Co-Design Team

Team Leader : Junichiro MAKINO contact : jmakino@riken.jp

What's Co-Design?

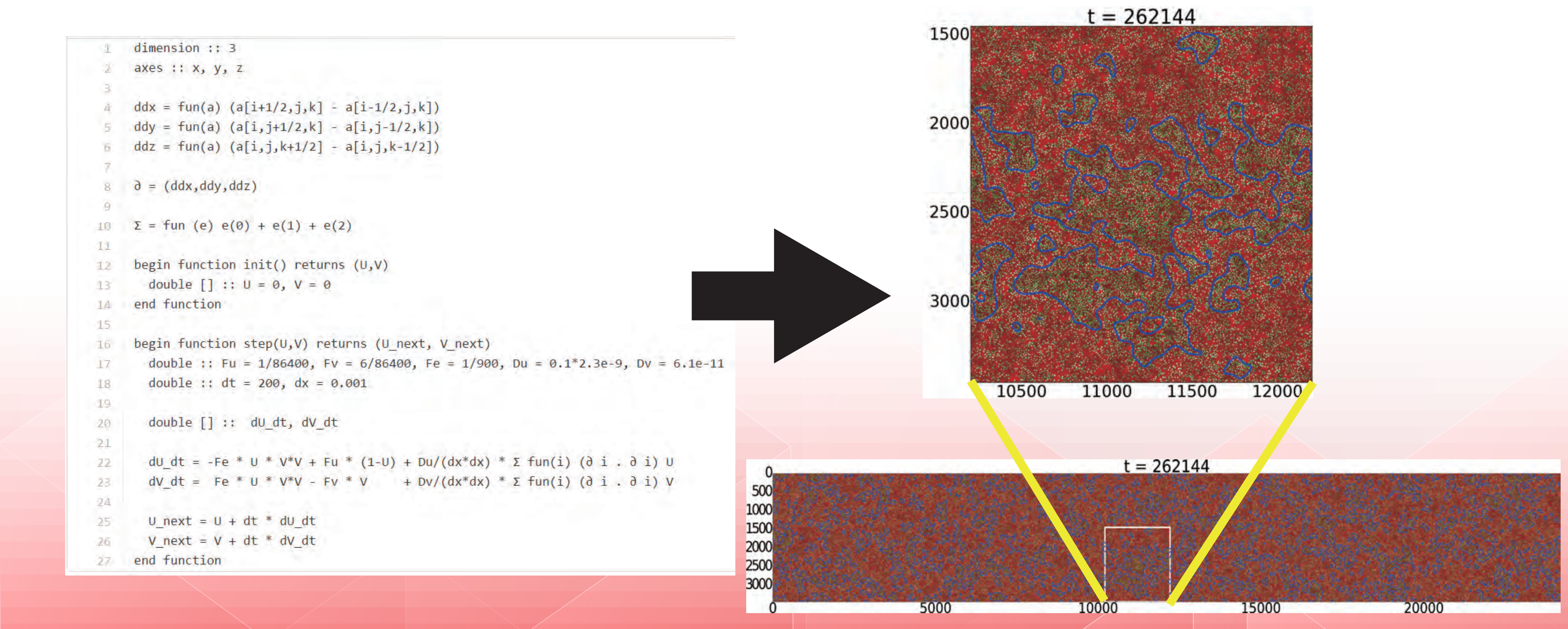


Highly optimized applications
on new supercomputer
with high efficiency/performance

Co-Design team develops framework for HPC users to implement advanced algorithms

Formura

- Domain specific language for optimized stencil computations
- Simple math notation for higher order integration scheme
- Auto tuning of generated C code with MPI
- Underground biology simulation runs with 1.184 Pflops on K computer (Muranushi et al., SC16, 2016, Salt Lake City, USA.)
- Available at https://github.com/nushio3/formura

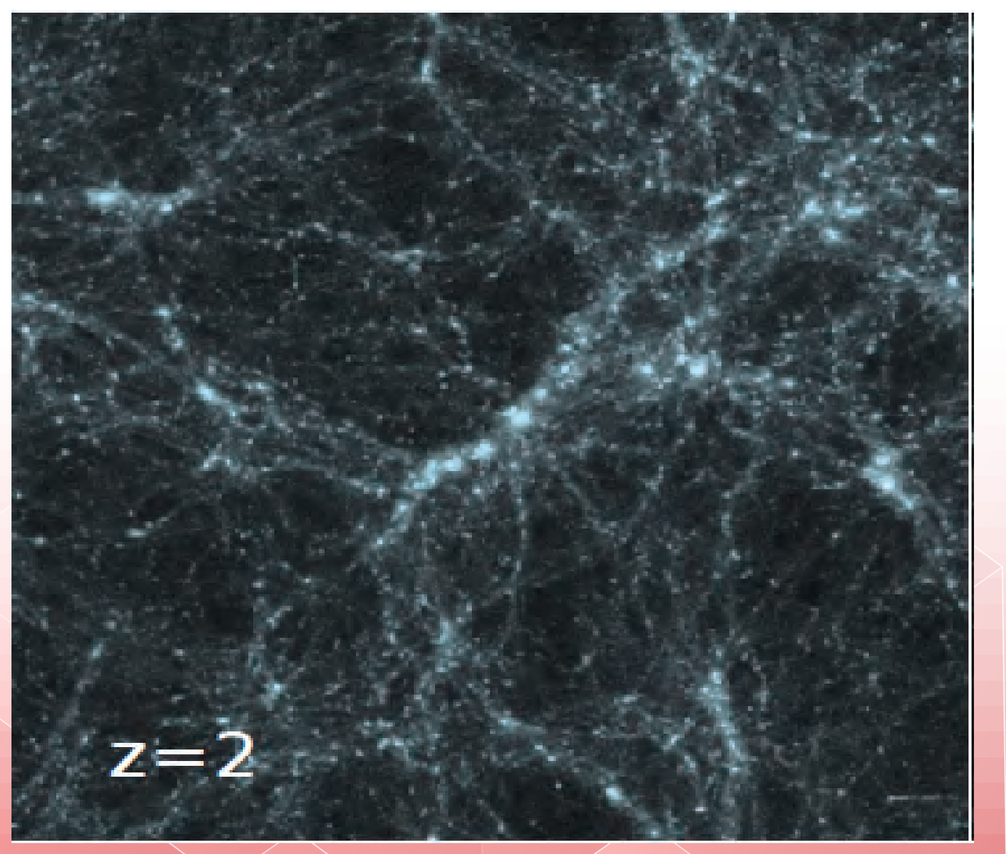


```
dimension :: 3
axes :: x, y, z

ddx = fun(a) (a[i+1/2,j,k] - a[i-1/2,j,k])
ddy = fun(a) (a[i,j+1/2,k] - a[i,j-1/2,k])
ddz = fun(a) (a[i,j,k+1/2] - a[i,j,k-1/2])

∂ = (ddx,ddy,ddz)

Σ = fun (e) e(0) + e(1) + e(2)

begin function init() returns (U,V)
  double [] :: U = 0, V = 0
end function

begin function step(U,V) returns (U_next, V_next)
  double :: Fu = 1/86400, Fv = 6/86400, Fe = 1/900, Du = 0.1*2.3e-9, Dv = 6.1e-11
  double :: dt = 200, dx = 0.001

  double [] ::  dU_dt, dV_dt

  dU_dt = -Fe * U * V*V + Fu * (1-U) + Du/(dx*dx) * Σ fun(i) (∂ i . ∂ i) U
  dV_dt =  Fe * U * V*V - Fv * V     + Dv/(dx*dx) * Σ fun(i) (∂ i . ∂ i) V

  U_next = U + dt * dU_dt
  V_next = V + dt * dV_dt
end function
```

Simple 27-column Formura code (left) is compiled to optimized code of large-scale simulation (right)

FDPS

- Framework for Developing Particle Simulator (Iwasawa et al., PASJ, 68, 54, 2016.)
- User only programs particle integration and force kernel w/o parallelization
- Optimized parallel MPI/OpenMP code generation
- GPU support / Fortran interface
- Available at https://github.com/FDPS/FDPS



Simulation of large-scale cosmic structure formation using FDPS