Post K Supercomputer of FLAGSHIP 2020 Project

The post K supercomputer of the FLAGSHIP2020 Project under the Ministry of Education, Culture, Sports, Science, and Technology began in 2014 and RIKEN has been appointed as the main organization for development. Our development policies can be summarized as follows:

- Developing the world's most advanced supercomputer as the successor to the K computer with the aim for launching operations in 2020
- Co-designing the system with the world's top-level applications in order to solve important social and scientific problems
- Developing new technologies and promoting international standardization of software and mini-applications through strategic international collaborations
- Installing the post K machine at AICS in Kobe to enable maximum use of the facilities, technologies, human resources, and applications established with the K

FLAGSHIP 2020 Project

FLAGSHIP 2020 Project
Ministry of Education, Culture, Sports, Science and Technology (MEXT)

Supercomputer Development (Post K Supercomputer)

RIKEN Advanced Institute for Computational Science

- System Software Development Team
- Architecture Development Team
- Application Development Team
- Co-Design Team

→See "Overview of the Post K Supercomputer Project Development Activities"

Vendor Partner : Fujitsu Ltd.

Application Development

Research institutes and universities which in charge of development applications for following social and scientific priority issues

- Priority issues (9 issues)
- Exploratory challenges (4 challenges)

→See "Social and Scientific Priority Issues to be Tackled by Using Post K Supercomputer"

Co-design

Hardware / Architecture

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

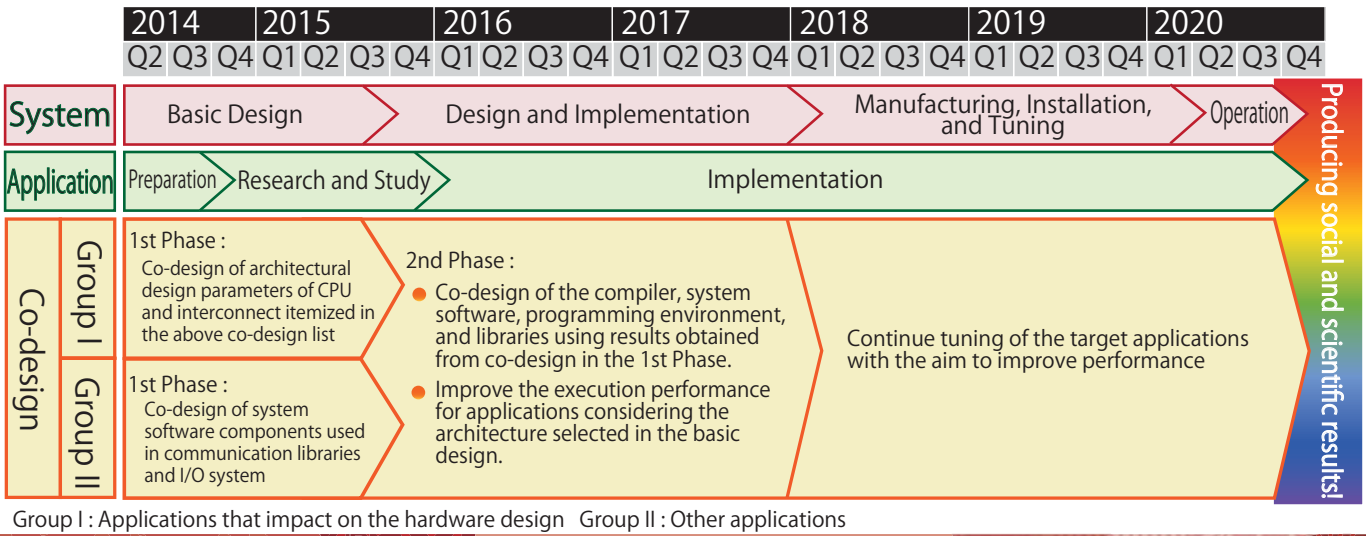
System Software

- operating system for many core architecture
- communication and file I/O libraries
- programming model and languages

Algorithms and Mathematics Libraries

- dense and sparse solvers
- eigen solvers
- domain-specific libraries and frameworks

Schedule



Group I : Applications that impact on the hardware design Group II : Other applications

Social and Scientific Priority Issues to be Tackled by Using Post K Supercomputer

RIKEN AICS

Priority Issues (9 Issues)

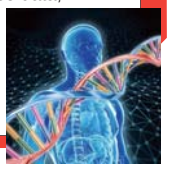
Health and Longevity

I. Innovative Drug Discovery Infrastructure through Functional Control of Biomolecular Systems

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.

II. Integrated Computational Life Science to Support Personalized and Preventive Medicine

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.



Disaster Prevention and Global Climate Problem

III. Development of Integrated Simulation Systems for Hazard and Disaster Induced by Earthquake and Tsunami

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.

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

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

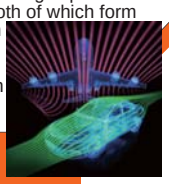
Development of Basic Science

IX. Elucidation of the Fundamental Laws and Evolution of the Universe

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.

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

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 efforts, high value-added product development can be achieved.



Enhancement of Industrial Competitiveness

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

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.

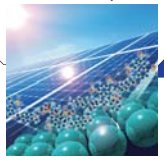
VI. Accelerated Development of Innovative Clean Energy Systems

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.

Energy Problem

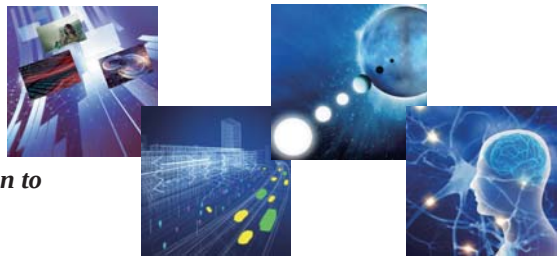
V. Development of New Fundamental Technologies for High-Efficiency Energy Creation, Conversion/Storage and Use

Develop new fundamental technologies to resolve energy-related problem, and 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.



Exploratory Challenges (4 Challenges)

- ★ *Frontiers of basic science: challenge to the limits*
- ★ *Construction of models for interaction among multiple socioeconomic*
- ★ *Elucidation of the birth of exoplanets (Second Earths) and the environmental variations of planets in the solar system*
- ★ *Elucidation of how neural networks realize thinking and its application to artificial intelligence*



System Software Development Team

Team Leader
Yutaka Ishikawa

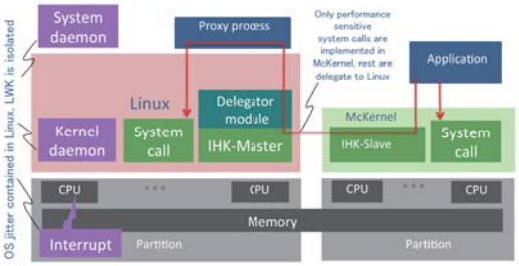
RIKEN AICS

The system software development team designs and develops system software for the post-K Supercomputer, focusing broadly on operating systems, high performance communication, I/O and storage

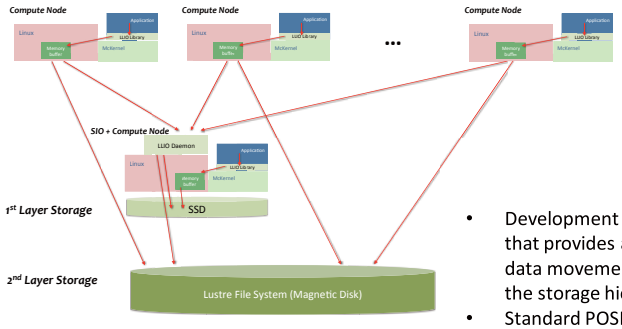
Operating System for Exa-scale and beyond

- Development of a hybrid OS stack (called IHK/McKernel) that seamlessly blends Linux with a light-weight kernel (LWK) designed specifically for high performance computing
- Our kernel infrastructure aims at the followings:
  - Provide scalable and consistent performance for large scale bulk synchronous HPC simulations
  - Support the full POSIX/Linux APIs by selectively offloading system calls to Linux
  - Provide efficient memory and device management so that resource contention and data movement are minimized inside the kernel
  - Eliminate OS noise by isolating OS services

IHK/McKernel Hybrid Lightweight Kernel Architecture

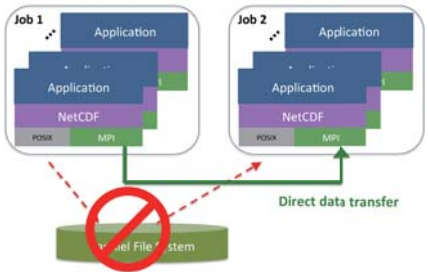


Hierarchical Storage



- Development of an I/O middleware layer that provides access to and orchestrates data movement across multiple layers of the storage hierarchy
- Standard POSIX system call interface
- Transparent access to:
  - Compute node memory buffers
  - SSDs on a subset of compute nodes (i.e., burst buffer)
  - Global parallel file system
- Asynchronous I/O processing

Direct Data Transfer for Workflows



- Support direct data transfer among MPI jobs relying on netCDF APIs
- No (or minimal) changes to application code

International Collaboration

International Collaboration between DOE (USA) and MEXT (Japan)

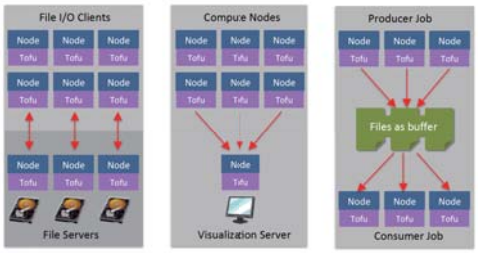
Purpose:
Work together where it is mutually beneficial to expand the HPC ecosystem and improve system capability

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

openHPC

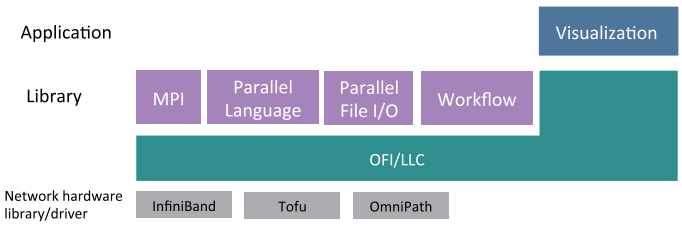
The system software stack developed at RIKEN is open source and will be contributed to the OpenHPC community

Low Level Communication Library



- Development of a unified low-level communication infrastructure
- Sits below MPI and above HW drivers to support the following use-cases
  - File I/O access
  - In-situ visualization
  - Cross-job communication
- Adopting OFI and called OFI/LLC

- The library aims at the followings:
  - Limit memory consumption to support scalable execution
  - Exploit Linux core when using McKernel
  - Exploit connection-less transport to reduce latency



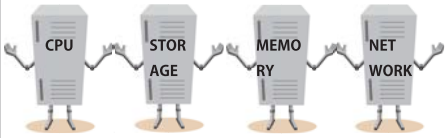
Architecture Development Team

Team Leader
Mitsuhisa Sato

RIKEN AICS

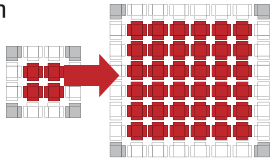
Overview: Co-design the post K supercomputer

Co-design is a bi-directional approach, where a system would be designed on demand from applications and applications must be optimized to the system.
The architecture development team designs and develops the architecture of the post K supercomputer in cooperation with our partner vendor based on the co-design concept.
We also have been developing various co-design tools and parallel programming language.



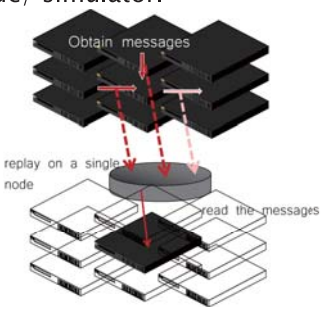
Scamable Network Simlator

The trace driven network simulator performs an important role in "co-design". However, sometimes, it is not appropriate for the simulation of large parallel systems since it is difficult to obtain the number of trace files for a target system if the current system is smaller than the target one. We propose SCAMP (SCAlable Mpi Profiler) to tackle the scaling-problem in the trace driven simulator.
- creates a large number of pseudo trace files based on the small number of trace files and application analysis
- drives the network simulator using the pseudo trace files to estimate the performance of the target system



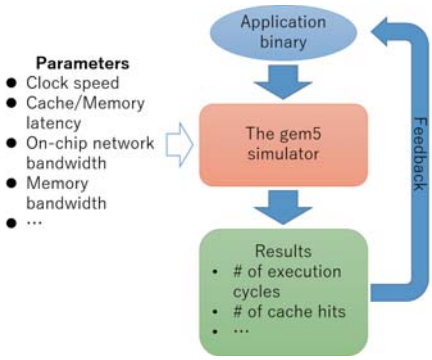
MPI Application Replay Tool

We have been developing libraries to be used to reply MPI applications on a single node and CPU simulator.
While some parallel applications behave differently when they run on a single node, we are interested in their behavior when they run in parallel. To investigate the performance of parallel applications on a single node, a library is used to obtain MPI messages on parallel systems (existing supercomputers) and another library is used to replay applications on a single node/ simulator.



CPU Simulator

Performance estimation and tuning of applications are difficult without real-systems.
So, we focus on the gem5 simulator, which is a modular platform for computer-system architecture research.
User can obtain more accurate results by setting parameters of assumed systems to the gem5.
These results are expected to help in application developments.



XcalableMP

XcalableMP (XMP) is a directive-based language extension which allows users to develop parallel programs for distributed memory systems easily and to tune the performance by having minimal and simple notations. For the detail of XMP, please visit the web-site http://www.xcalablemp.org

```
int array[YMAX][XMAX];
#pragma xmp nodes p(*)
#pragma xmp template t(0:YMAX-1)
#pragma xmp distribute t(block) onto p
#pragma xmp align array[i][*] with t(i)

main(){
#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];}}
```

Compiler for Parallel Programming for Many-Core Architectures

For larage scale systems based on many core processor, easy programming model and efficient runtime are required.
We have been developing a compilier to exploit intra and inter-node parallelism by combining Omni-XMP compiler developed at RIKEN/AICS and Argobots developed at Argonne National Laboratory. In the compiler,
Intra-node parallelism primary using OpenMP
- Lightweight threads in OpenMP
- Compiler support for intelligent scheduling of lightweight threads

Inter-node parallelism primary using XMP PGAS language
- Investigating MPI-3 capabilities and its benefits to PGAS

Co-Design Team

Team Leader
Junichiro Makino

RIKEN AICS

We are in charge of ``co-design'' of the hardware, the system software, and the application software for the post-K supercomputer.

Why we need co-design?

Modern processors are complicate system with
- Many processor, many cores, long SIMD
- Complicated memory & communication structure

Vacuum tube module of IBM 700 Series (1953-)

(c)Autopilot CC BY-SA wikipedia.org

Facom VP-100 (1982)

(c) Easy Management, Inc. Fujitsu Limited

The K computer (2012-)

Hardware makers alone...

find it difficult to make *the* general-purpose processor that execute *any* program optimally.

Programmers alone...

find it difficult to learn details of hardware features to write fast programs.

Therefore, we need to design and optimize hardware and software together.

→ That's co-design!!

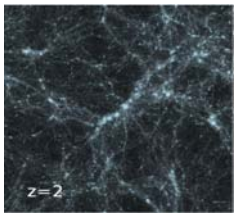
That's our Mission!!

Software for co design

We design and develop application frameworks and domain specific languages (DSLs) to help HPC users implement advanced algorithms.

FDPS

... is a library for massively parallel particle simulations. Users only need to program particle interactions and do not need to parallelize the code with MPI. FDPS generates a parallel code that scales up to the K computer using highly-optimized communicationalgorithms. Now, FDPS supports GPU clusters.



Simulation of large-scale cosmic structure formation, using FDPS.

FDPS is available at https://github.com/FDPS/FDPS !!

(For more detail, see Iwasawa et al., 2016, preprint [arXiv:1601.03138])

Development of Post-K computer

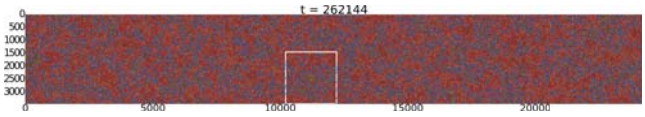
●Initial Phase (2014-2015):
Analyse application performance, locate bottleneck
→ Co-improvement of hardware and application software

●Late Phase (2015-):
More improvement on applications

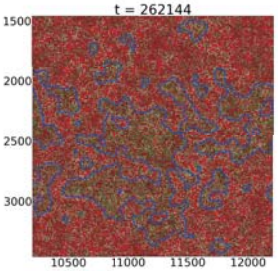
Formura

... Is a domain specific language that provides access to optimized stencil computations. Higher-order integration schemes can be defined using mathematical notations.

Formura generates C code with MPI calls, and realizes portable performance via automated tuning. Formura have been applied to magnetohydrodynamics (MHD) and belowground biology simulations. For the latter, scaling up to the full nodes of the K computer, with 1.157 Pflops, 11.06% floating-point operation efficiency, is demonstrated.



(↑) The below-ground biology simulation using Formura.
(→) Close -up of the white box
(↓) The source code for this simulation




```
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}

I = 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) * I fun(i) {∂ i . ∂ i} U
  dV_dt =  Fe * U * V*V - Fv * V     + Dv/(dx*dx) * I fun(i) {∂ i . ∂ i} V

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

available at https://github.com/nushio3/formura

Application Development Team

Team Leader
Hirofumi Tomita

Toward maximizing outcome with the post K-computer, the Application Development Team conducts research and development with related researchers in architecture, system software and algorithm fields. Its core missions are as follows;

Co-design based on Target applications

Target Applications are representative applications chosen from nine social and scientific priority issues for the post K supercomputer.
The application development team carries out performance analysis and optimization of the target applications to reflect the co-design of applications and the system. The team works in co-operation with the executing agencies of the target applications and other colleague teams of this project.

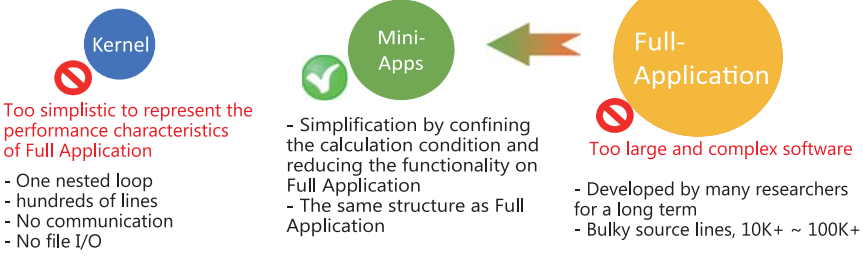
	Program	Brief description
I	GENESIS	MD for proteins
II	Genomon	Genome processing (Genome alignment)
III	GAMERA	Earthquake simulator (FEM in unstructured & structured grid)
IV	NICAM+LETKF	Weather prediction system using Big data (Structured grid stencil & ensemble Kalmanfilter)
V	NTChem	Molecular electronic (Structure calculation)
VI	ADVENTURE	Computational mechanics system for large scale analysis and design (Unstructured grid)
VII	RSDFT	An ab-initio program (Density functional theory)
VIII	FFB	Large eddy simulation (Unstructured grid)
IX	LQCD	Lattice QCD simulation (Structured grid Monte Carlo)

Selected target applications for co-design

Development of Fiber Mini-Applications

We develop and maintain "mini-apps" that are designed to represent the same performance characteristics as full applications from major computational science domains.
Our aim is to establish a mini-app benchmark suite that is widely and internationally used for system performance evaluation.
Collaborative research with universities will be conducted, aiming for establishment of system performance evaluation methodologies.

FIBER
http://fiber-miniapp.github.io/



Name	Calculation method	Parallel method			Communication type
		MPI	OpenMP	OpenACC	
CCS QCD	Lattice QCD	✓	✓	✓(**)	Boundary + Collective
MARBLE	MD(PME)	✓	✓		Boundary + All to All
MODYLAS	MD(FMM)	✓	✓		Boundary + Collective
FFVC	CFD	✓	✓		Boundary + Collective
FFB	CFD(FEM)	✓			Collective
NGS Analyzer	Genome sequence matching	✓			through the File I/O
NICAM-DC	Climate simulations	✓	✓	ongoing	Boundary + Collective
mVMC	Quantum Monte Carlo	✓	✓		Collective
Ntchem	Molecular electronic structure calculation	✓	✓		Collective

List of Fiber Mini Applications (as of May 2016)

Research and Development of Application Infrastructures

We develops general numerical libraries and domain-specific frameworks for improving application programming infrastructure on the post K supercomputer.

Furthermore, we also leads the promotional activities to continue the research work started by the Application Working Group of the "Feasibility Study on Future HPC Infrastructures" project, extracting social and scientific challenges to be solved by HPC during the next 5-10 years.