Chapter 22

Flagship 2020 Project

22.1 Members

Primary members are only listed.

22.1.1 System Software Development Team

Yutaka Ishikawa (Team Leader)

Masamichi Takagi (Senior Scientist)

Atsushi Hori (Research Scientist)

Balazs Gerofi (Research Scientist)

Masayuki Hatanaka (Research & Development Scientist)

Takahiro Ogura (Research & Development Scientist)

Tatiana Martsinkevich (Postdoctoral Researcher)

Fumiyoshi Shoji (Research & Development Scientist)

Atsuya Uno (Research & Development Scientist)

Toshiyuki Tsukamoto (Research & Development Scientist)

22.1.2 Architecture Development

Mitsuhisa Sato (Team Leader)

Yuetsu Kodama (Senior Scientist)

Miwako Tsuji (Research Scientist)

Hidetoshi Iwashita (Research & Development Scientist)

Jinpil Lee (Postdoctoral Researcher)

Tetsuya Odajima (Postdoctoral Researcher)

Hitoshi Murai (Research Scientist)

Toshiyuki Imamura (Research Scientist)

Table 22.1: Development Teams

Team Name	Team Leader
Architecture Development	Mitsuhisa Sato
System Software Development	Yutaka Ishikawa
Co-Design	Junichiro Makino
Application Development	Hirofumi Tomita

22.1.3 Application Development

Hirofumi Tomita (Team Leader)

Yoshifumi Nakamura (Research Scientist)

Hisashi Yashiro (Research Scientist)

Seiya Nishizawa (Research Scientist)

Hiroshi Ueda Research (Scientist)

Yukio Kawashima (Research Scientist)

Naoki Yoshioka (Research Scientist)

Yiyu Tan (Research Scientist)

Soichiro Suzuki (Research & Development Scientist)

Kazunori Mikami (Research & Development Scientist)

22.1.4 Co-Design

Junichiro Makino (Team Leader)

Keigo Nitadori (Research Scientist)

Yutaka Maruyama (Research Scientist)

Takayuki Muranushi (Postdoctoral Researcher)

22.2 Project Overview

The Japanese government launched the FLAGPSHIP 2020 project [1] in FY 2014 whose missions are defined as follows:

- Building the Japanese national flagship supercomputer, the successor to the K computer, which is tentatively named the post K computer, and
- developing wide range of HPC applications that will run on the post K computer in order to solve the pressing societal and scientific issues facing our country.

RIKEN AICS is in charge of co-design of the post K computer and development of application codes in collaboration with the Priority Issue institutes selected by Japanese government, as well as research aimed at facilitating the efficient utilization of the post K computer by a broad community of users. Under the co-design concept, AICS and the selected institutions are expected to collaborate closely.

As shown in Table 22.1, four development teams are working on post K computer system development with the FLAGSHIP 2020 Planning and Coordination Office that supports development activities. The primary members are listed in Section 22.1.

The Architecture Development team designs the architecture of the post K computer in cooperation with Fujitsu and designs and develops a productive programming language, called XcalableMP (XMP), and its

[1] FLAGSHIP is an acronym for Future LAtency core-based General-purpose Supercomputer with HIgh Productivity.

tuning tools. The team also specifies requirements of standard languages such as Fortran and C/C++ and mathematical libraries provided by Fujitsu.

The System Software Development team designs and specifies a system software stack such as Linux, MPI and File I/O middleware for the post K computer in cooperation with Fujitsu and designs and develops multi-kernel for manycore architectures, Linux with light-weight kernel (McKernel), that provides a noise-less runtime environment, extendability and adaptability for future application demands. The team also designs and develops a low-level communication layer to provide scalable, efficient and portability for runtime libraries and applications.

The Co-Design team leads to optimize architectural features and application codes together in cooperation with AICS teams and Fujitsu. It also designs and develops an application framework, FDPS (Framework for Developing Particle Simulator), to help HPC users implement advanced algorithms.

The Application Development team is a representative of nine institutions aimed at solving Priority Issues. The team figures out weakness of target application codes in terms of performance and utilization of hardware resources and discusses them with AICS teams and Fujitsu to find out best solutions of architectural features and improvement of application codes.

22.3 Target of System Development and Achievements in FY2016

The post K's design targets are as follows:

- A one hundred times speed improvement over the K computer is achieved in maximum case of some target applications. This will be accomplished through co-design of system development and target applications for the nine Priority Issues.

- The maximum electric power consumption should be between 30 and 40 MW.

In FY2016, the second phase of the detailed design was completed. The major components of system software are summarized as follows:

- Highly productive programming language, XcalableMP
  XcalableMP (XMP) is a directive-based PGAS language for large scale distributed memory systems that combines HPF-like concept and OpenMP-like description with directives. Two memory models are supported: global view and local view. The global view is supported by the PGAS feature, i.e., large array is distributed to partial ones in nodes. The local view is provided by MPI-like + Coarray notation.

- Domain specific library/language, FDPS
  FDPS is a framework for the development of massively parallel particle simulations. Users only need to program particle interactions and do not need to parallelize the code using the MPI library. The FDPS adopts highly optimized communication algorithms and its scalability has been confirmed using the K computer.

- MPI + OpenMP programming environment
  The current de facto standard programming environment, i.e., MPI + OpenMP environment, is supported. Two MPI implementations are being developed. Fujitsu continues to support own MPI implementation based on the OpenMPI. RIKEN is collaborating with ANL (Argonne National Laboratory) to develop MPICH, mainly developed at ANL, for post K computer. In FY2016, we fixed the issue that launching many processes took unacceptable time on a large scale manycore-based cluster such as OakForest-PACS. Other achievements[3] have been described in Section 1.3.1.

- New file I/O middleware
  The post K computer does not employ the file staging technology for the layered storage system. The users do not need to specify which files must be staging-in and staging-out in their job scripts in the post K computer environment. The LLIO midleware, employing asynchronous I/O and caching technologies, has been being designed by Fujitsu in order to provide transparent file access with better performance. The functional design of LLIO was completed in FY2016.

- Application-oriented file I/O middleware
  In scientific Big-Data applications, such as real-time weather prediction using observed meteorological data, a rapid data transfer mechanism between two jobs, ensemble simulations and data assimilation,

is required to meet their deadlines. In FY2016, a framework called Data Transfer Framework (DTF), based on PnetCDF file I/O library, that silently replaces file I/O with sending the data directly from one component to another over network was designed and its prototype system was implemented and evaluated. The detailed achievement[4] has been described in Section 1.3.2.

- Process-in-Process
  "Process-in-Process" or "PiP" in short is a user-level runtime system for sharing an address space among processes. Unlike the Linux process model, a group of processes shares the address space and thus the process context switch among those processes does not involve hardware TLB flushing. It was implemented in FY2016, and its applicability to a communication mechanism has been tested. The detailed achievement has been described in Section 1.3.4.

- Multi-Kernel for manycore architectures
  Multi-Kernel, Linux with light-weight Kernel (McKernel) is being designed and implemented. It provides: i) a noiseless execution environment for bulk-synchronous applications, ii) ability to easily adapt to new/future system architectures, e.g., manycore CPUs, a new process/thread management, a memory management, heterogeneous core architectures, deep memory hierarchy, etc., and iii) ability to adapt to new/future application demands, such as Big-Data and in-situ applications that require optimization of data movement. In FY2016, McKernel was improved for NUMA CPU architectures. The detailed improvements have been described in Section 1.3.5.

It should be noted that these components are not only for post K computer, but also for other manycore-based supercomputer, such as Intel Xeon Phi.

22.4 International Collaborations

22.4.1 DOE/MEXT Collaboration

The following research topics were performed under the DOE/MEXT collaboration MOU.

- Optimized Memory Management
  This research collaboration explores OS supports for deep memory hierarchies. In FY2016, the movepages system call was parallelized in McKernel and its applicability for a manycore processor with two memory hierarchies, KNL, was evaluated using a simple stencil code.

- Efficient MPI for exascale
  In this research collaboration, the next version of MPICH MPI implementation, mainly developed by Argonne National Laboratory (ANL), has been cooperatively developed. The FY2016 achievements have been described in the previous section.

- Dynamic Execution Runtime
  This research collaboration shares designs for asynchronous and dynamic runtime systems. In FY2016,

- Metadata and active storage
  This research collaboration, run by the University of Tsukuba as contract, studies metadata management and active storage.

- Storage as a Service
  This research collaboration explores APIs for delivering specific storage service models. This is also run by the University of Tsukuba.

- Parallel I/O Libraries
  This research collaboration is to improve parallel netCDF I/O software for extreme-scale computing facilities at both DOE and MEXT. To do that, the RIKEN side has designed DTF as described in the previous section.

- OpenMP/XMP Runtime
  This research collaboration explores interaction of Argobots/MPI with XscalableMP and PGAS models.

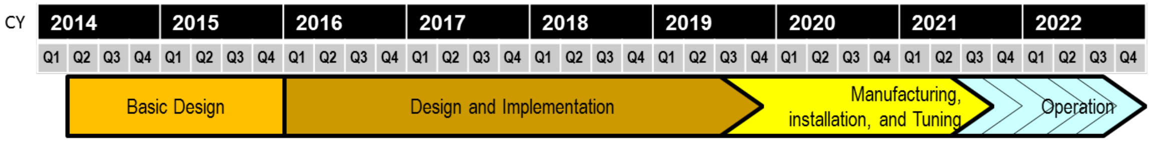

Figure 22.1: Schedule

- Exascale co-design and performance modeling tools
  This collaborates on an application performance modeling tools for extreme-scale applications, and shared catalog of US/JP mini-apps.

- LLVM for vectorization
  This research collaboration explores compiler techniques for vectorization on LLVM.

- Power Monitoring and Control, and Power Steering
  This research collaboration explores APIs for monitoring, analyzing, and managing power from the node to the global machine, and power steering techniques for over-provisioned systems are evaluated.

22.4.2 CEA

RIKEN and CEA, Commissariat *à* l'*é*nergie atomique et aux *é*nergies alternatives, signed MOU in the fields of computational science and computer science concerning high performance computing and computational science in January 2017. The following collaboration topics are now taken into account:

- Programming Language Environment

- Runtime Environment

- Energy-aware batch job scheduler

- Large DFT calculations and QM/MM

- Application of High Performance Computing to Earthquake Related Issues of Nuclear Power Plant Facilities

- Key Performance Indicators (KPIs)

- Human Resource and Training

22.5 Schedule and Future Plan

As shown in Figure 22.1, the design and prototype implementations will be done before the end of 2019, and the system will be deployed after this phase. The service is expected to start public operation at the range from 2021 to 2022.

22.6 Publications

22.6.1 Conference Papers

[1] Y. Kawashima, K. Sawada, T. Nakajima, and M. Tachikawa. Journal of Computer Chemistry, (15):203209, 2016.
[2] Y. Nakamura, Y. Kuramashi, S. Takeda, and A. Ukawa. Critical endline of the Finite temperature phase transition for 2+1 flavor QCD around the SU(3)-flavor symmetric point. PHYSICAL REVIEW D, (94, 114507).

[3] Masayuki Hatanaka, Takahiro Ogura, Masamichi Takagi, Atsushi Hori, and Yutaka Ishikawa. Prototype implementation and evaluation of an mpi persistent neighborhood collective operation using tofu2 protocol offlaoding capability. In IPSJ SIG Technical Report, volume 2016-HPC-157, December 2016. (In Japanese).

[4] Tatiana Martsinkevich, Wei ken Liao, Balazs Gerofi, Yutaka Ishikawa, and Alok Choudhary. Improving multi-component application performance by silently replacing File i/o with direct data transfer: Preliminary results. In IPSJ SIG Technical Report, volume 2017-HPC-158. IPSJ, March 2017.

[5] Y. Nakamura. Lattice qcd with cg and multi-shift cg on xeon phi. In 5th JLESC workshop, Lyon, France, Jun 2016.

[6] Y. Nakamura. Towards high performance lattice qcd simulations on exascale computers. In 6th JLESC workshop, Kobe, Japan, Dec 2016.