Chapter 23

Flagship 2020 Project

23.1 Members

Primary members are only listed.

23.1.1 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)

23.1.2 Architecture Development

Mitsuhisa Sato (Team Leader)

Yuetsu Kodama (Senior Scientist)

Miwako Tsuji (Research Scientist)

Jinpil Lee (Postdoctoral Researcher)

Tetsuya Odajima (Postdoctoral Researcher)

Hitoshi Murai (Research Scientist)

Toshiyuki Imamura (Research Scientist)

Kentaro Sano (Research Scientist)

23.1.3 Application Development

Hirofumi Tomita (Team Leader)

Yoshifumi Nakamura (Research Scientist)

Hisashi Yashiro (Research Scientist)

Seiya Nishizawa (Research Scientist)

Yukio Kawashima (Research Scientist)

Soichiro Suzuki (Research & Development Scientist)

Kazunori Mikami (Research & Development Scientist)

Kiyoshi Kumahata (Research & Development Scientist)

Kengo Miyamoto (Research & Development Scientist)

Mamiko Hata (Technical Staff I)

Kazuto Ando (Technical Staff I)

Hiroshi Ueda (Research Scientist)

Naoki Yoshioka (Research Scientist)

Yiyu Tan (Research Scientist)

23.1.4 Co-Design

Junichiro Makino (Team Leader)

Keigo Nitadori (Research Scientist)

Yutaka Maruyama (Research Scientist)

Masaki Iwasawa (Research Scientist)

Takayuki Muranushi (Postdoctoral Researcher)

Daisuke Namekata (Postdoctoral Researcher)

Long Wang (Postdoctoral Researcher)

Kentaro Nomura (Research Associate)

Miyuki Tsubouchi (Technical Staff)

23.2 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.

Table 23.1: Development Teams

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

RIKEN 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, RIKEN and the selected institutions are expected to collaborate closely.

As shown in Table 23.1, 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 23.1.

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 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 RIKEN 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 RIKEN teams and Fujitsu to find out best solutions of architectural features and improvement of application codes. Published papers, presentation and posters are summerized in the final subsecton in this chapter.

23.3 Target of System Development and Achievements in FY2017

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. In 2017, we finished the front-end for Fortran 2008 Standard for Omni XcalableMP compiler, and are still working on C++ Front-end based on LLVM clang. We are currently working on XcalableMP 2.0 which newly supports task-parallelism and the integration of PGAS models for distributed memory environment.

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

- 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. Achievements of our MPI implementation 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 implementation of LLIO started in FY2017 and will be completed in FY2018.

- 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 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 1.3.3.

- 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 1.3.4.

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.

The architecture development team is also working on the researches on co-design tools as well as the design of the post K supercomputer:

GEM5 processor simulator for the post-K processor

We are developing a cycle-level processor simulator for the Post-K processor based on GEM-5, which is a general-purpose processor simulator commonly used for the processor architecture research. ARM provided us the source code of GEM-5 Atomic-model processor simulator for ARM v8 with Scalable Vector Extension (SVE). The Atomic model enables an instruction-level simulation. We deployed and tested it, and extend it for the cycle-level Out-Of-Order(O3) model processor simulator with the post-K hardware parameters. It enables the cycle-level performance evaluation of application kernels. Currently, we are working on the adjustment of parameters and performance with Fujitsu-in-house processor simulator for more accurate performance evaluation. In 2017, we started the service to provide "post-K performance evaluation environment" including this simulator for performance evaluation and tuning by potential post-K users. And, we presented the preliminary study on the performance of multiple vector lengths by SVE vector-length agnostic feature [2].

Performance estimation tools for co-design study

We have tools for co-design study for future huge-scale parallel systems. The MPI application replay tool is a system to investigate a performance and behavior of parallel applications on a single node using MPI traces. SCAMP (SCAlable Mpi Profiler) is other system to simulate a large scale network from a small number of profiling results.

Study on performance metrics

We have been developing a new metric, called Simplified Sustained System Performance (SSSP) metric, based on a suite of simple benchmarks, which enables performance projection that correlates with applications. In 2017, we presented the preliminary results in the conference [3].

In addition to co-design tools, we are working on the evaluation of compilers for ARM SVE. There are two kinds of compiler for ARM SVE: Fujitsu Compiler and ARM compiler. The Fujitsu compiler is a proprietary compiler supporting C/C++ and Fortran. The ARM compiler is developed by ARM based on LLVM. Initially, LLVM only supports C and C++, and supports Fortran recently by flang. We are evaluating the quality of code generated by both of the compilers with collaboration of Kyoto University. Since these compilers are still immature, we give several feedbacks by examining the generated code. And our team is carrying out several collaborations with ARM compiler team on LLVM. In 2017, we have proposed the extension of OpenMP SIMD directive for SVE in the collaboration with Arm's researchers [4].

23.4 International Collaborations

23.4.1 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 23.1: 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.

23.4.2 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

23.5 Schedule and Future Plan

As shown in Figure 23.1, 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.

23.6 Publications

23.6.1 presentation and poster

[1] Hisashi Yashiro, Koji Terasaki, Takemasa Miyoshi, and Hirofumi Tomita: “Towards an extreme scale global data assimilation on the post-K supercomputer: development of a throughput-aware framework for ensemble data assimilation”, The 1st JpGU-AGU joint meeting, Makuhari, Japan, May 2017.

[2] Hisashi Yashiro: “Recent Extreme-Scale Simulation Efforts for NICAM in BoF ”Cloud Resolving Global Earth-System Models: HPC at its Extreme” ISC High Performance 2017. Frankfurt, Germany, Jun. 2017.
[3] Yoshifumi Nakamura, Y. Kuramashi, S. Takeda: “Critical endline of the finite temperature phase transition for $2+1$ flavor QCD away from the SU(3)-flavor symmetric point” Lattice 2017, Granada, Spain, Jun. 2017.
[4] Yoshifumi Nakamura: “Lattice QCD with pseudo-fermionparallelization” 7th JLESC Workshop, Urbana, USA, Jul. 2017.
[5] Yukio Kawashima: “Toward path integral molecular dynamics simulation of biomolecules” 3rd Japan-Thai workshop on Theoretical and Computational Chemistry 2017, Yokohama, Japan, Jul. 2017.
[6] Kiyoshi Kumahata, Kazuo Minami, Yoshinobu Yamade, Chisachi Kato: “Performance improvement of the general-purpose CFD code FrontFlow/blue on the K computer” HPC Asia 2018, Tokyo, Japan, Jan. 2018.
[7] Y. Nakamura: “Investigating Columbia plot with clover fermions” XQCD 2017, Pisa, Italy, Jun. 2017.
[8] Y. Kawashima, K. Hirao: “Long-Range Corrected Density Functional Theory with Periodic Boundary Condition”, The 11th Annual Meeting of Japan Society for Molecular Science, Sendai, Japan, Sep. 2017.

23.6.2 Articles and Technical Paper

[1] Y. Kawashima, K. Hirao: “”Singularity Correction for Long-Range-Corrected Density Functional Theory with Plane-Wave Basis Sets” The Journal of Physical Chemistry A, Vol. 121, No. 9, P. 2035-2045, 2017
[2] Yuetsu Kodama, Tetsuya Odajima, Motohiko Matsuda, Miwako Tsuji, Jinpil Lee, Mitsuhisa Sato. “Preliminary Performance Evaluation of Application Kernels using ARM SVE with Multiple Vector Lengths,” Re-Emergence of Vector Architectures Workshop (Rev-A), HI, USA, Sep. 2017.
[3] Miwako Tsuji, William Kramer, Mitsuhisa Sato. “A Performance Projection of mini-Applications onto Benchmarks toward the Performance Projection of real-Applications,” Workshop on Representative Applications (WRAp), HI, USA, Sep. 2017.
[4] Jinpil Lee, Francesco Petrogalli, Graham Hunter, Mitsuhisa Sato. “Extending OpenMP SIMD support for target specific code and application to ARM SVE,” 13th International Workshop on OpenMP, NY, USA, Sep. 2017.