Chapter 24

Flagship 2020 Project

24.1 Members

Primary members are only listed.

24.1.1 System Software Development Team

Yutaka Ishikawa (Team Leader)

Masamichi Takagi (Senior Scientist)

Atsushi Hori (Senior Scientist)

Balazs Gerofi (Senior Scientist)

Takahiro Ogura (Research & Development Scientist)

Fumiyoshi Shoji (Research & Development Scientist)

Atsuya Uno (Research & Development Scientist)

Toshiyuki Tsukamoto (Research & Development Scientist)

Toyohisa Kameyama (Technical Staff I)

Jie Yien (Technical Staff I)

24.1.2 Architecture Development Team

Mitsuhisa Sato (Team Leader)

Yuetsu Kodama (Senior Scientist)

Miwako Tsuji (Research Scientist)

Masahiro Nakao (Research Scientist)

Jinpil Lee (Research Scientist)

Yutaka Maruyama (Research Scientist)

Tetsuya Odajima (Postdoctoral Researcher)

Hitoshi Murai (Technical Scientist)

Motohiko Matsuda (Technical Scientist)

Itaru Kitayama (Technical Staff)

Toshiyuki Imamura (Research Scientist)

Kentaro Sano (Research Scientist)

Table 24.1: Development Teams

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

24.1.3 Application Development

Hirofumi Tomita (Team Leader)

Yoshifumi Nakamura (Research Scientist)

Soichiro Suzuki (Research & Development Scientist)

Kazunori Mikami (Research & Development Scientist)

Kiyoshi Kumahata (Research & Development Scientist)

Mamiko Hata (Technical Staff I)

Hiroshi Ueda (Research Scientist)

Naoki Yoshioka (Research Scientist)

Yiyu Tan (Research Scientist)

24.1.4 Co-Design

Junichiro Makino (Team Leader)

Masaki Iwasawa (Research Scientist)

Daisuke Namekata (Postdoctoral Researcher)

Kentaro Nomura (Research Associate)

Miyuki Tsubouchi (Technical Staff)

24.2 Overview of Research Activities

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, Fugaku supercomputer, 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 is in charge of co-design of the Fugaku supercomputer 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 Fugaku supercomputer by a broad community of users. Under the co-design concept, RIKEN and the selected institutions are expected to collaborate closely. The official name of the Fugaku supercomputer was decided in May 2019. It is now Fugaku.

As shown in Table 24.1, four development teams are working on Fugaku supercomputer system development with the FLAGSHIP 2020 Planning and Coordination Office that supports development activities. The primary members are listed in Section 24.1.

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

The Architecture Development team designs the architecture of the Fugaku supercomputer in cooperation with Fujitsu. And, the team designs and develops a productive programming language, called XcalableMP (XMP), and its compiler, and 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 Fugaku 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.

24.3 Target of System Development and Achievements in FY2019

The Fugaku'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.

The nine priority issues successfully finished at the end of March 2019. Since this fiscal year was the last of the priority projects, we have concentrated on more accurate performance estimation regarding the nine target application, using several nodes of Fugaku. Collaborating with compiler and system software groups, we conducted the tunning and improvement of the application algorithm. We confirmed that almost all of the nine target applications would obtain their target performance within the power cap.

In FY2019, the installation of Fugaku hardware started in December. Fujitsu continued to implement and test the Fugaku software stack. The major components system software is summarized as follows:

- Highly productive programming language, XcalableMP
  XcalableMP (XMP) is a directive-based PGAS language for large scale distributed memory systems that combine 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. We finished the front-end for Fortran 2008 Standard for Omni XcalableMP compiler. In 2019, we have performed the performance evaluation using the test system of Fugaku. The results are reported in Section 2.3.1. We performed the performance improvement of the communication runtime using Tofu-D. We are still working on C++ Front-end based on LLVM clang. And, we are working on the research for XcalableMP 2.0 which newly supports task-parallelism with the integration of PGAS models for distributed memory environment.

- 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 Fugaku supercomputer. Achievements of our MPI implementation have been described in Section 1.3.1.

- New file I/O middleware
  The Fugaku supercomputer 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 Fugaku supercomputer environment. The LLIO middleware, 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 was completed in FY2019.

- 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. 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 developed.

- 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. The detailed achievement has been described in Section 1.3.2.

- 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 FY2019, McKernel was ported to Arm architecture machines, Marvell's ThunderX-2 system and the Fugaku prototype system. Section 1.3.3 describes this effort.

The architecture development team carries out the research on co-design tools as well as the design of the Fugaku supercomputer:

GEM5 processor simulator for the Fugaku processor
We are developing a cycle-level processor simulator for the Fugaku 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 Fugaku hardware parameters. It enables the cycle-level performance evaluation of application kernels. In 2019, we continued the service to provide "Fugaku performance evaluation environment" including this simulator for performance evaluation and tuning by potential Fugaku users. We investigated the adjustment of parameters and performance with Fujitsu-in-house processor simulator and a test chip of A64FX for more accurate performance evaluation. The result are described in Section 2.3.2.

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 2020 (FY2019), we organized the mini-symposium in SIAM PP20 titled "Meaningful Performance Indicators for Scientific Computing" and presented our SSSP study for the dissemination of our proposed metrics.

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 have been evaluating the quality of code generated by both compilers with collaboration of Kyoto University. Since these compilers are still immature, we gave several feedbacks by examining the generated code. And our team carried out several collaborations with ARM compiler team on LLVM.

24.4 International Collaborations

24.4.1 DOE/MEXT Collaboration

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

- 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.

- 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.

- 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.

24.4.2 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 selected:

- 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

While continuing these topics in 2019, two new topics of particular relevance emerged from the evolving HPC landscape:

- big data and AI became increasingly important in many sectors and applications. The combination of big data and AI with numerical simulation and HPC became increasingly important
- ARM-based processors made significant breakthroughs in the HPC ecosystem, in particular with the Fujitsu A64FX processor co-designed by RIKEN, arriving in 2019.

24.5 Schedule and Future Plan

A prototype implementation and its basic evaluation were completed in FY2018. The hardware will be installed from the end of 2019 and the development of the system software, conducted by Fujitsu, will be finished in September 2020. The service is expected to start public operation in 2021.

24.6 Publications

24.6.1 Articles/Journal

24.6.2 Conference Papers

24.6.3 Posters

24.6.4 Invited Talks

[1] Yutaka Ishikawa, "An Overview of Fugaku Supercomputer," Russian Supercomputing Days 2019, Sep.24, 2019.
[2] Mitsuhisa Sato, Arm-SVE enabled post-K processor for for energy-efficiency and sustained application performance, Linaro Connect Bangkok, Tuesday, April 2, 2019.
[3] Mitsuhisa Sato, The post-K system and ARM-SVE enabled A64FX processor or energy-efficiency and sustained application performance, IHPCF 2019, Chengdu, China, 17 May, 2019.
[4] Mitsuhisa Sato, FLAGSHIP 2020 project and Supercomputer Fugaku, Post-K activities sessi in ISC 2019, Frankfurt, 17th June 2019.
[5] Mitsuhisa Sato, Co-design for Post -K, ARM for HPC Co-design Opportunities, ISC 2018 BOF, June 25, 201
[6] Mitsuhisa Sato, The post-K project and Fujitsu ARM-SVE enabled A64FX processor, VCEW 2019, Vail CO, USA, 24th June 2019.
[7] Mitsuhisa Sato, Update of Fugaku and FLAGSHIP 2020 project, FTJI Meing, Kashiwa,T, 28 Oct 20.
[8] Mitsuhisa Sato, What We Did for Co-Design in Development of Fugaku, ORAP Forum Paris, France,November 29, 2019.
[9] Mitsuhisa Sato, Co-Design for Fugaku. HPCAsia 2020@ Fukuoka, Jap, , Jan 17th 2019

24.6.5 Oral Talks

[1] Yoshifumi Nakamura, Y. Kuramashi, H. Ohno, S. Takeda: Critical endpoints of the finite temperature QCD, Frontiers in Lattice QCD and related topics, Kyoto, Japan, Apr. 2019

[2] Yoshifumi Nakamura, Y. Kuramashi, H. Ohno, S. Takeda: Critical endpoint in the continuum limit and critical endline at $N_t = 6$ of the finite temperature phase transition of QCD with clover fermions, the 37th international conference on lattice field theory, Wuhan, China, Jun. 2019
[3] Yoshifumi Nakamura: Towards computing the standard model of particle physics by tensor renormalization group, Tensor Network States: Algorithms and Applications (TNSAA) 2019-2020, Taipei, Taiwan, Dec. 2019
[4] Yoshifumi Nakamura: QCD Wide SIMD Library (QWS) for Fugaku, Fugaku QCD coding workshop, Kobe, Japan, Dec. 2019

24.6.6 Software

24.6.7 Patents