【機密性1（公開）資料】
[Confidentiality class 1 (Public) document]

MulticoreWorld 2026:
(February 16-20, 2026)

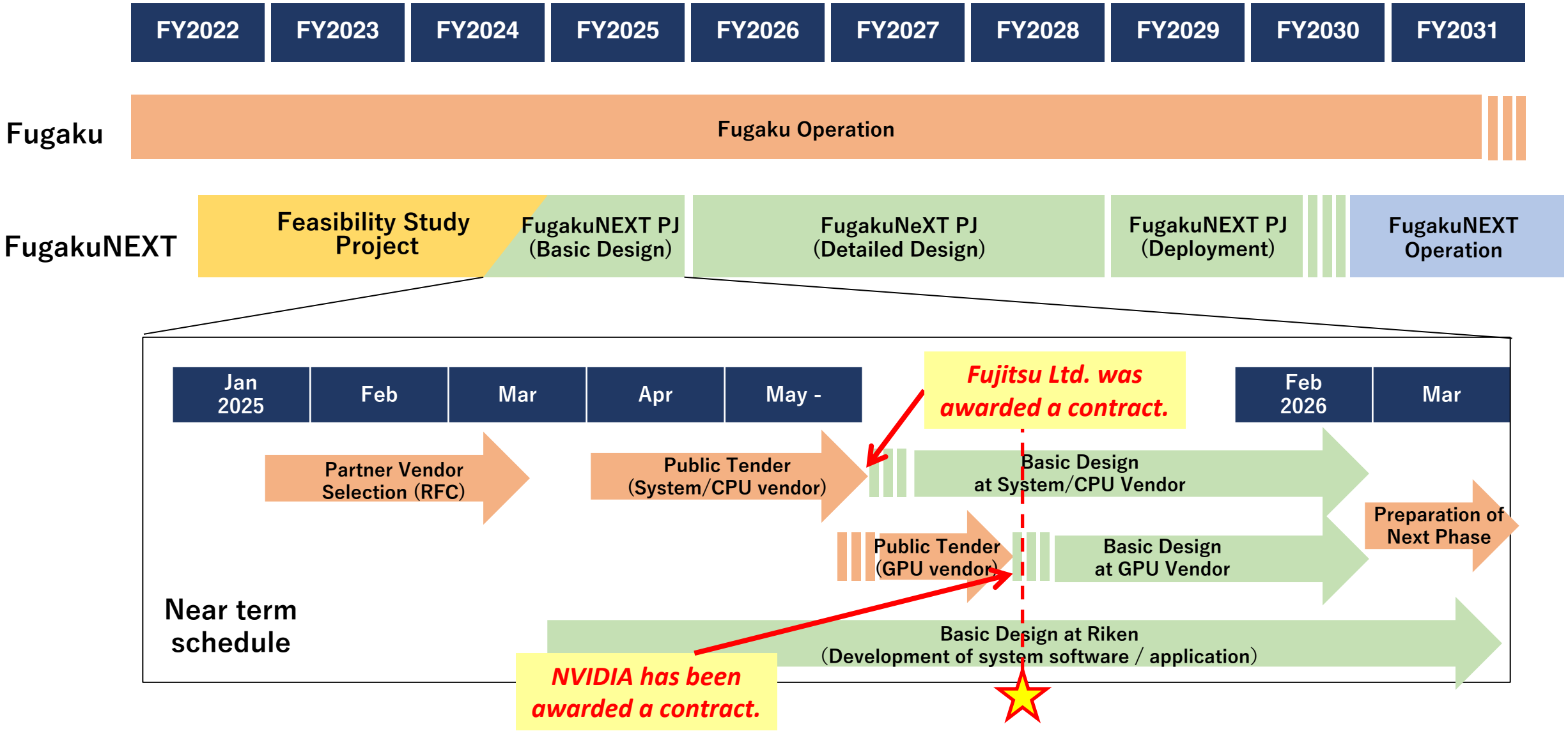
System Software Solutions for FugakuNEXT and Beyond

Kento Sato

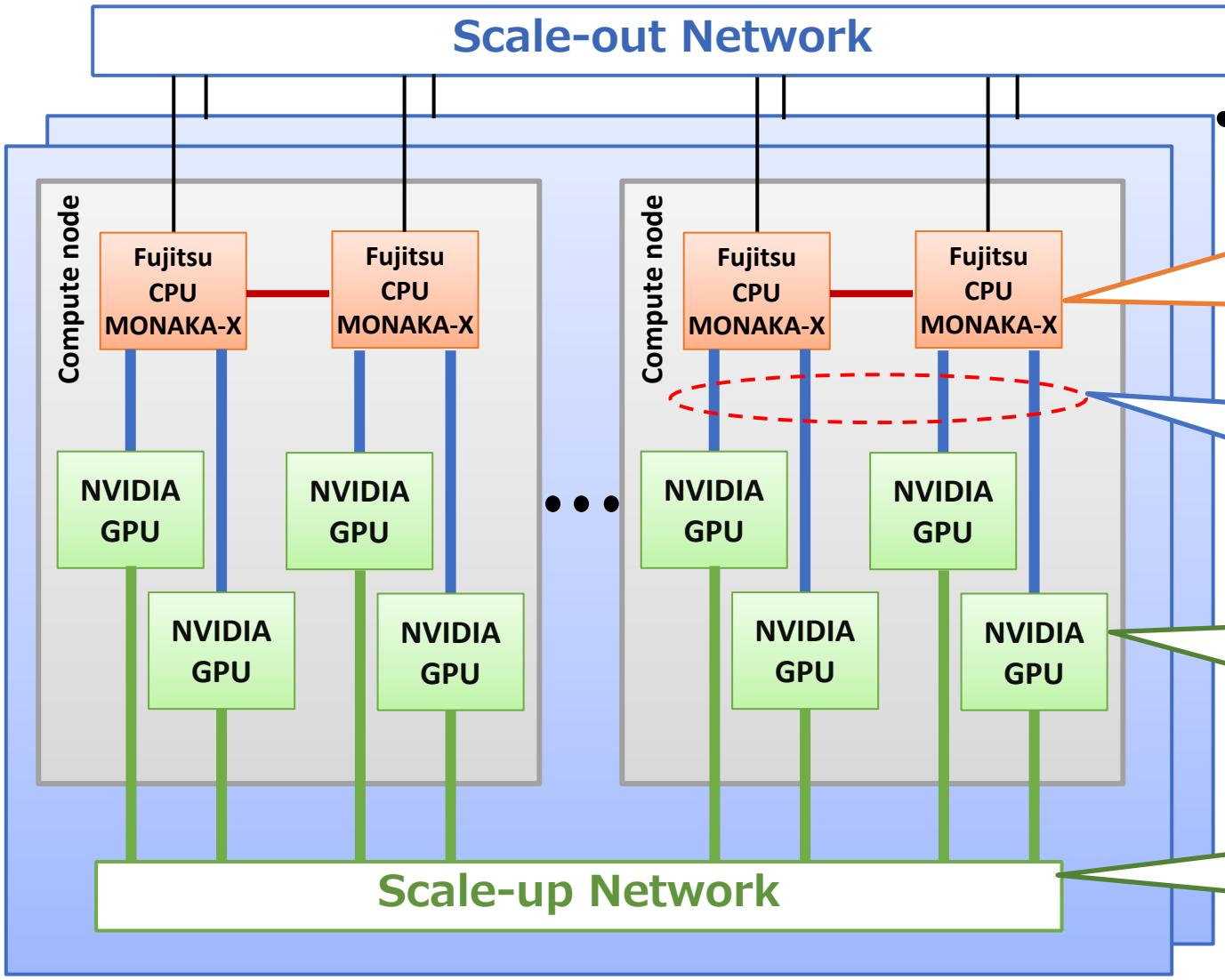
- Team Principal, High Performance Big Data Research Team, RIKEN R-CCS
- Unit leader, Data management platform development unit, AI for Science division, RIKEN R-CCS
- **Unit leader, Advanced HPC Technologies Development Unit, Next-Generation HPC Infrastructure Development Division, RIKEN R-CCS**
- Visiting professor, Kobe graduate university
- Visiting associate professor, Tohoku graduate university
- Adjunct lecturer, Osaka graduate university

v1.7

Expected Timeline of Fugaku-NEXT R&D



Overview of FugakuNEXT System Architecture



Entity in charge

- **Fujitsu MONAKA-X CPU** — Fujitsu, RIKEN
  - Many-core CPU with ARM ISA using chiplets
  - Considering NPU based on ARM SME (low-precision matrix computing unit)
- **CPU-GPU connection** — Fujitsu, NVIDIA, RIKEN
  - Considering high-bandwidth, coherent CPU-GPU connection technology
- **NVIDIA SoTA GPU** — NVIDIA, RIKEN
  - Highly performant (vector HPC, matrix AI)
  - Considering advanced stacked memory
- **Scale-up network for GPUs** — NVIDIA, RIKEN
  - Considering high-bw interconnects for multiple GPUs in future-generation NVLink

Detailed configurations and specs. will be determined in Basic design phase.

System Performance to be Targeted for FugakuNEXT

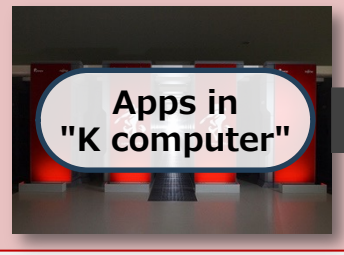
Goal / Concept of FugakuNEXT

- **5 to 10x higher effective performance** than Fugaku for existing HPC applications
- **Effective performance of 50 EFLOPS or higher for AI processing** with Zetta-FLOPS-scale peak performance in mind
- **Target 10-100x improvement for apps** by combining simulation and AI

- **"Application First" philosophy** that prioritizes application performance – **"co-design"**
- **Heterogeneous architecture** combining power-efficient CPUs and bandwidth-oriented GPUs to achieve the goal under power constraint
- CPUs that **can utilize the assets such as application software** developed for "Fugaku"

Items	CPUs	GPUs	Fugaku	Improvements
Total node count	>= 3,400		158,976	
Theoretical peak perf. of FP64 vector	>= 48 PFLOPS	>= 3.0 EFLOPS	488-537 PFLOPS	x5.7~
Theoretical peak perf. of FP16/BF16 matrix	>= 1.5 EFLOPS	>= 150 EFLOPS	1.95-2.15 EFLOPS	x70.5~
Theoretical peak perf. of FP8 matrix	>= 3.0 ELOPS	>= 300 EFLOP	—	
(Sparsity considered FP8 matrix)	—	>= 600 EFLOPS	—	
Main memory capacity	>= 10 PiB	>= 10 PiB	4.85 PiB	x4.1~
Main memory bandwidth	>= 7 PB/s	>= 800 PB/s	163 PB/s	x4.9~
System power consumption	40 MW (compute node and storage)		about 30 MW	

Road to Achieving up to 100x Application Performance

	"K computer" (2011-)	Supercomputer "Fugaku" (2020-)	"FugakuNEXT" (2030-)
Application Perf.	 Apps in "K computer"	100x  Apps in "Fugaku"	App Perf. **Up to 100x** Speed up  Apps in "FugakuNEXT"
Hardware Perf. (FP64/Mem. BW)	10.6PFLOPS 5.64PB/s	Vendor contributions Approx.40x **442.0PFLOPS 163 PB/s**	II Vendor contributions **Up to 6x** **>2.6EFLOPS >800PB/s**
Software Perf. (incl. algorithm advancement)	1 (Baseline: "K" computer)	Led by RIKEN Approx. 3x **3**	X Led by RIKEN **Up to 20x** **Up to 60**
			• Surrogate models / PINNs (10-20x) • Mixed-precision arithmetic • Emulation of FP64 (e.g., Ozaki scheme) } (2-8x)
AI Hardware Perf. (FP16/FP8)	Unsupported	New **2.0 EFLOPS**	**Up to 300x** Vendor contribution **> 600 EFLOPS** (with FP8 sparsity)

System Software in Supercomputers

- System software refers to a set of software that maximizes the performance of computer hardware, streamlines the development of applications such as scientific simulation and AI, and also ease the operation of computers

System Software

Programming env.	A suite of software (compilers, frameworks, debuggers, profilers/tracers) to support program development for HPC. Necessary for fast and efficient application development.
Numerical Library	A set of programs (matrix operations, FFT, etc.) that have been pre-accelerated or optimized for the target hardware. By utilizing numerical calculation libraries, development costs can be reduced and hardware performance can be efficiently extracted.
AI Framework	A programming environment that supports the implementation of machine/deep learning. The AI framework enables fast learning and inference of AI models and effective use of computational resources.
Communication Library	A group of programs for high-speed transmission and reception of data between computation nodes. Essential for parallel execution of applications.
File system	Software for easy and efficient use of storage. Necessary to store, manage, and access data at high speed.
Scheduler	Software that manages the execution order of computation jobs and computation resource allocation. Appropriate scheduling allows many users to use computing resources fairly and efficiently.
OS / Container	(OS) hardware management infrastructure and (container) software for virtualization. Essential to enable the management of computing resources and the isolation of environments to allow different processes to run safely and efficiently.
System Management	A group of systems that take charge of monitoring, fault handling, and user management of computers, contributing to stable system operation and reduced maintenance costs.

Strategy for System Software Development in FugakuNEXT

We adopt OSS as much as possible for sustainable maintenance. We also release software developed in this project as OSS and contribute to the community. In addition, if necessary, we adopt commercial/proprietary software that are expected to have sustainable development/improvement

System Software

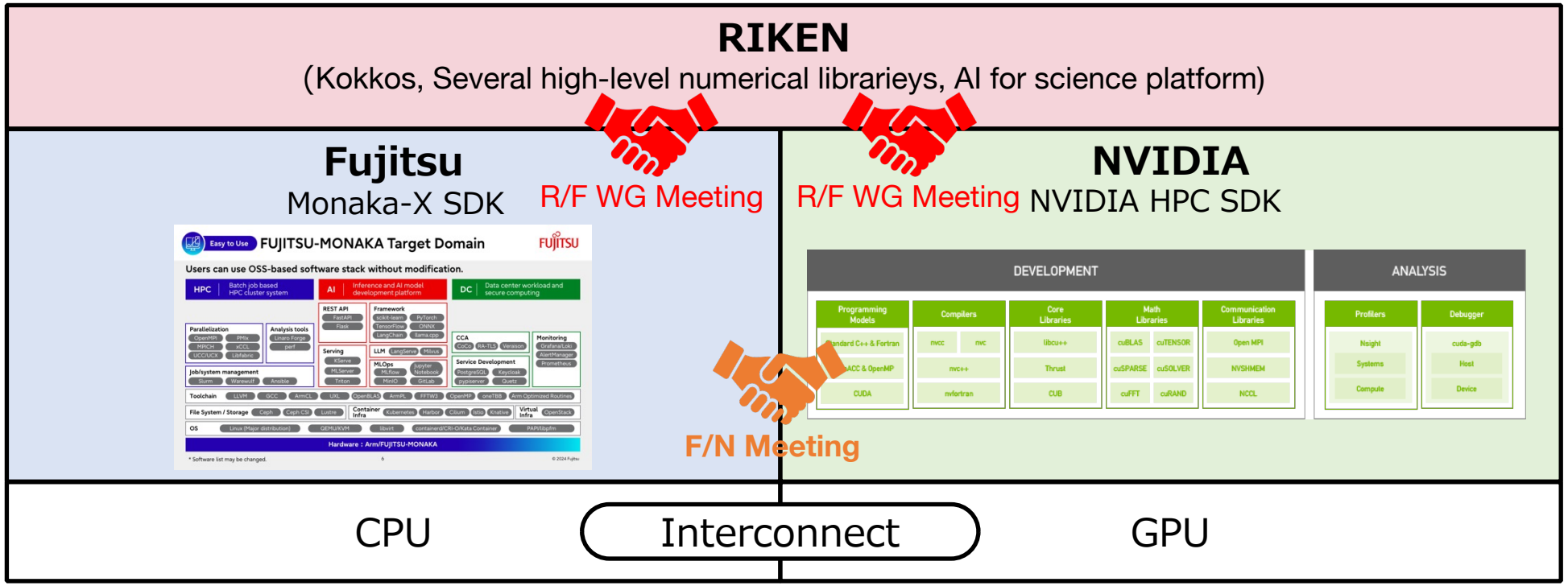
- Programming Env.
- Numerical Library
- AI Framework
- Communication Library
- File System
- Scheduler
- OS / Container
- System Management

- **High compatibility and continuous software improvement**
  - Based on lessens learned from Fugaku, we develop system software that can be continuously developed and improved through OSS
  - Build a software ecosystem by utilizing OSS in collaboration with domestic and international communities
  - Enables seamless transition from "Fugaku" to "FugakuNEXT" for application users via Fugaku OnDemand ("Cloudified" Fugaku)
- **Deployment to cloud and domestic/international computing platforms**
  - Provide software environment by packaging HPC environment via Spack/Container
  - Expand Virtual Fugaku ("Fugaku-fied" Cloud) environmer deploy a software environment on the cloud that provide equivalent environment to the FugakuNEXT system

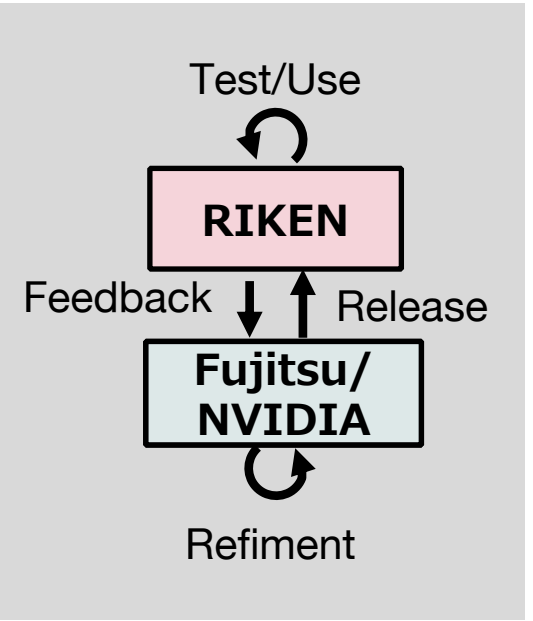
Fugaku Open OnDemand

Collaborations in System Software Development

- **Fujitsu/NVIDIA will develop SDK for Monaka-X/SToA GPU**
- **RIKEN will develop system software on top of Monaka-X/NVIDIA HPC SDK**
  - Kokkos, Several high-level numerical librarieys, Scale-up/out perfomrance analysis, AI for science platform (e.g., Multi-agent system, vibe coding platform)



Collaboration loops

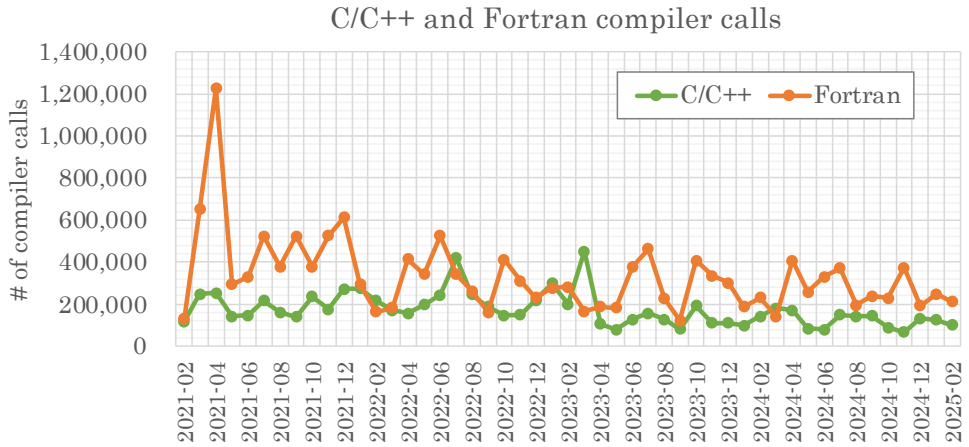


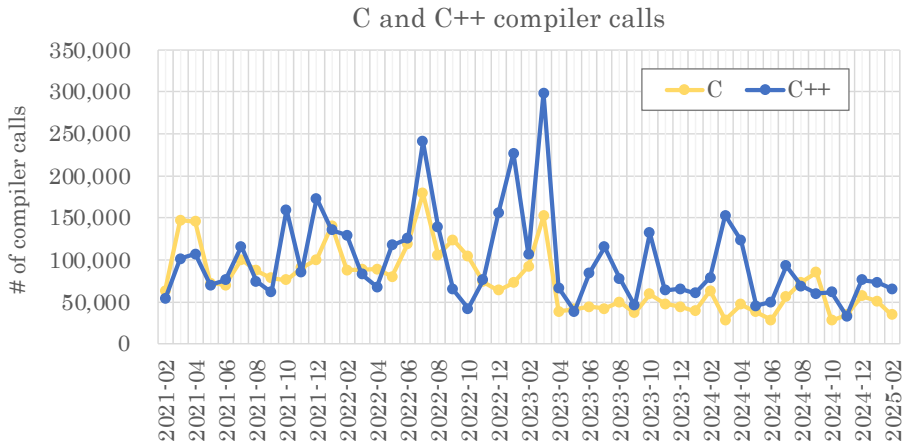
FugakuNEXT System Software Development Goal:
Programming Environment WG

Programming Environment WG

- Support the majority of existing HPC applications in Fugaku and programs implemented in the following languages and framework environments
  - Languages: C/C++, Fortran, Python (and others)
  - Frameworks: OpenACC, OpenMP
- Provide better development tool environment (e.g., debugger, profiler) than Fugaku
- Support frameworks to efficiently work across different CPUs/GPUs with high compatibility
  - Kokkos, StdPar, SYCL, etc.

Compiler Usage in Fugaku





FugakuNEXT System Software Development Goal:
Numerical Calculation Library and Middleware WG

Numerical Computing Library and Middleware WG

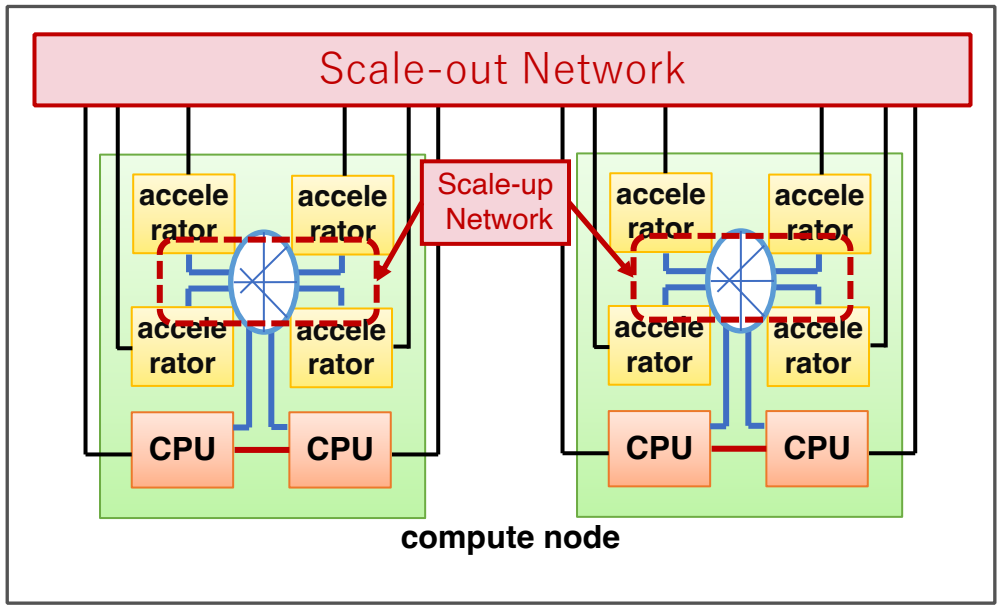
- Develop numerical libraries that efficiently use hardware and accelerate applications
  - Accelerate applications requiring high precision by utilizing low precision arithmetic units based on emulation techniques (e.g., OZAKI-Scheme).
  - If possible, we consider to apply emulation techniques to other cases other than matrix operations
- Middleware supported by Fugaku should also be supported by FugakuNEXT depending on the needs/seeds

Items	CPUs	Accelerators	Fugaku	Improvement
Total node count	>= 3,400		158,976	
Theoretical peak perf. of FP64 vector	>= 48 PFLOPS	>= 3.0 EFLOPS	488-537 PFLOPS	x5.7~
Theoretical peak perf. of FP16/BF16 matrix	>= 1.5 EFLOPS	>= 150 EFLOPS	1.95-2.15 EFLOPS	x70.5~
Theoretical peak perf. of FP8 matrix	>= 3.0 ELOPS	>= 300 EFLOP	—	
(Sparsity considered FP8 matrix)	—	>= 600 EFLOPS	—	
Main memory capacity	>= 10 PiB	>= 10 PiB	4.85 PiB	x4.1~
Main memory bandwidth	>= 7 PB/s	>= 800 PB/s	163 PB/s	x4.9~
System power consumption	<= 40 MW (compute node and storage)		about 30 MW	

FugakuNEXT System Software Development Goal:
Communication Library WG

Communication Library WG

- Support communication libraries deployed on Fugaku and also following libraries
  - MPI, UCX/UCC, etc.
- Develop system software to efficiently use hierarchical networks with scale-up (e.g., UALink, NVLink) and scale-out (e.g., Ultra Ethernet & InfiniBand) networks so that existing frameworks and applications can efficiently run in FugakuNEXT



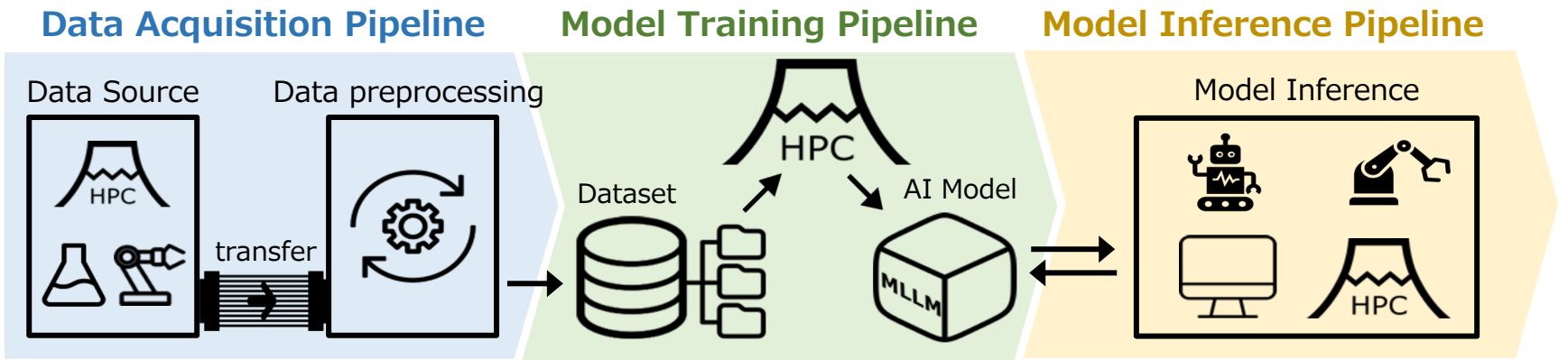
- Computational node with CPU and acceleration unit connected
  - CPU compatible with "Fugaku" at the binary level
  - Accelerated part of the arch based on GPUs with a proven track record of utilization in large scale supercomputers
- **System network suitable for both strong and weak scaling**
  - **Combined Scale-up/Scale-out network**
  - **Direct network link connection between accelerators**
- Tens of thousands of accelerator sockets throughout the system

FugakuNEXT System Software Development Goal:
AI Software WG

AI Software WG

- Support individual AI software which is widely used in 2029-30 and tune to efficiently run on FugakuNEXT
  - Including SoTA collective communication libraries: namely xCCL, including NCCL, RCCL, HCCL, oneCCL, MSCCL, ACCL
- Develop AI and data processing pipelines/workflows that seamlessly integrate individual AI tools to automate a wide range of AI-for-Science research tasks.

- **Acceleration of AI for Science by FugakuNEXT**
  - (1) Data acquisition pipeline: Fast generation of training data, fast transfer of various scientific data
  - (2) Model training pipeline: Large-scale pre-training, post-training, and fine-tuning of AI models
  - (3) Model inference pipeline: Automated workflow for scientific experiments by AI
- **"AI for Science" by integrating AI with simulations and scientific experiments**
  - Automation of generation of simulation program code, experiments, data analysis of results and proposal of the next steps
  - AI models are also self-reinforced to improve AI for Science research cycle



Continous benchmarking platform (Benchpark)

Continuous Benchmarking Platform (Benchpark)

R-CCS

Benchmark WG in App. Dev. Area

App. Dev. Area

Application Community

- Application Development Area
- HPCI Consortium
- CPU/GPU Vendors
- HAIRDESC
- Other domestic and international universities and research institutions

Benchmarks

Open CB Platform

Development history collection (application, system software)

Applications Evaluation

System Software Evaluation

App. Benchmarks

System Software Benchmarks
(including profilers and other performance analysis tools)

Multiple CPU/GPU benchmark environments

Env. 1 | Env. 2 | ••••••••• | Env. N

Source Code Repository (log)

Benchmark Results (log)

Automatic visualization

Benchmarks

System Software & Operations Area

Systems software community

- System Software Area & Operation Area
- CPU/GPU Vendors
- AI for Science Platform Division
- Quantum HPC Collaboration Platform Division
- Other domestic and international universities and research institutions

Architecture Area

AI for Science Div. or AI soft WG

AI (for Scinece) Factory

- **AI for general supports**: General application development and execution support with AI chatbots
- **Advanced development supports**: Program generation, code conversion, and auto-tuning by AI

Vibe coding Platform

Open evaluation results in publice

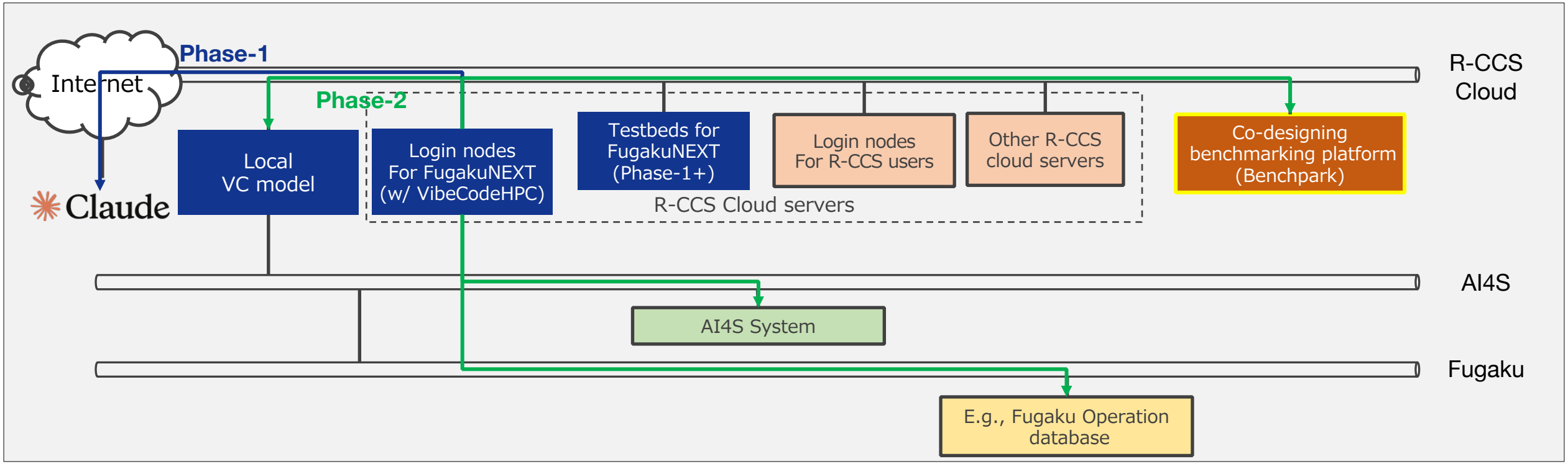
$N = N_1 \times N_2 \times N_3 \ldots$

- N_1 : CPU/GPU Environment
- N_2 : Problem Size
- N_3 : Configurations (compiler, library, version)
- …

Feedback of evaluation results to the app/system software community and architecture

Vibe coding Platform

Local genAI Platform for FugakuNEXT



- **Vibe Coding environment**
  - **Phase-1 genAI**: **Claude Code**
  - **Phase-2 genAI**: **Claude Code + Local Models**
    - Open-source or self-developed Vibe coding (VC) AI models in the local environment (e.g., Kimi, ChatHPC, Fortran2CPP etc.)
    - Integration to AI4S system
    - Connect with MCP servers (e.g., Fugaku operation database, **Open CB platform**)

AI-for-Science Supercomputer Overview [1]

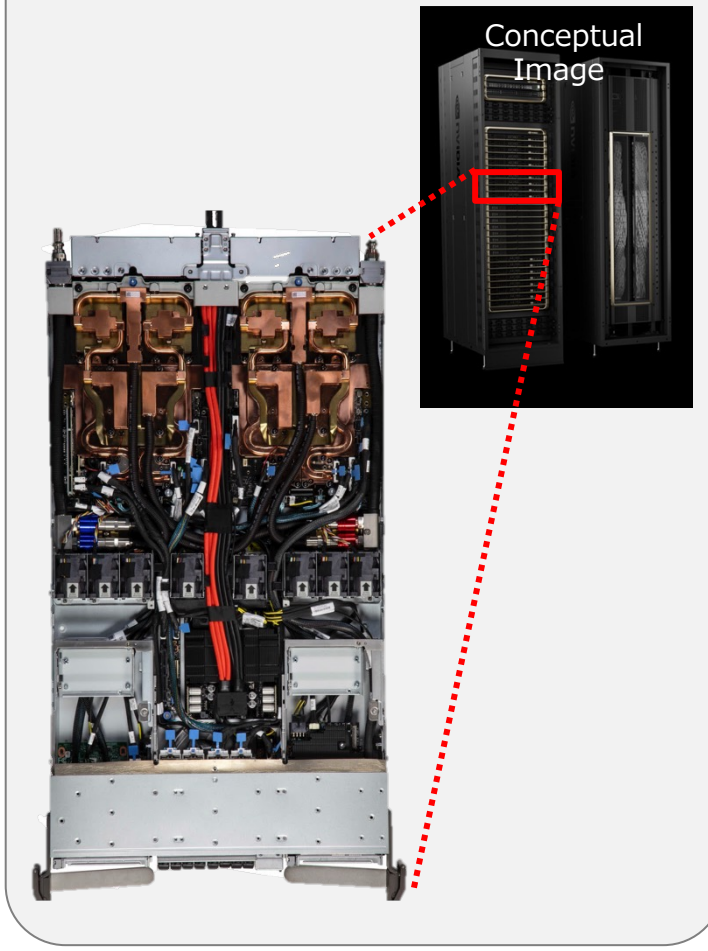
■ Background and Purpose
- New computational infra. to advance "AI for Science" through fusion of AI & ModSim
- Accelerating scientific research through world-leading AI performance and interplay with Fugaku
- Dramatically accelerate research cycles by scientific foundation models

■ System Configuration
- **Nodes**: 400 nodes equipped with NVIDIA Grace Blackwell superchips (over 1,600 GPUs)
- **Interconnect**: NVIDIA InfiniBand XDR (up to 3.2 Tbps)
- **Performance**: Over 64.16 PFLOPS (FP64), Over 15.539 EFLOPS (FP8)
- **Rack**: Supermicro servers with hot-water cooling capability adopted, achieving both high performance and energy efficiency

■ Comparison with "Fugaku"
- FP8 computational performance is approximately 7.23 times that of Fugaku's low-precision (FP16) unit
- Collaboration with "Fugaku," which excels at FP64, enables AI×Science fusion research

■ Future Plans
- Installation by fiscal year 2025, full-scale operation starting early fiscal year 2026
- Expected to develop and accelerate collaboration with institutions like the U.S. Argonne National Laboratory (MOU signed in 2024) and others

AI Inference Performance
15.539 ExaFLOPS or higher (FP8 performance)



[1] System Selected for AI for Science Development Supercomputer, https://www.riken.jp/pr/news/2025/20250728_1/index.html (July 28, 2025)

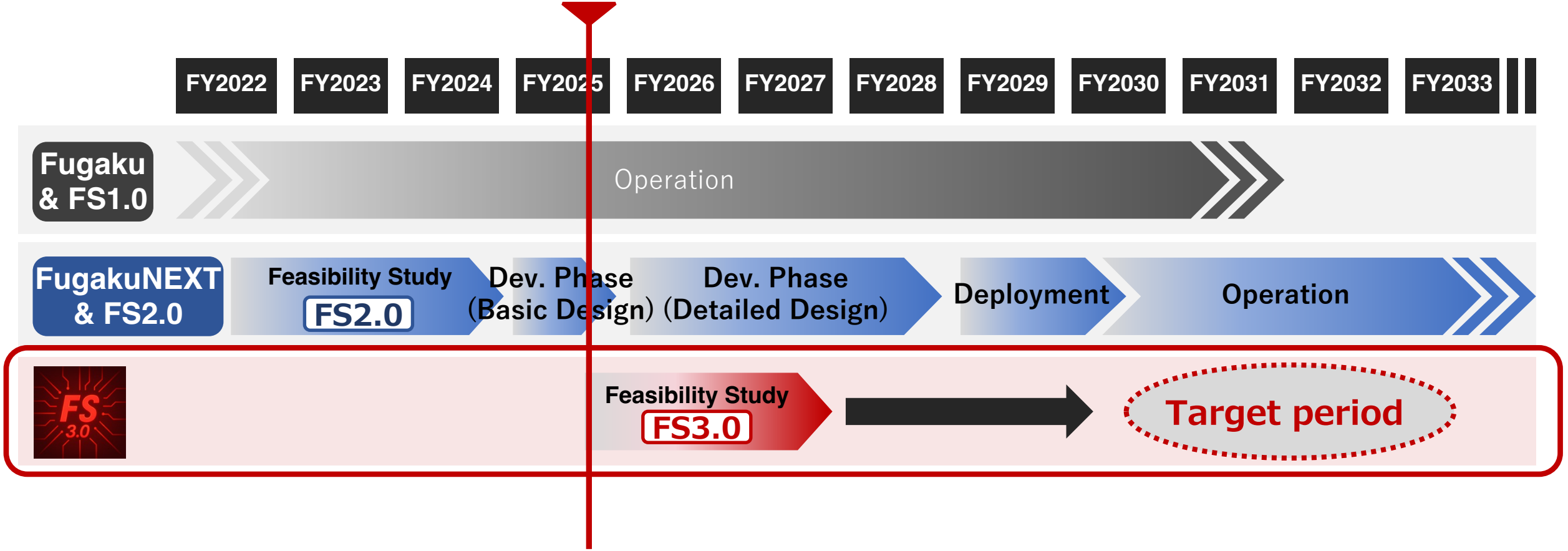
FS3.0 Web site

FS3.0: Feasibility Study on the future HPCI
(HPCI System Planning)

「HPCI整備計画調査研究」-- 運用システム整備計画調査研究

PI: Kento Sato, RIKEN R-CCS
Co-PI: Kazuhiko Komatsu, Tohoku University

Expected Timeline of FugakuNEXT and FS3.0



FS3.0: Overview (Lead organization: RIKEN R-CCS)

Background

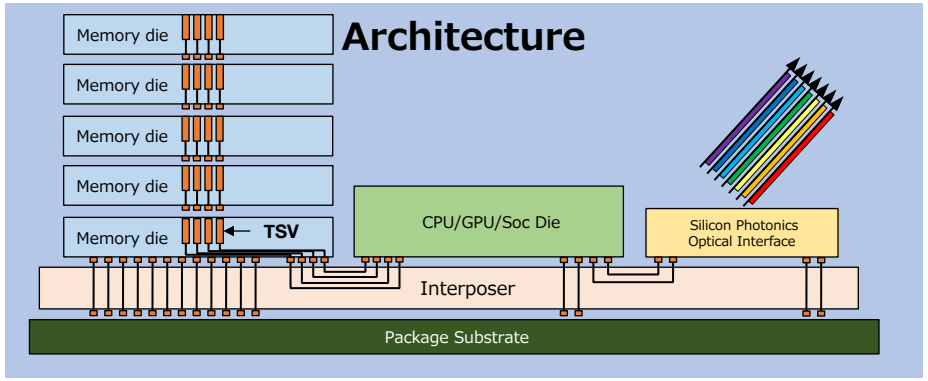
- The development of the next flagship system that will succeed the supercomputer Fugaku (hereinafter referred to as “FugakuNEXT”) and the enhancement of the HPCI (High Performance Computing Infrastructure for Innovation), which integrates diverse computational resources across Japan, have been progressing in parallel.
- In the FugakuNEXT era, the rapid advancement of generative AI will significantly increase the importance of AI for Science, and the computational resources required to support such workloads are becoming increasingly diverse.
- To meet these emerging requirements, the FS3.0 project conducts feasibility studies on computer architecture, system software, and next-generation application development, and proposes procurement/development plans for HPCI systems in the FugakuNEXT era and beyond. (FS1.0: Feasibility Study for Fugaku; FS2.0: Feasibility Study for FugakuNEXT; FS3.0: Feasibility Study for Systems Beyond FugakuNEXT.)

Feasibility Study Areas

- **Architecture Feasibility Study**
  - In collaboration with participating vendors, we investigate advanced technologies such as next-generation semiconductor technologies, 2.5D/3D/3.5D stacked packaging, and chip-to-chip optical interconnects, and explore a range of possible system configurations for HPCI systems in the FugakuNEXT era.
- **System Software Feasibility Study**
  - Building on the software ecosystem inherited from Fugaku and FugakuNEXT, we will define Japan’s system-software development positioning within the international community and examine system software that supports AI for Science in addition to traditional large-scale simulations.
- **Application Feasibility Study**
  - We investigate trends in innovative application use and development enabled by generative AI—such as automated computational science experiments and automatic code generation—across diverse scientific domains, and clarify the system requirements needed to enable such next-generation HPC applications.

Applications
enabled by AI and HPC computational resources

System software
bridging applications and next-generation architecture



FS3.0 Project Organization

FS3.0 Steering Group
RIKEN
GL: Kento Sato, GSL: Kazuhiko Komatsu, AD: Masaaki Kondo
Project Management Office
GL: Yoji Shimada, GSL: Naoki Nishi,
Mem: Hiroki Iwasaki, Tokuko Masui, Naoko Otani

GL/GSL: Group Leaders/Group Sub-Leaders
SGL/SGSL: Subgroup Leaders/Subgroup Sub-Leaders
AD: Advisor

Architecture Group
Tohoku University / Science Tokyo / RIKEN
GL: Kazuhiko Komatsu@Tohoku Univ.,
GSL: Ryohei Kobayashi@Science Tokyo, Masayuki Sato@Tohoku Univ.,
Shigeki Tomishima@RIKEN AD: Kentaro Sano@RIKEN

- **Architecture Sub-G1**
  Intel
  SGL: Joshua Fryman
- **Architecture Sub-G2**
  NVIDIA
  SGL: Dan Ernst
- **Architecture Sub-G3**
  HPE
  SGL: Fumiki Negishi
- **Architecture Sub-G4**
  Fujitsu
  SGL: Naoki Shinjo
- **Architecture Sub-G5**
  NEC
  SGL: Shintaro Momose

System Software Group
Osaka University
GL: Keichi Takahashi, GSL: Arata Endo,
AD: Kento Sato@RIKEN

- **HPC Programming & Communication Env. Sub-G**
  University of Tsukuba
  SGL: Norihisa Fujita@Univ. of Tsukuba,
  AD: Hitoshi Murai@RIKEN, Miwako Tsuji@RIKEN,
  Takeshi Nanri@Kyushu Univ.
- **Advanced Accelerator Programming Env. Sub-G**
  Meiji University
  SGL: Takaaki Miyajima
- **Numerical Library Sub-G**
  Hokkaido University
  SGL: Takeshi Fukaya , AD: Toshiyuki Imamura
- **Storage and File Systems Sub-G**
  University of Tsukuba
  SGL: Osamu Tatebe

Application Group
RIKEN
SGL: Rio Yokota,
GSL: Mohamed Wahib, Jens Domke,
AD: Takeshi Iwashita, Takeshi Shimokawabe

- **Next Gen. App. Needs Sub-G**
  University of Tokyo
  SGL: Yohei Miki ,
  SGSL: Takashi Shimokawabe
- **Next Gen. App. Dev. Methodology Sub-G**
  Nagoya University
  SGL: Tetsuya Hoshino

Research Group/Sub-Group

Operation Technology Advisory Group
RIKEN
AD: Keiji Yamamoto, Atsuko Takefusa

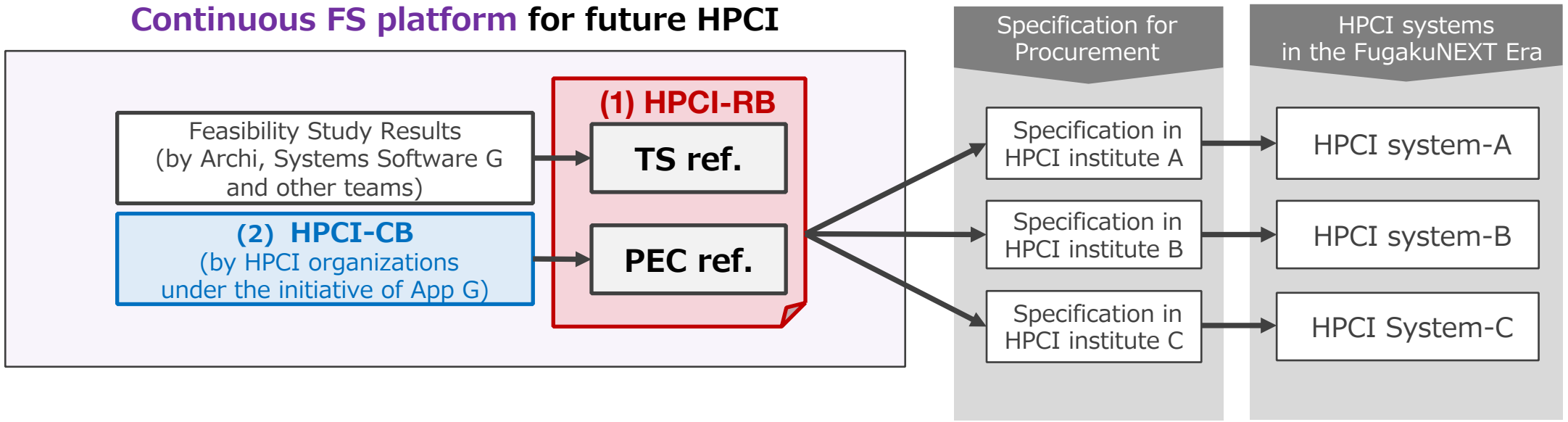
Quantum HPC Hybrid Advisory Group
Tohoku University
AD: Kazuhiko Komatsu

Strategic Advisory Group

V1.4 (As of Feb. 16, 2026)

Planned Deliverables: Continuous FS Platform for future HPCI

- **Continuous FS Platform for future HPCI**
  - Ensure long-term preservation and utilization of FS outcomes, including results and know-how from **HPCI-RB**/**HPCI-CB**
  - (1) **HPCI-RB**: HPCI Reference Blueprint
    - Reference describing system requirements for future HPCI systems based on feasibility studies
    - **TS ref. (Tech. Spec. Reference) :** Provide references for architecture and system software roadmaps
    - **PEC ref. (Performance Evaluation Criteria Reference):** Provide effective performance to be achieved by the target applications on systems proposed in TS ref.
  - (2) **HPCI-CB**: HPCI Continuous Benchmarking Env.
    - Enables continuous performance evaluation of target applications and performance prediction in future HW



R&D on AI for Science Platform

Kento Sato

- **Team Principal, High Performance Big Data Research Team, RIKEN R-CCS**
- **Unit Leader, Data management platform development unit, AI for Science division, RIKEN R-CCS**
- Unit Leader, Advanced HPC Technologies Development Unit, Next-Generation HPC Infrastructure Development Division, RIKEN R-CCS
- Visiting Professor, Kobe graduate university
- Visiting Associate Professor, Tohoku graduate university
- Adjunct lecturer, Osaka graduate university

Our mission: HPC, AI & Big Data Fusion

The mission of our team is to investigate state-of-the-art techniques for efficiently running large-scale applications on HPC systems. In addition to fundamental R&D of system software in HPC, our team conducts performance analysis and software development to accelerate deep learning and big data processing on HPC systems (HPC for AI/BD), while we also apply AI techniques to solve technical challenges in HPC (AI/BD for HPC)

HPC for AI/BD

Research and software development for accelerating AI/Big data applications by techniques in HPC

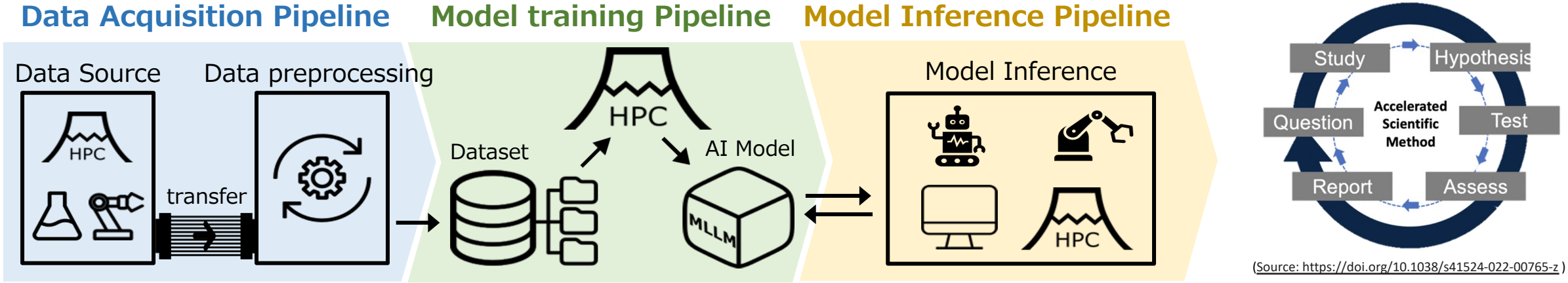
AI/BD for HPC

Research and software development for resolving technical challenges in HPC by AI/ Big Data techniques

Fundamental R&D for HPC

AI Platform in the FugakuNEXT era and beyond:
HPC/AI platform enabled by state-of-the-art system software technologies

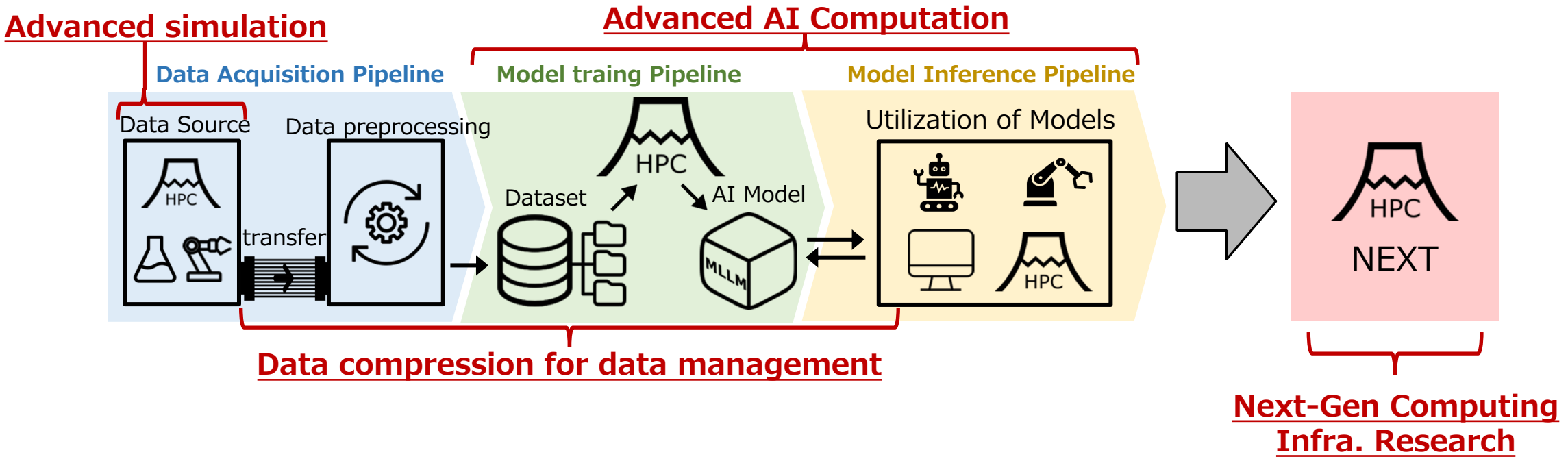
- **AI-for-Science Platform**
  - Develop AI and data processing pipelines that seamlessly integrate individual AI tools to automate a wide range of AI-for-Science research tasks
- **Acceleration of AI for Science**
  - **(1) Data acquisition pipeline**: Fast generation of training data, fast transfer of various scientific data measured and produced and management of the data
  - **(2) Model training pipeline**: Large-scale pre-training, post-training, and fine-tuning of AI models
  - **(3) Model inference pipeline**: Automated workflow for scientific experiments by AI



(Source: https://doi.org/10.1038/s41524-022-00765-z)

Automate research cycles:
program code generation, execution of program, data generation, data analysis on experimental results, proposal for the next steps

Research Plan and End-user software adoption strategy:
Fundamental/core technology research for AI for Science Platform



Research Plan and End-user software adoption strategy:
Fundamental/core technology research for AI for Science Platform

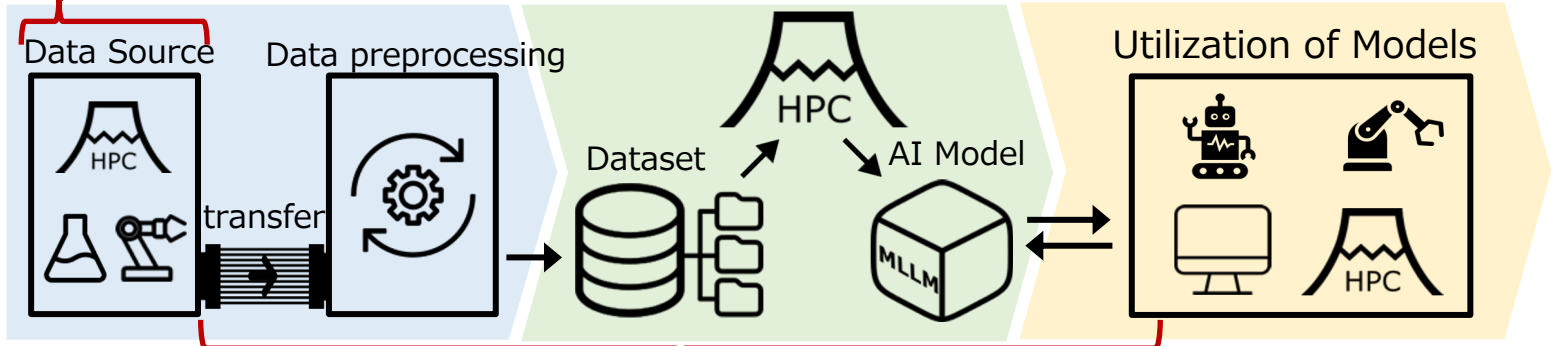
(T2) Data aggregation | (T2) MPI I/O | (T2) Debugging tools | (T2) AutoCheck

Advanced simulation

(T2) 3D parallelisms | (T2) C/R for AI | (T2) Data plancement

Advanced AI Computation

FugakuNEXT & FS3.0



Data compression for data management

(T1) Text data compression | (T1) TEZip enhamcement

(T1) Image data compression | (T1) XCT data compression

Next-Gen Computing Infra. Research

(T3) DSE

(T3) Software Usage Study

AI platform development and AI-assisted next-generation operations

(T1) AI Pipeline Dev. | (T3) Advanced Operaton by AI | (T2) Energy/Power Efficency Mechanism

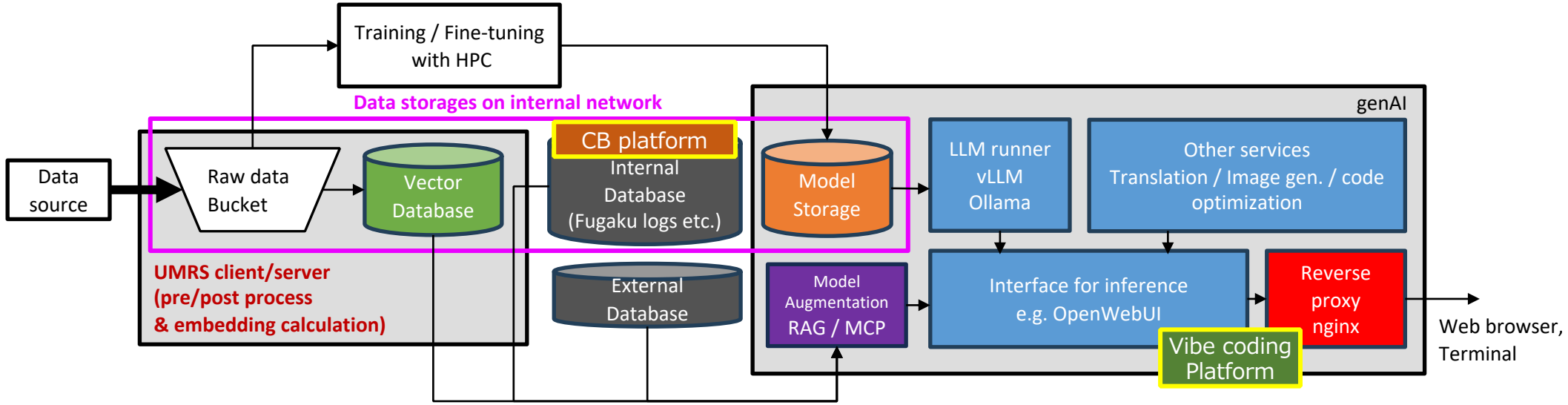
We aim to develop “**default-on**" software incorporated into the platform so that end-users adopt our software without consciously

Secured & Extendable AI Workflow Orchestration for AI4S

Masaru Nagaso[†1], Jens Domke[†1], Eliott Jacopin[†2], Emmanuel Jeannot[†1],
Kento Sato[†1], Kozo Nishida[†2]
†1: RIKEN R-CCS, †2: RIKEN BDR

WIP

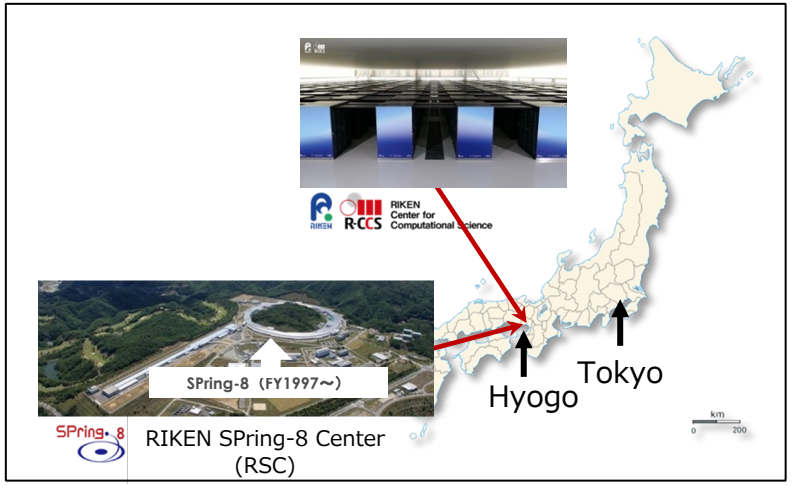
- **Background**
  - Scientific AI workflows span **data transfer, data management, model training, inference, and customized data processing & analysis**, requiring tight integration.
  - Manual handling slows research and increases risk of errors.
  - **Security is critical** in some use cases (e.g., handling pre-published or sensitive scientific data).
- **Approach: In-house AI for Science Plat form**
  - **Data pipeline**: Parallel, secure transfers with/without Globus.
  - **Storage & automation**: Leverage **UMRS** (Emmanuel Jeannot) for pre/post-processing, embeddings, and data management.
  - **Training & fine-tuning**: Managed execution with monitoring of progress and resources.
  - **GenAI** services (Jens Domke): Network-isolation of models and data for security, supporting inference, RAG, MCP servers, etc.
  - **Extendibility**: Users can deploy additional services into the workflow.
  - **Centralized management layer** to coordinate data feeding, customized pre/post-processing, training status, and inference deployment.
  - Built with **scalability, automation, and secure handling of sensitive datasets** as guiding principles.

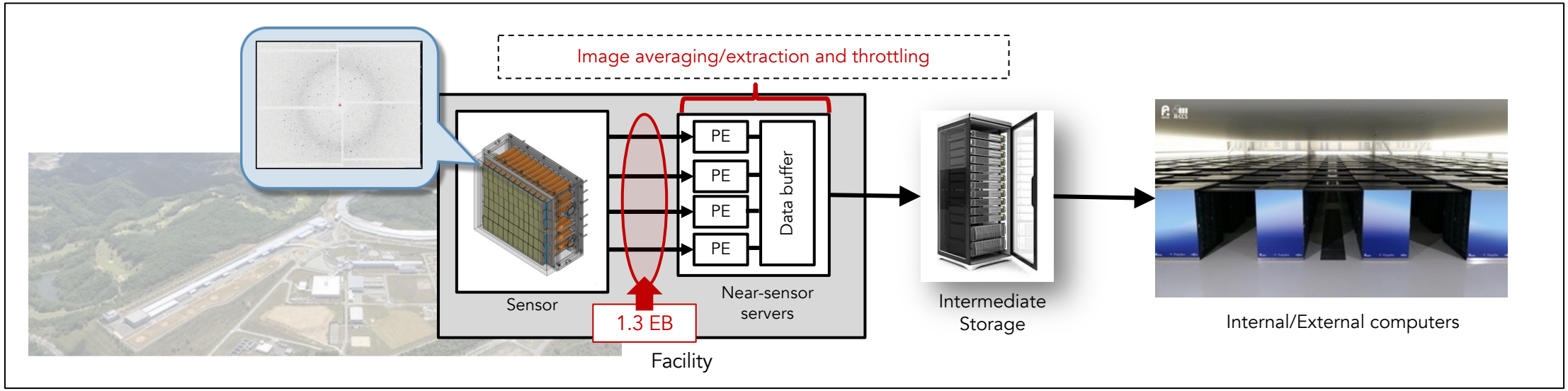


Compression of Time Evolutionary Image Data through Predictive Deep Neural Networks (IEEE/ACM CCGrid 2021 [15])

Rupak Roy[†1], Kento Sato[†2], Subhadeep Bhattacharya[†1], Xingang Fang[†1], Yasumasa Joti[†3], Takaki Hatsui[†4], Toshiyuki Hiraki[†4], Jian Guo[†2] and Weikuan Yu[†1]
†1: Florida State University, †2: RIKEN R-CCS, †3: JASRI, †4: RIKEN RSC,

- **Background:**
  - The new detector (CITIUS) in the SPring-8 Center (RSC) expected to generate about 100~400 PB per year
  - To analyze and/or train an AI model with the data, data transfer from the sensors to a large-scale computer is necessary
  - However, the transfer of large data becomes a performance bottleneck for AI pipeline
  - X-ray CT data is one of popular data in RSC. Scientists periodically take snapshots of a target specimen while rotating it (→ Time evolutionary data)



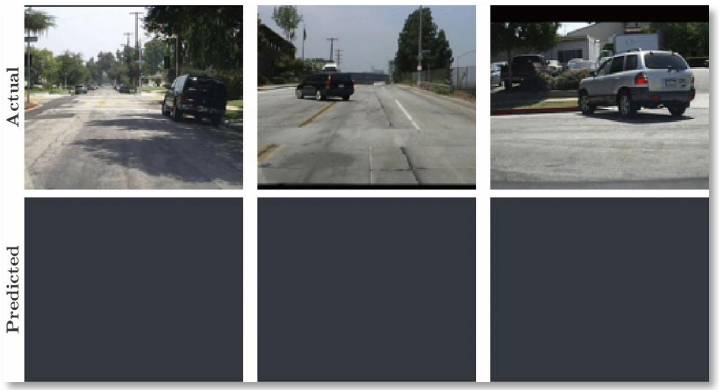


[15] Rupak Roy, Kento Sato, Subhadeep Bhattacharya, Xingang Fang, Yasumasa Joti, Takaki Hatsui, Toshiyuki Hiraki, Jian Guo and Weikuan Yu, "Compression of Time Evolutionary Image Data through Predictive Deep Neural Networks", In the proceedings of the 21 IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGrid 2021), May, 2021

Compression of Time Evolutionary Image Data through Predictive Deep Neural Networks (IEEE/ACM CCGrid 2021 [15])

Rupak Roy[†1], Kento Sato[†2], Subhadeep Bhattacharya[†1], Xingang Fang[†1], Yasumasa Joti[†3], Takaki Hatsui[†4], Toshiyuki Hiraki[†4], Jian Guo[†2] and Weikuan Yu[†1]
†1: Florida State University, †2: RIKEN R-CCS, †3: JASRI, †4: RIKEN RSC,

- **Key idea**
  - Prediction is one of key aspects in data compression (**Fig.1**)
  - With accurate prediction, target data can be converted data with sequence of zeros
- **Approach:**
  - We has been developing and enhancing an AI-based data compression tool (TEZip)
  - The AI model predicts or reconstruct target images and TEZip only store the delta values (**Fig.2**)
    - E.g.) PredNet: 1st image fame –[predict]-> 2nd, 3rd,… Nth image frame
    - B: Original image frames, P: Predicted image frames, D: Difference between B and P, C: Compressed image frames by a series of encoding
  - Pre-processing (by AI Training): Train a NN to learn the pattern of the movement of the specimen (**Fig.2**)
  - Data (De)compression (by AI Inference): Predict future images, compute delta and apply encoding/compressor (**Fig.2**)



PredNet (video from drive recorder)
https://coxlab.github.io/prednet/

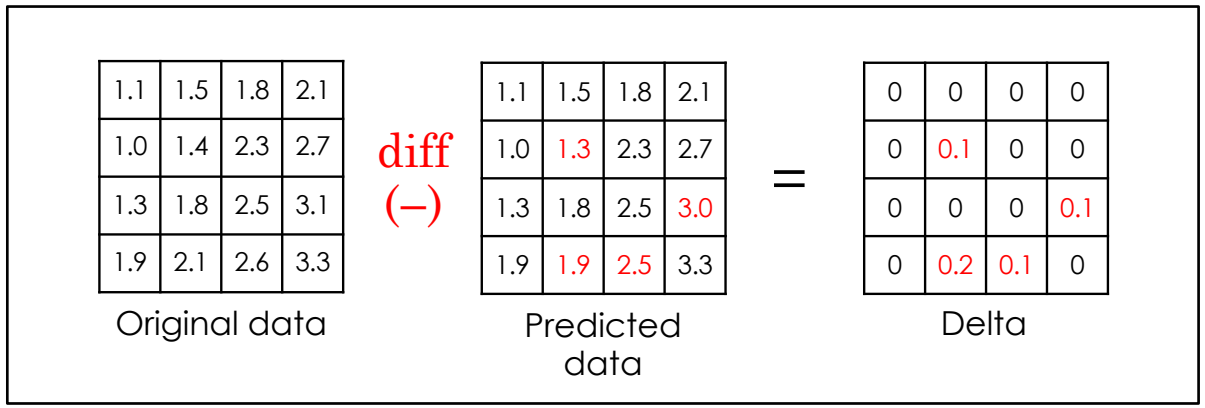


Figure 1

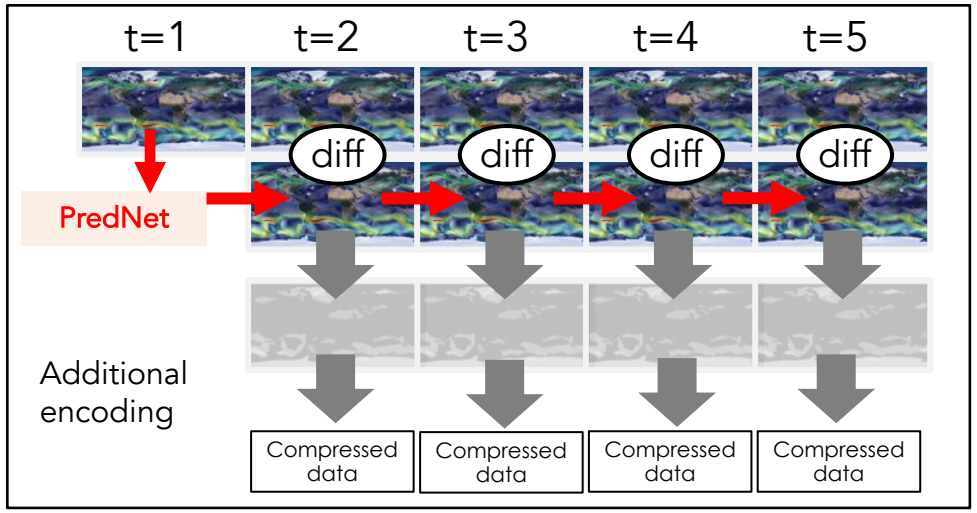


Figure 2

[15] Rupak Roy, Kento Sato, Subhadeep Bhattacharya, Xingang Fang, Yasumasa Joti, Takaki Hatsui, Toshiyuki Hiraki, Jian Guo and Weikuan Yu, "Compression of Time Evolutionary Image Data through Predictive Deep Neural Networks", In the proceedings of the 21 IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGrid 2021), May, 2021

Compression of Time Evolutionary Image Data through Predictive Deep Neural Networks (IEEE/ACM CCGrid 2021 [15])

Rupak Roy[†1], Kento Sato[†2], Subhadeep Bhattacharya[†1], Xingang Fang[†1], Yasumasa Joti[†3], Takaki Hatsui[†4], Toshiyuki Hiraki[†4], Jian Guo[†2] and Weikuan Yu[†1]
†1: Florida State University, †2: RIKEN R-CCS, †3: JASRI, †4: RIKEN RSC,

- **Results (Fig.8, 10):**
  - TEZip gives higher compression ratio than major compression tools (e.g., X.265, SZ)
  - Lossless mode: 9 to 15 / Lossy mode (w/ a few % of errors): 40 to 50

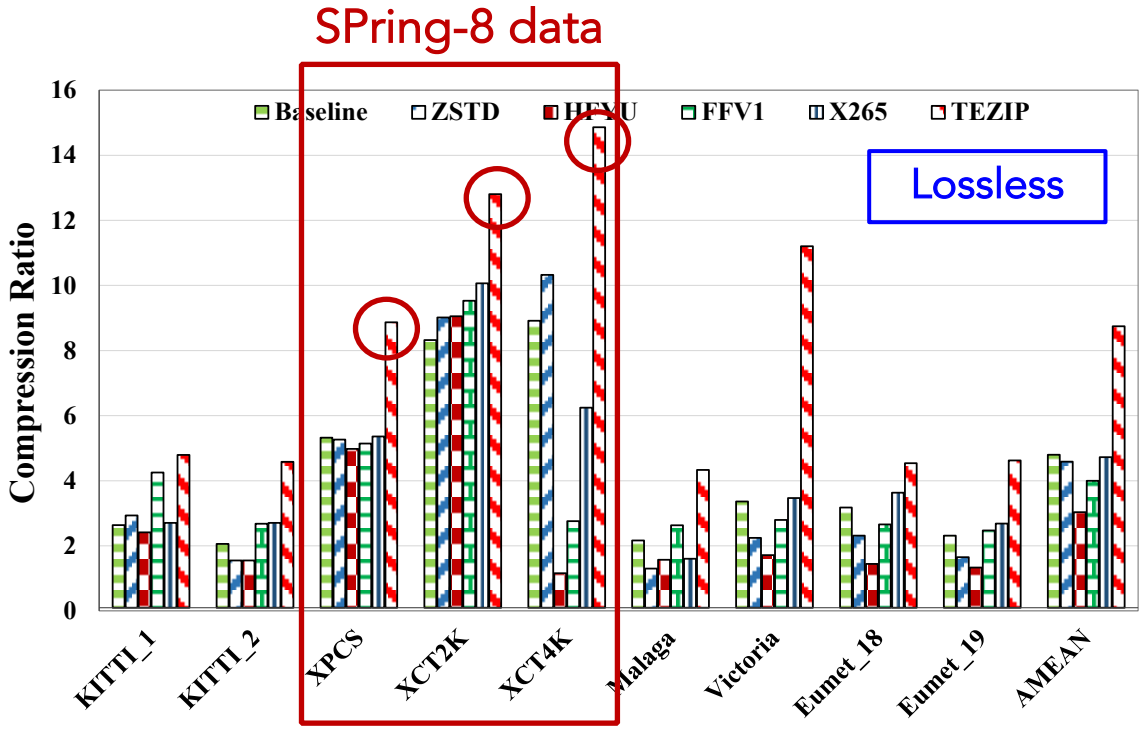


Fig. 8. Compression ratio with lossless compressors.

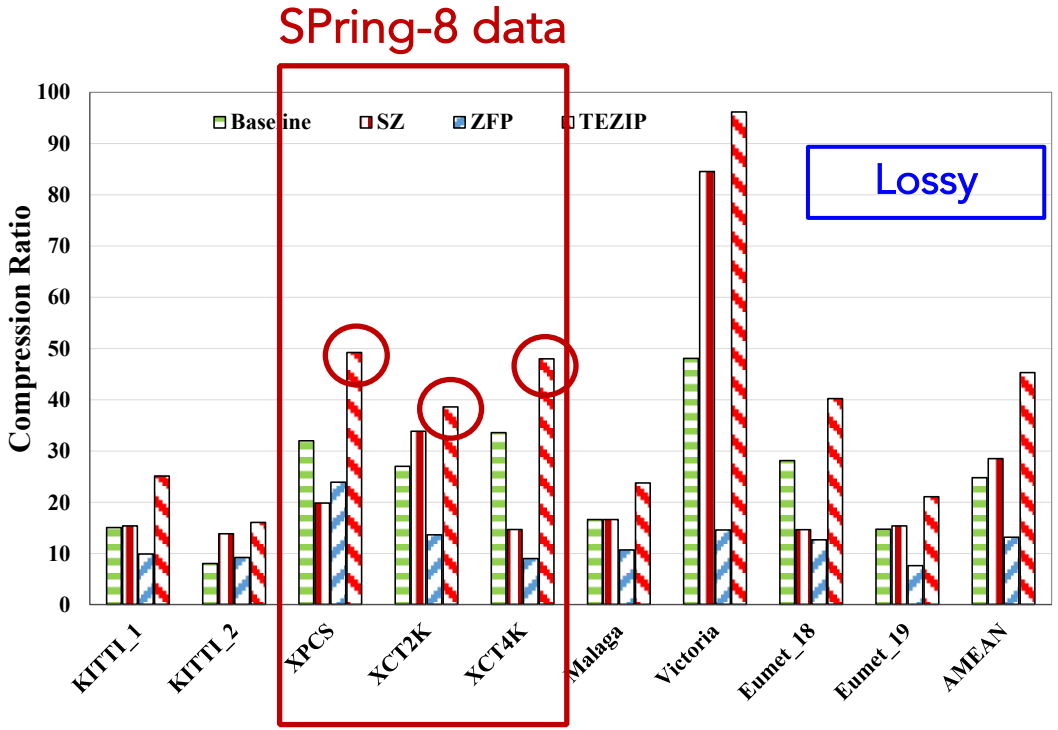


Fig. 10. Compression ratio with different lossy compressors

[15] Rupak Roy, Kento Sato, Subhadeep Bhattacharya, Xingang Fang, Yasumasa Joti, Takaki Hatsui, Toshiyuki Hiraki, Jian Guo and Weikuan Yu, "Compression of Time Evolutionary Image Data through Predictive Deep Neural Networks", In the proceedings of the 21 IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGrid 2021), May, 2021

System Software Utilization on an ARM-Based Supercomputer: Insights from a Production-Scale System (HPCAsia2026 [73])

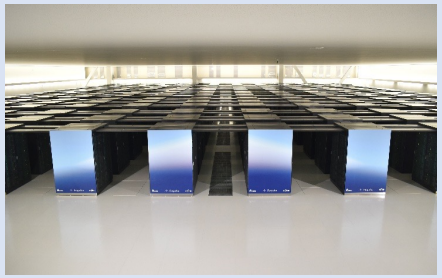
Yosuke Asai[†1], Kento Sato[†1], Keiji Yamamoto[†1], Hitoshi Murai[†1], Kohei Yoshida[†1,2]
†1: RIKEN R-CCS, †2: The University of Electro-Communications

- **Background:**
  - RIKEN is currently developing Japan's next flagship supercomputer, the successor to Fugaku, named "FugakuNEXT".
  - It is scheduled to begin operation around 2030, will continue to use Fujitsu MONAKA-X CPU and the NVIDIA’s state-of-the-art GPU
  - To prioritize system software development for FugakuNEXT, we need to understand usage of system software, library, compiler tool and other system software.
- **Approach:** Analize usage of **(i) shared libraries**, **(ii) Spack recipes** and **(iii) compilers**
  - **(i) Shared libraries**: App log produced when a user runs an MPI application
  - **(ii) Spack recipes**: Spack log produced when a user loads Spack package. (Spack: A software package management tool developed for HPC environments)
  - **(iii) Compilers**: License server log produced when a user invokes a Fujitsu compiler (C, C++, Fortran)

Period

Log Types	Start date	End date	Number of logs
App log	May 21, 2021	March 31, 2025	11,359,316 Jobs
Spack log	July 19, 2022	March 31, 2025	5,807,006 Jobs
Compiler log	Feb. 19, 2021	March 31, 2025	121,478 Calls

System software developoment: Fugaku to FugakuNEXT



Metrics and types of aggregated data

Log Types	Metrics	Aggregated
App log	Shared library, path of exec. app.	Usage count, node hour, num of user, num of user’s group (For the entire period and for each month)
Spack log	Spack package	Usage count, node hour, num of user, num of user’s group (For the entire period and for each month)
Compiler log	Language type, compilation options	Invocation count, invocation count as a linker (For the entire period and for each month)

[73] Yosuke Asai, Kento Sato, Keiji Yamamoto, Hitoshi Murai, and Kohei Yoshida. 2026. System Software Utilization on an ARM-Based Supercomputer: Insights from a Production-Scale System. In Proceedings of the Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region Workshops (SCA/HPCAsiaWS '26). Association for Computing Machinery, New York, NY, USA, 110–121. https://doi.org/10.1145/3784828.3785349

System Software Utilization on an ARM-Based Supercomputer: Insights from a Production-Scale System (HPCAsia2026 [73])

Yosuke Asai†1, Kento Sato†1, Keiji Yamamoto†1, Hitoshi Murai†1, Kohei Yoshida†1,2
†1: RIKEN R-CCS, †2: The University of Electro-Communications

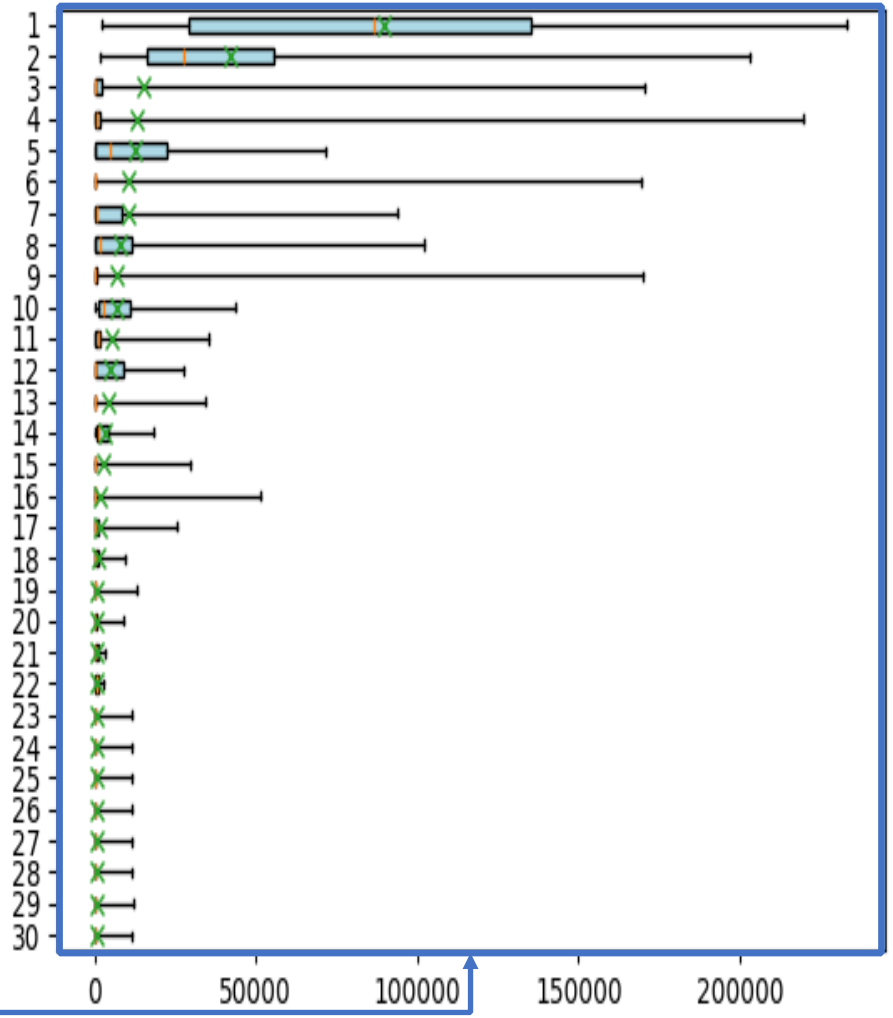
Spack log – package usage count

- **netCDF (for Fortran) is most used via Spack**
  - netcdf-fortran (ranked 1st) is used more than twice as frequently as the next-ranked (2nd) package
- **Many of MD-related application/libraries are used via Spack**
  - GROMACS, ABINIT-MP, Quantum ESPRESSO, LAMMPS, pymatgen etc.
- **Spack users use GROMACS and netCDF**
  - The rest of packages ranked below third place are relatively low usage

Usage counts for the entire period

Rank	Spack package	Count
1	netcdf-fortran	2,966,582
2	gromacs	1,397,102
3	py-pandas	496,051
4	py-pip	436,878
5	abinitmp	420,233
6	glew	356,763
7	fujitsu-mpi	350,796
8	quantum-espresso	266,086
9	lammps	229,913
10	py-numpy	226,151
11	fujitsu-fftw	175,070
12	py-pymatgen	164,821
13	libxc	142,440
14	hdf5	102,145
15	akaikkr	94,919
16	netlib-lapack	61,421
17	py-scipy	55,259
18	py-mpi4py	33,748
19	fftw	30,672
20	python	28,233
21	gcc	24,883
22	netcdf-c	24,856
23	py-pillow	24,540
24	py-python-dateutil	22,878
25	py-pytz	22,573
26	py-pyparsing	22,550
27	py-cycler	22,545
28	py-kiwisolver	22,542
29	openblas	22,428
30	paraview	20,049

Monthly usage counts



[73] Yosuke Asai, Kento Sato, Keiji Yamamoto, Hitoshi Murai, and Kohei Yoshida. 2026. System Software Utilization on an ARM-Based Supercomputer: Insights from a Production-Scale System. In Proceedings of the Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region Workshops (SCA/HPCAsiaWS '26). Association for Computing Machinery, New York, NY, USA, 110–121. https://doi.org/10.1145/3784828.3785349

System Software Utilization on an ARM-Based Supercomputer: Insights from a Production-Scale System (HPCAsia2026 [73])

Yosuke Asai†1, Kento Sato†1, Keiji Yamamoto†1, Hitoshi Murai†1, Kohei Yoshida†1,2
†1: RIKEN R-CCS, †2: The University of Electro-Communications

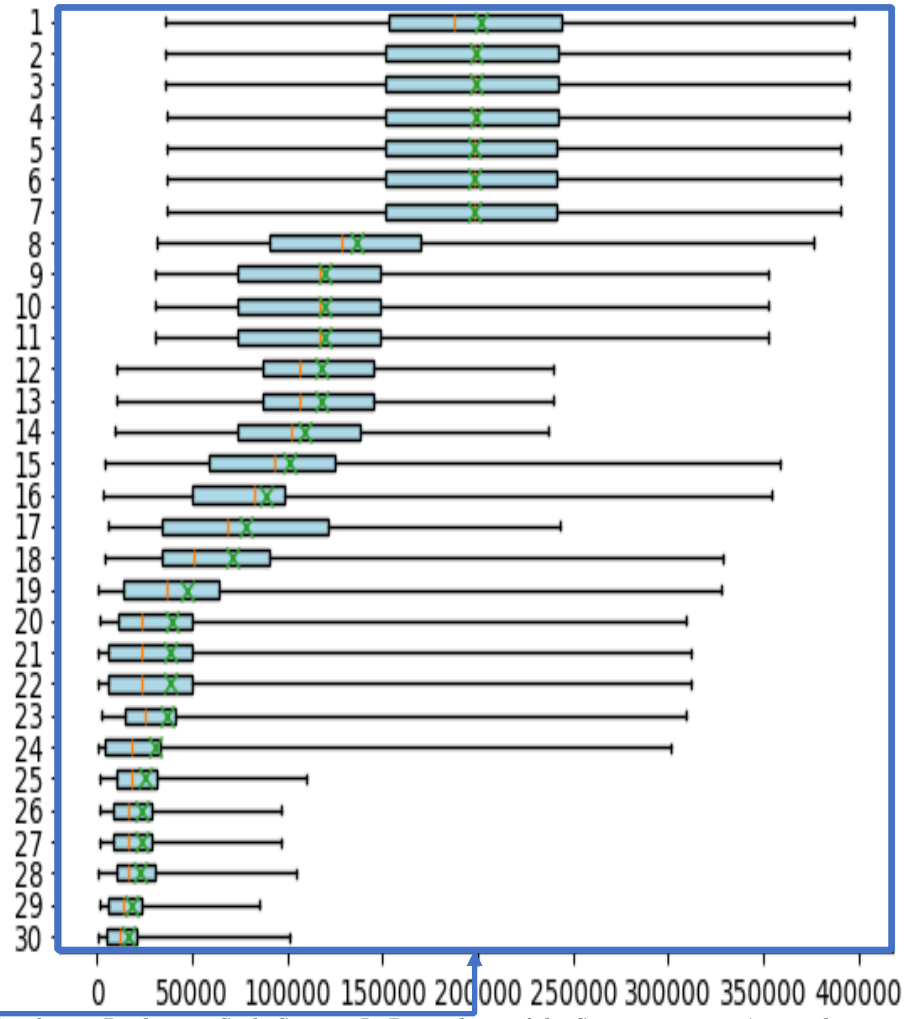
App log – Shared library usage count

- **Basic shared libraries are majority**
  - Libraries related to MPI, OpenMP, Fortran, C++, profiler
  - Note:
    - libfj*.so: Fujitsu libraries.
    - libmpg.so: handles large shared-memory pages.
    - libelf.so: provides access to files in the Executable and Linkable Format (ELF).
    - libmpi*.so/lib*mpi.so: MPI-related libs
- **Well-known libraries used many times**
  - LAPACK
  - GROMACS
  - FFTW
  - …
- **Usage of those libraries are relatively consistent across different months**

Usage counts for the entire period

Rank	Shared library	Count
1	libmpg.so	9,493,441
2	libfjprofcore.so	9,342,392
3	libfjprofomp.so	9,342,392
4	libelf.so	9,339,212
5	libfj90i.so	9,321,471
6	libfj90f.so	9,321,471
7	libfjsrcinfo.so	9,321,150
8	libatomic.so	6,421,803
9	libfjomp.so	5,633,687
10	libfjomphk.so	5,633,197
11	libfjompcrt.so	5,633,107
12	libmpi_usempi_ignore_tkr.so	5,541,646
13	libmpi_usempif08.so	5,541,378
14	libfjprofmpif.so	5,142,711
15	libmpi_cxx.so	4,757,229
16	libfjprofmpi.so	4,154,245
17	libstdc++.so	3,679,499
18	libfjcrt.so	3,331,647
19	libmpi_mt.so	2,238,543
20	**libfjlapacksve.so**	**1,856,063**
21	libc++.so	1,799,068
22	libc++abi.so	1,799,068
23	libfj90rt.so	1,713,729
24	**libgromacs_mpi.so**	**1,429,380**
25	libmpi.so.0	1,185,445
26	libfjc++.so	1,103,723
27	libfjc++abi.so	1,103,676
28	**libfftw3.so**	**1,051,486**
29	libfjdemgl.so	836,751
30	**libfjlapackexsve.so**	**763,892**

Monthly usage count



[73] Yosuke Asai, Kento Sato, Keiji Yamamoto, Hitoshi Murai, and Kohei Yoshida. 2026. System Software Utilization on an ARM-Based Supercomputer: Insights from a Production-Scale System. In Proceedings of the Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region Workshops (SCA/HPCAsiaWS '26). Association for Computing Machinery, New York, NY, USA, 110–121. https://doi.org/10.1145/3784828.3785349

System Software Utilization on an ARM-Based Supercomputer: Insights from a Production-Scale System (HPCAsia2026 [73])

Yosuke Asai[†1], Kento Sato[†1], Keiji Yamamoto[†1], Hitoshi Murai[†1], Kohei Yoshida[†1,2]
†1: RIKEN R-CCS, †2: The University of Electro-Communications

- **C/C++ vs. Fortran compiler calls (Figure 1)**
  - Fortran compilers are used more than C/C++
  - Many of applications/programs in Fugaku are written in Fortran

  → Continuous support for Fortran language/compier in HPC is necessary for FugakuNEXT and beyond
- **C vs. C++ compiler calls (Figure 2)**
  - C++ compilers are used more than C across different months

  → Reasonable to consider deployment of C++-based frameworks such as Kokkos, StdPar and SYCL
- **Future work:** See other metrics, e.g., counting # of unique users (in total and in each research area)

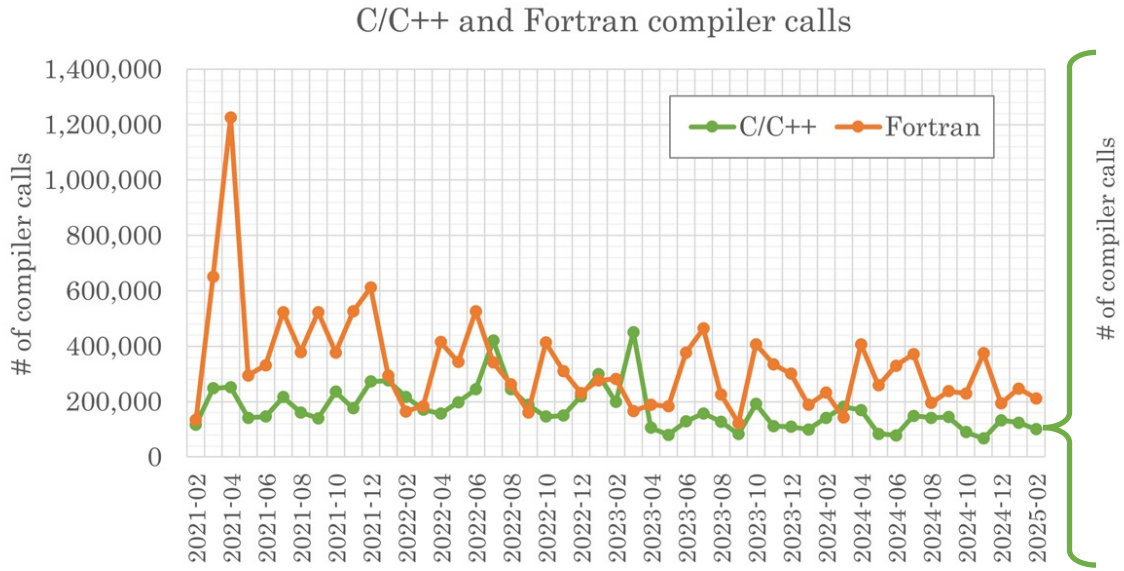


Figure 1

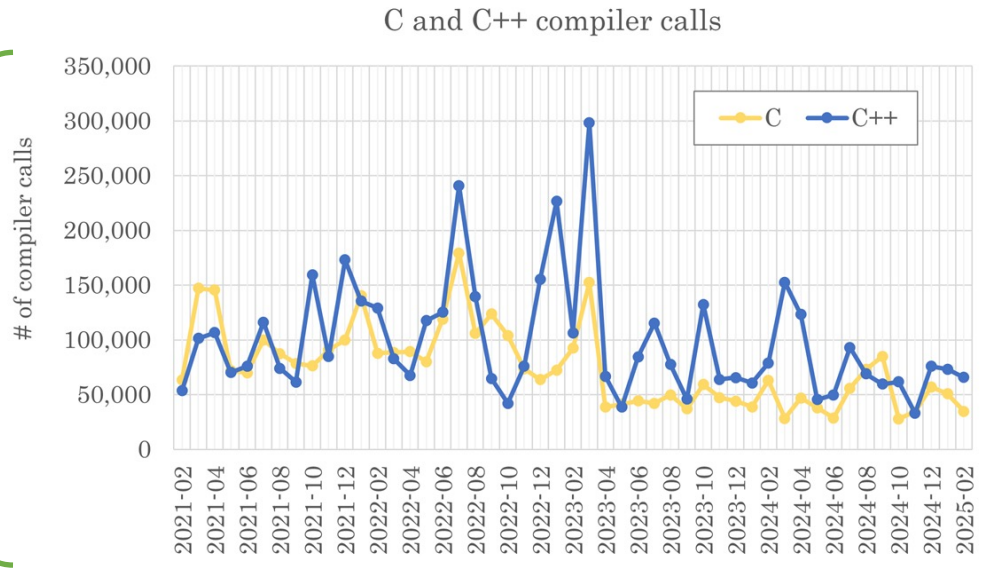


Figure 2

[73] Yosuke Asai, Kento Sato, Keiji Yamamoto, Hitoshi Murai, and Kohei Yoshida. 2026. System Software Utilization on an ARM-Based Supercomputer: Insights from a Production-Scale System. In Proceedings of the Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region Workshops (SCA/HPCAsiaWS '26). Association for Computing Machinery, New York, NY, USA, 110–121. https://doi.org/10.1145/3784828.3785349

Summary: AI for Science Platform towards FugakuNEXT

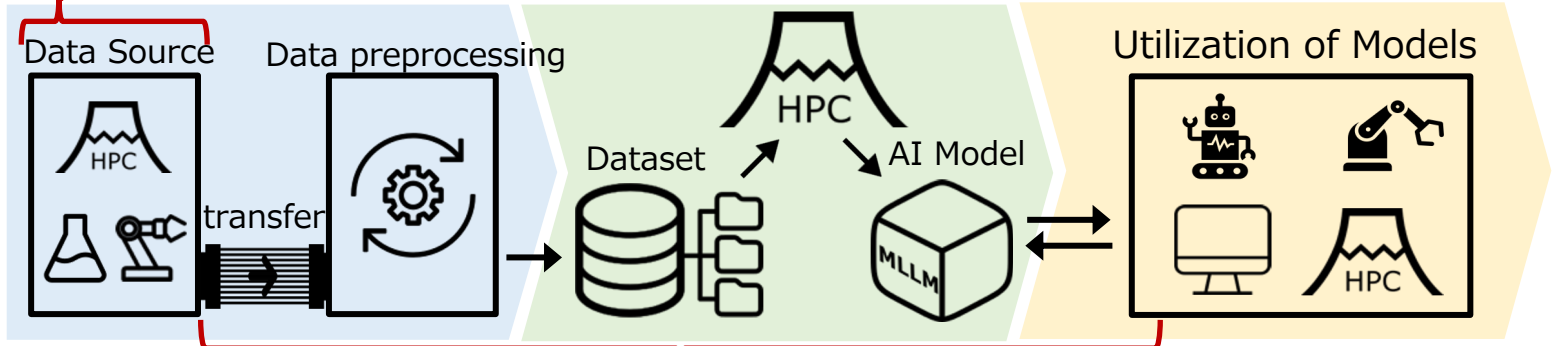
(T2) Data aggregation | (T2) MPI I/O | (T2) Debugging tools | (T2) AutoCheck

Advanced simulation

(T2) 3D parallelisms | (T2) C/R for AI | (T2) Data plancement

Advanced AI Computation

FugakuNEXT & FS3.0



Data compression for data management

(T1) Text data compression | (T1) TEZip enhamcement

(T1) Image data compression | (T1) XCT data compression

Next-Gen Computing Infra. Research

(T3) DSE

(T3) Software Usage Study

AI platform development and AI-assisted next-generation operations

(T1) AI Pipeline Dev. | (T3) Advanced Operaton by AI | (T2) Energy/Power Efficency Mechanism