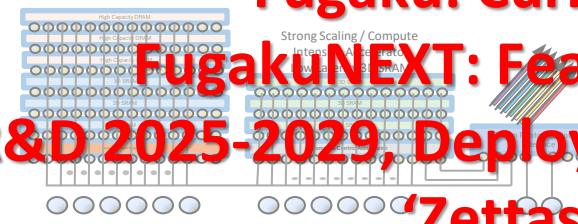
The Convergence of HPC, AI, and Quantum: Towards FugakuNEXT “made with Japan”



Satoshi Matsuoka
Director, Riken R-CCS
Hyperion HPC User’s Forum Paris
2025/10/7

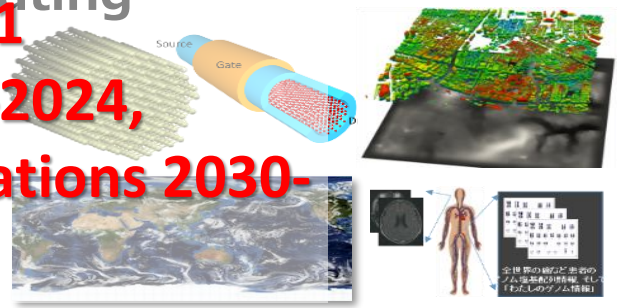
Riken R-CCS Strategy for Innovation by Computing Future of Science 'of' and 'by' Computing

- Science **of** High Performance Computing (towards 'Zettascale')

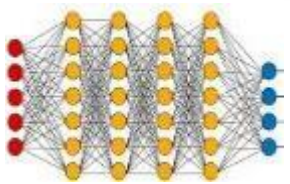


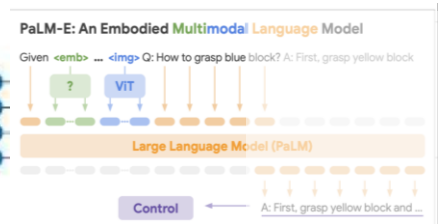
Fugaku: Current until 2030~2031
FugakuNEXT: Feasibility Study 2022-2024,
R&D 2025-2029, Deployment ~2029, Operations 2030-
'Zettascale' @ 40MW

- Science **by** High Performance Computing



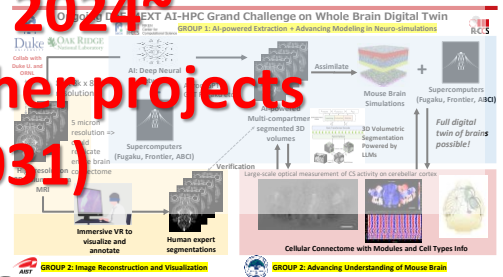
- Science **of** High Performance AI

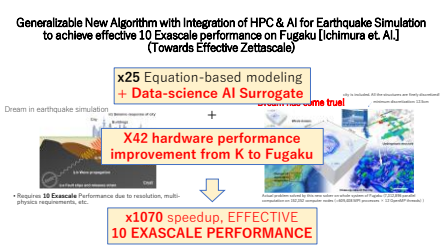




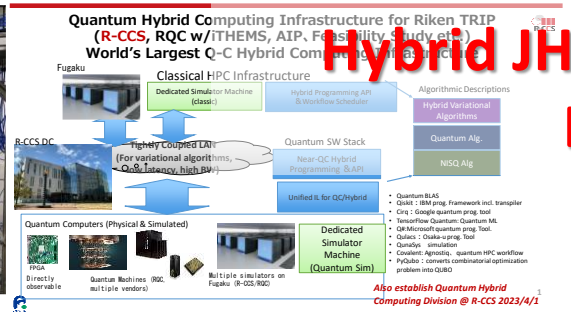
Riken AI for Science FY 2024~
including TRIP-AGIS and other projects
(TRIP-AGIS 2024~2031)

- Science **by** High Performance AI (AI for Science) w/HPC Simulations



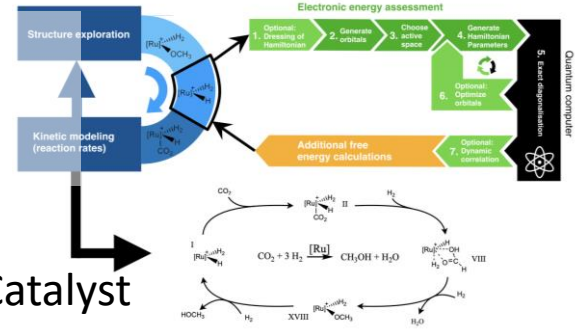


- Science **of** Quantum-HPC Hybrid Computing



Hybrid JHPC-Quantum Infrastructure Project
Deployment FY2023~2027

- Science **by** Quantum-HPC Hybrid Computing



Evolutions on National Lab-Vendor Relations for Japanese Flagship Supercomputers

- **Earth Simulator (2002) – Main development by NEC(+Fujitsu), JAMSTEC ES center @ Yokohama hosting**
  - #1 Top500 achieved via NEC's aging SX design due to unprecedented machine size
- **K Computer (2011) – Main development by Fujitsu w/ Riken management office in Tokyo, later AICS as a small research & operation center was formed @ Kobe**
  - Transitioned the JP community from classical vector to weak scaling massive parallel
  - #1 Top500 & 10 petaflops goal achieved by Fujitsu's HW technologies

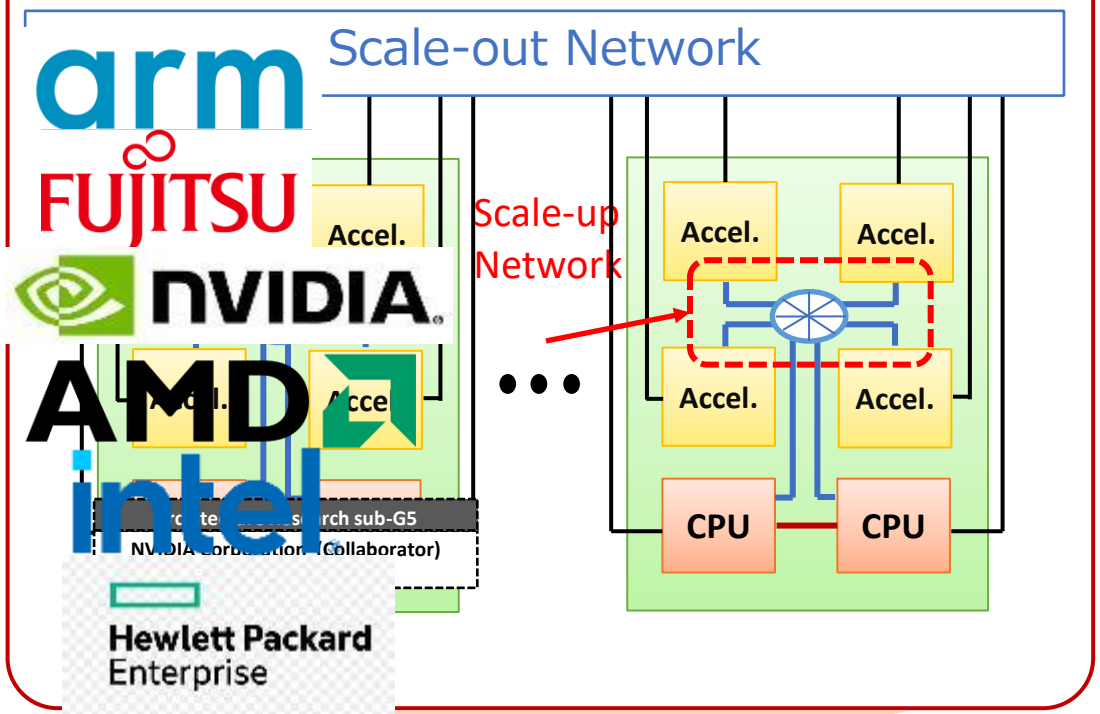
- **Fugaku (2020) – Main development by Fujitsu w/co-design management by Riken AICS dev office & application teams (more Riken involvement c.f. K-computer)**
  - Transition to R-CCS w/R&D Riken involvement e.g., DL4Fugaku, Graph500/HPL-MxP, COVID prog, etc.
  - Some alignment to international standards, e.g. Arm64, RHEL, Lustre (FEFS), …
  - 50-100x performance gain achieved thru 25-40x HW x 2-3x SW (algorithms)
  - R-CCS now one of the top HPC & AI-HPC & QC-HPC centers of the world

- **FugakuNEXT (2029) – Main development by R-CCS, w/partnerships with CPU & GPU vendors**
  - **CPU & GPU – JP and US vendors**
  - **System, network, storage – co-investigation by 3 parties**
  - **System software, application & algorithms, operations, testbeds, etc. – by R-CCS and partners**

FugakuNEXT (2029~2030) 'Post-Exa' Feasibility Study Design Evolution (Masaaki Kondo et. al.)

2025 After two years of design work with multiple vendors

System Architecture for AI-for-Science Computing Infrastructure



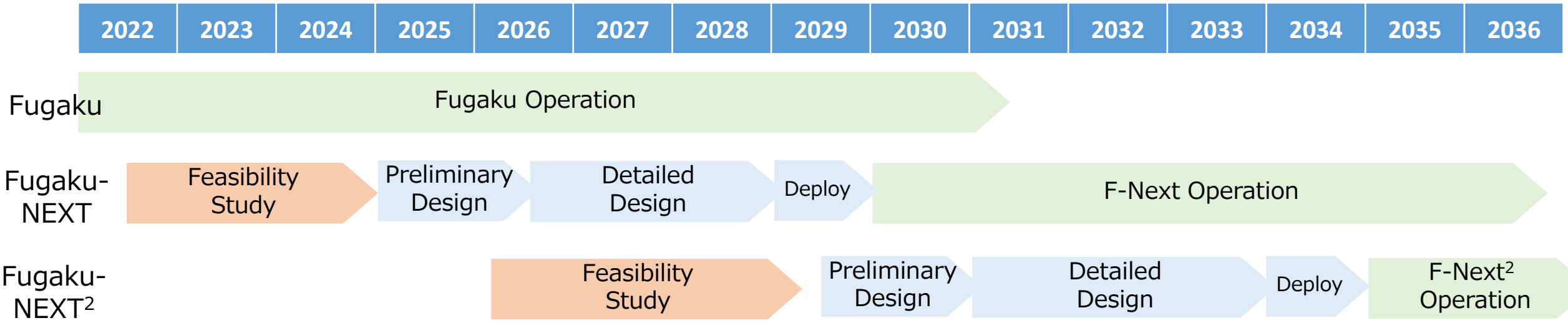
	CPU	GPU
Total Num. of Nodes	>= 3400 Nodes	
FP64 Vector FLOPS	>= 48PFLOPS	>= 3.0EFLOPS
FP16/BF16 AI FLOPS	>= 1.5EFLOPS	>= 150EFLOPS
FP8/INT8 AI (FL)OPS	>= 3.0EFLOP	>= 300E(FL)OPS
FP8 AI FLOPS (w/ sparsity)	—	>= 600EFLOPS
Memory Size	>= 10PiB	>= 10PiB
Memory Bandwidth	>= 7PB/s	>= 800PB/s
Total power consumption	< 40MW (compute node & storage)	

2023 Preliminary System target:
More than 5-10x effective performance improvement in HPC applications and more than 50EFLOPS AI training performance

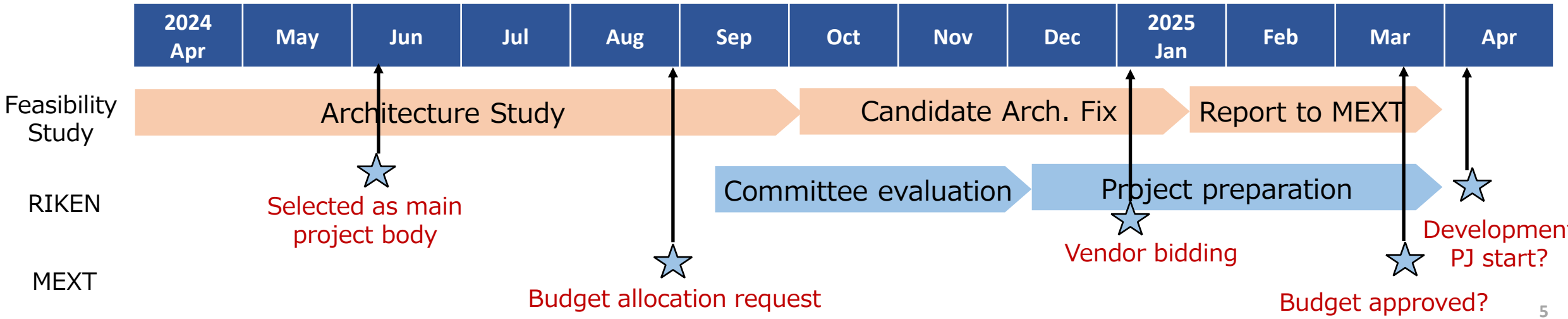
Effective Zettascale for AI and non-AI
(by 2029-2030 FugakuNEXT)

Expected Timeline of Fugaku-NEXT R&D and Future Plan

- **Expected schedule**



- **What's going on in FY2024 for Fugaku-NEXT development**



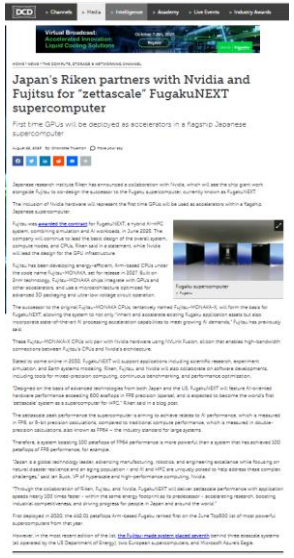
FugakuNEXT Vendor Partner Announcement Aug. 22, 2025

We announced partnerships with Fujitsu and NVIDIA to develop FugakuNEXT by 2029 and deployment in 2030 as a Japanese national supercomputing project. The co-design development will involve the Monaka-X AI capable Arm enterprise CPU by Fujitsu and 2029 generation GPU by NVIDIA

Nikkei Newspaper Headline, NHK TV and many other national news media

Media	
Web&Newspaper	39
TV Coverage	7
Total	**46**

News overages during 2025/8/22-28


Japan's Riken partners with Nvidia and Fujitsu for "zettascale" FugakuNEXT supercomputer


RIKEN, Fujitsu, NVIDIA Collaborate on FugakuNEXT Supercomputer

FUJITSU

NVIDIA

FUJITSU MONAKA-X (English translation of Fujitsu's original slides)

FUJITSU-MONAKA-X(1.4nm)

Follow-on to Fujitsu MONAKA
Arm CPU 2026-7 (2nm)

FUJITSU

Japan's state-of-the-art domestic CPU for AI **2029**

High-Perf. AI with NPU

- World's first implementation of low-precision matrix **Arm SME** in a server CPU, enabling low-latency AI processing
- Top AI-CPU performance for standalone & w/GPU
- Expansion of AI acceleration frameworks and libraries

High Scalability for HPC

- Ultra-many-core integration through next-generation 3D many-core architecture
- High-speed computation enabled by SIMD extensions
- Enhancement of high-performance compilers and libraries for HPC

Tight Integration with GPUs

- High-speed AI training and GPU-optimized apps through adoption of high-bandwidth data transfer with GPUs

Power Efficiency & Security

- Adoption of advanced semiconductor processes
- ultra-low-voltage operation control
- Confidential Computing

HPC

Big data processing

- Climate change modeling
- Development of new drag discovery methods
- Advancement of financial service, etc.

Datacenter

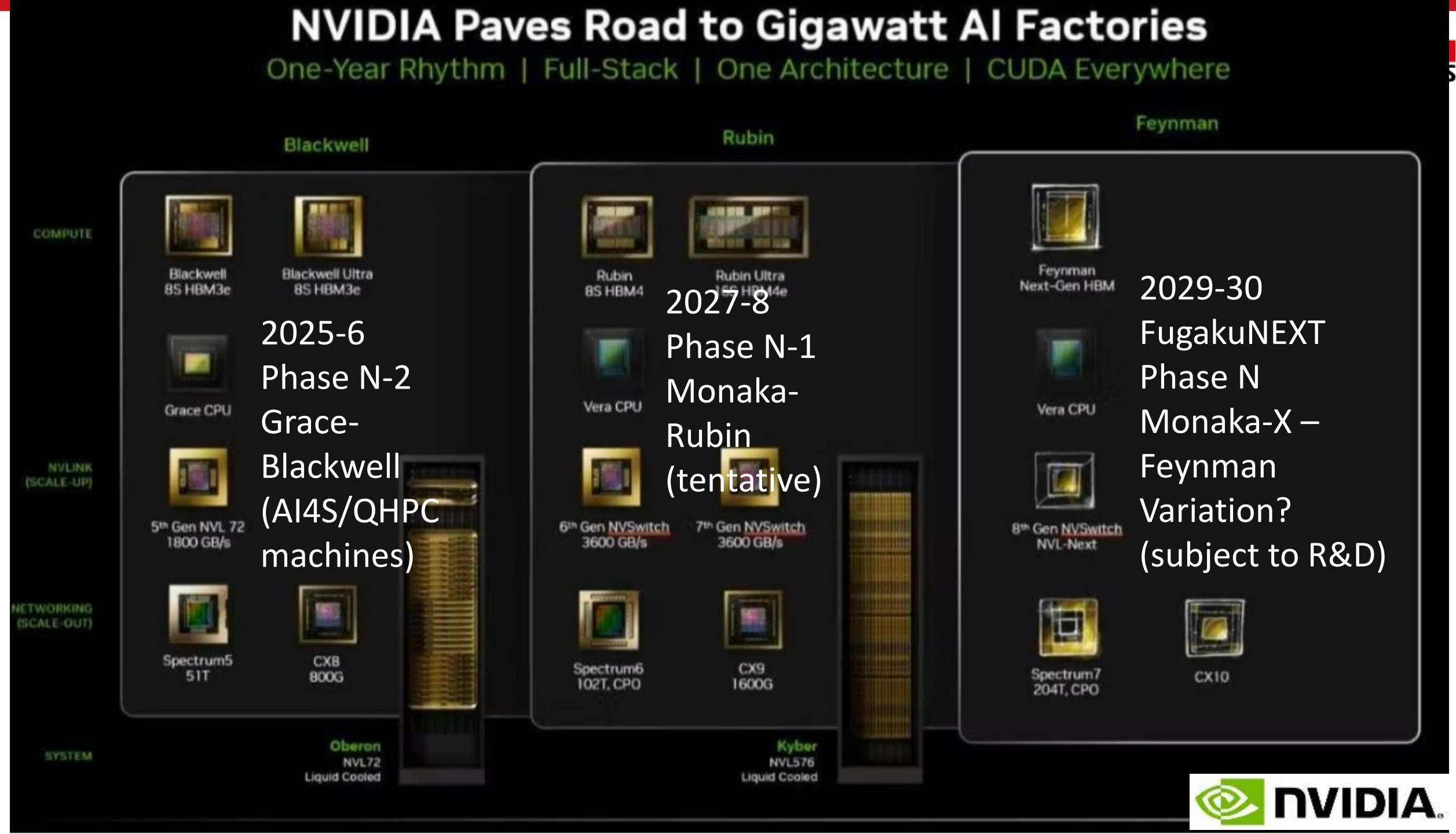
Scalability for Cloud Computing

- Energy and space efficiency
- Optimization for AI training and inference infrastructure
- Advanced security, etc.

Edge Computing

Real-Time and Edge AI

- National security
- Telecommunication infrastructure
- Robotics, etc.


NVIDIA Paves Road to Gigawatt AI Factories
One-Year Rhythm | Full-Stack | One Architecture | CUDA Everywhere
Blackwell
Rubin
Feynman
COMPUTE
NVLINK (SCALE-UP)
NETWORKING (SCALE-OUT)
SYSTEM
Blackwell 8S HBM3e
Blackwell Ultra 8S HBM3e
Grace CPU
5th Gen NVL 72 1800 GB/s
Spectrum5 51T
CX8 800G
Oberon NVL72 Liquid Cooled
Rubin 8S HBM4
Rubin Ultra
Vera CPU
6th Gen NVSwitch 3600 GB/s
7th Gen NVSwitch 3600 GB/s
Spectrum6 102T, CPO
CX9 1600G
Kyber NVL576 Liquid Cooled
Feynman Next-Gen HBM
Vera CPU
8th Gen NVSwitch NVL-Next
Spectrum7 204T, CPO
CX10
2025-6 Phase N-2 Grace-Blackwell (AI4S/QHPC machines)
2027-8 Phase N-1 Monaka-Rubin (tentative)
2029-30 FugakuNEXT Phase N Monaka-X – Feynman Variation? (subject to R&D)
NVIDIA

Project Objectives for FugakuNEXT (in order of priority)

1. Fujitsu to develop CPUs, including Monaka and Monaka-X… in collaboration with GPU vendors, cultivate the AI-HPC hyperscaler market to establish a sustainable and profitable CPU business – “Made with Japan”
2. Modernize scientific applications, where Japan has lagged, by integrating GPU-enablement as well as "AI for Science," and transitioning to a modern DEV environment centered on IDEs, CI/CD/CB (Continuous Benchmarking), and AI-assisted coding. This will enable the societal implementation of software and research outcomes, significant impact in Japan and the world.
3. Establish a sustainable, long-term research and development framework for HPC-AI-Quantum computing centered at R-CCS, encompassing FugakuNEXT, FugakuNEXT-NEXT, and future projects, solidifying our position as a world-leading HPC lab.
4. To successfully build FugakuNEXT (lowest priority? :-).

FugakuNEXT R&D Organizations

- **Riken R-CCS to assume leadership of the project, in collaboration with the GPU & CPU vendors Fujitsu and NVIDIA, as well as international leadership HPC/AI organizations**
- **The Next-generation platform division headed by M. Kondo to assume day-to-day development activities, but the entire R-CCS will participate in the R&D**
- **R&D will be open and sustainable (unlike previous projects)**

DOE-MEXT MoU on HPC（2024/4/9）

Participation by US & JP vendors from the onset (FS)

Town Hall Meetings & other user engagements on phase 2/3 platforms

Univ. & National Labs

HPCI Centers w/Riken partnership

Riken R-CCS + GPU/CPU Vendors

Other vendor contractors

DoE Labs

User Community

EuroHPC Centers

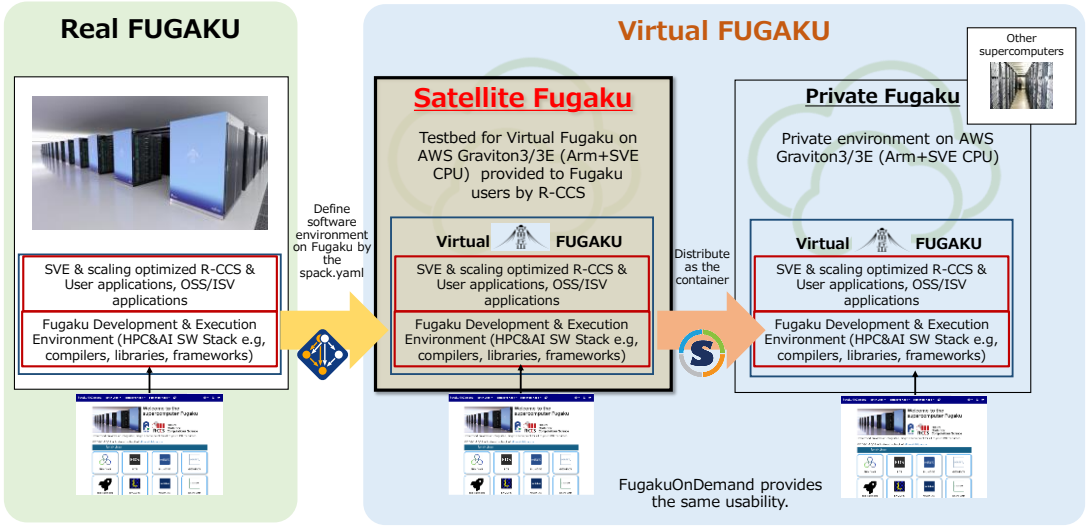
DoE-MEXT workshops（quarterly since 2023）

Open source community e.g. HPSF

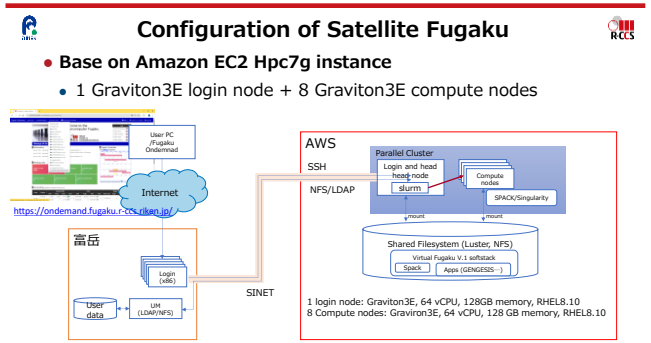
ECP EXASCALE COMPUTING PROJECT E4S HPSF HIGH PERFORMANCE SOFTWARE FOUNDATION

Virtual Fugaku: Vendor-independent, general-purpose, high-functionality HPC software stack for Clouds, Fugaku/FugakuNEXT and Other SCs

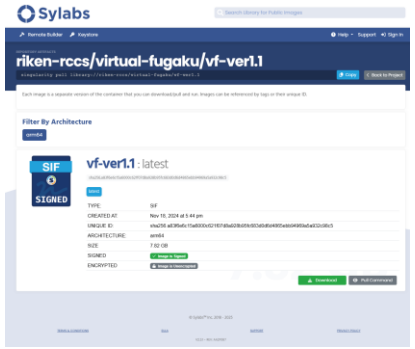
- **Initial release (V 1.0) August 5, 2024,**
- **Enhanced version (V1.1) released Nov 17, 2024**
  - Added industry applications and AI frameworks
- **Started providing the following two environments targeting AWS Graviton CPUs (August 5, 2024)**
  - Now v. 1.2 and continue to improve
  - Will provide x86/GPU environments in the future



① ‘Satellite Fugaku’ Certification environment for Fugaku users (on AWS)



② ‘Private Fugaku’ Singularity container distro for AWS users



Packages in Virtual Fugaku 1.x

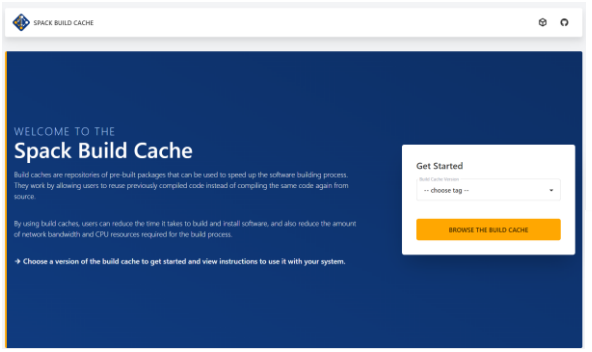
- **Virtual Fugaku 1.x includes the following software packages along with their many dependencies.**
  - Selected from the most frequently used Spack packages on *real* Fugaku since July, 2022.
  - Built with GCC 14.1.0 and EFA-enabled OpenMPI 4.1.6.

Name	Description	Vesrion	Spack Package
GENESIS	molecular dynamics	2.1.3	genesis
Gnuplot	graphing utility	6.0.0	gnuplot
GROMACS	molecular dynamics	2024.2	gromacs
GNU Scientific Library (GSL)	numerical library	2.7.1	gsl
Julia	programming language	1.10.2	julia
LAMMPS	molecular dynamics	20230802.3	lammps
Metis	graph partitioner	5.1.0	metis
Open Babel	chemical toolbox	3.1.1	openbabel
OpenFoam	CFD	2312	openfoam
Paraview	visualization	5.12.1	paraview
Parmetis	parallel graph partitioner	4.0.3	Parmetis
Atomic Simulation Environment	atomistic simulation	3.21.1	py-ase
Matplotlib	Visualization	3.9.0	py-matplotlib
MPI for Python	Python bindings for MPI	3.1.6	py-mpi4py
NumPy	array computing in Python	1.26.4	py-numpy

Name	Description	Vesrion	Spack Package
pandas	data analysis and manipulation	2.1.4	py-pandas
scikit-learn	machine learning and data mining	1.5.0	py-scikit-learn
SciPy	Fundamental algorithms for scientific computing	1.13.1	py-scipy
TOML	Python library for TOML	0.10.2	py-toml
Quantum Espresso	ab initio calculation	7.3.1	quantum-espresso
SCALE	weather and climate	5.4.4	scale
tmux	terminal multiplexer	3.4	Tmux
CP2K	quantum chemistry	2024.1	cp2k
CPMD	ab-initio molecular dynamics	4.3	cpmd
FrontISTR	Large-Scale Parallel FEM	5.3	frontistr
AutoDock-Vina	molecular docking	1.2.3	autodock-vina
PyTorch	Tensors and Dynamic neural networks	2.1.1	py-torch
TensorFlow	machine learning framework	2.14.1	py-tensorflow

Virtual Fugaku as a basis of FugakuNEXT Software

- **Development Related**
  - Expansion of included packages and improved packaging efficiency
    - ISV addition trial: STAR-CCM (Siemens under consideration)
    - Automation through CI/CD/CT process (already started)
    - Graviton4 evaluation
  - Simplified introduction for Private Fugaku
    - Trial of Spack Build Cache
    - Utilization of Amazon Machine Images (AMI) and other AWS frameworks
  - Application to R-CCS planned systems → Inheritance as a standard stack for Fugaku NEXT, etc.
    - Trial deployment to other architectures such as x86 → Application to RIKEN AI4S, RIKEN JHPC-Quantum supercomputers, etc., and provision to other HPCI centers
    - Deployment in Phase 1, Phase 2
- **Initiatives for community building and cultivation**
  - Strategic participation in the HPC Software Foundation => Collaboration with DoE E4S HPC Stack
  - Community building through the SPACK community and HPSF
  - Participation in various events (APAN59 3/6 Cloud WG, SCAsia2025 AWS tutorial, introduction at 2025 ACM ASEAN School)
- **Breaking away from vendor dependence => As a base for the next-generation Fugaku system software development**



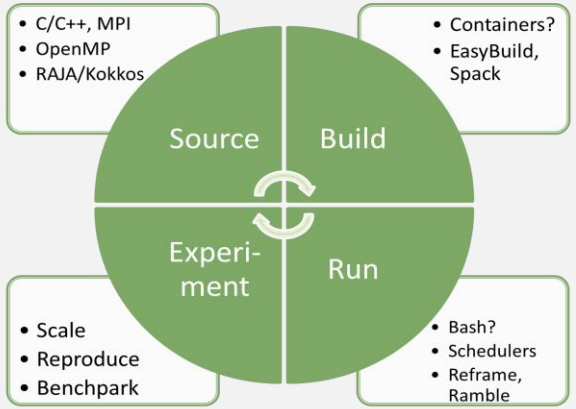
Amazon Machine Image (AMI)

HPSF
HIGH PERFORMANCE SOFTWARE FOUNDATION

Riken Leadership in FugakuNEXT Applications

① Code Porting & Evaluation

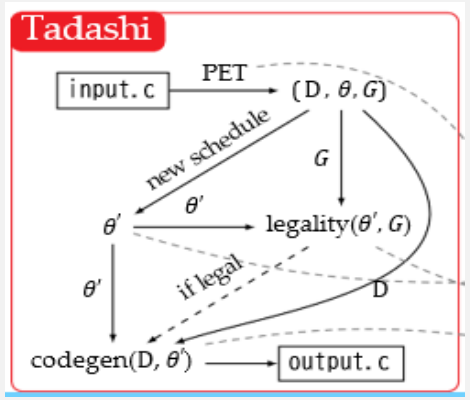
Code Porting Support on Phase 2 machines w/CI/CD/CB



- Collaboration w/DoE and Vendors to establish common CI/CD/CB
- Early participation of wide-ranging apps community

② AI-based Code Modernization

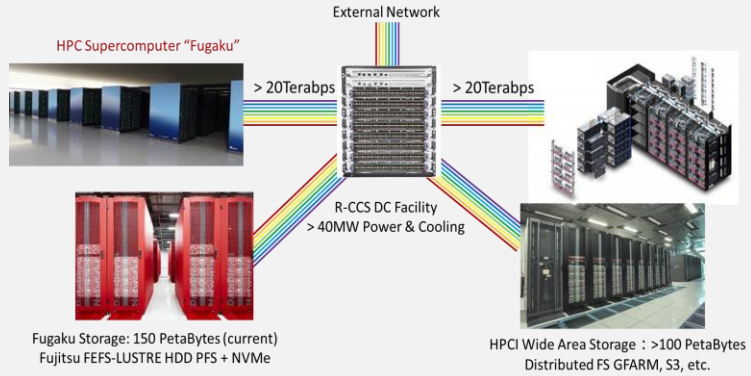
Coding AI to generate & Port Legacy HPC Codes



- AI already being incorporated into Fugaku Services
- AI coding support to port, tune, incorporate new algorithms

③ FugakuNEXT Phase 2 Proxy

As targets for code dev & porting, onto Phase 3 and 4



- Phased deployment of N-2, N-1, … platforms
- Utilize production platforms (AI4S, Quantum-HPC) to test system software and apps

FugakuNEXT Partnership Program for Early Access to Development

Phased Dev&Deployment Towards FugakuNEXT

Riken-lead continuous development of system software and apps utilizing Fugaku+AI4S+JHPC-Q & preproduction machines

Current R-CCS Cloud

- Phase1 (2025/4) – AI4S phase1, A set of small cluster of a variety of GPUs (total 200GPU), First platform towards Virtual Fugaku based SW

New Cluster Room

- Phase2 (End of FY2025) – AI4S phase 2 + JHPC-Q, Total 2130 NVIDIA GB200NVL4 GPUs in APU config + Virtual Fugaku + DoE E4S + AI + Hybrid QC Software Stack and CI/CD/CB Platform
- Phase 3 (FY2027) – Dedicated mid-sized cluster consisting of GPU/APU one generation prior to FugakuNEXT, Riken SysSoft + Application CI/CD/CB + operations rehearsal

New Datacenter

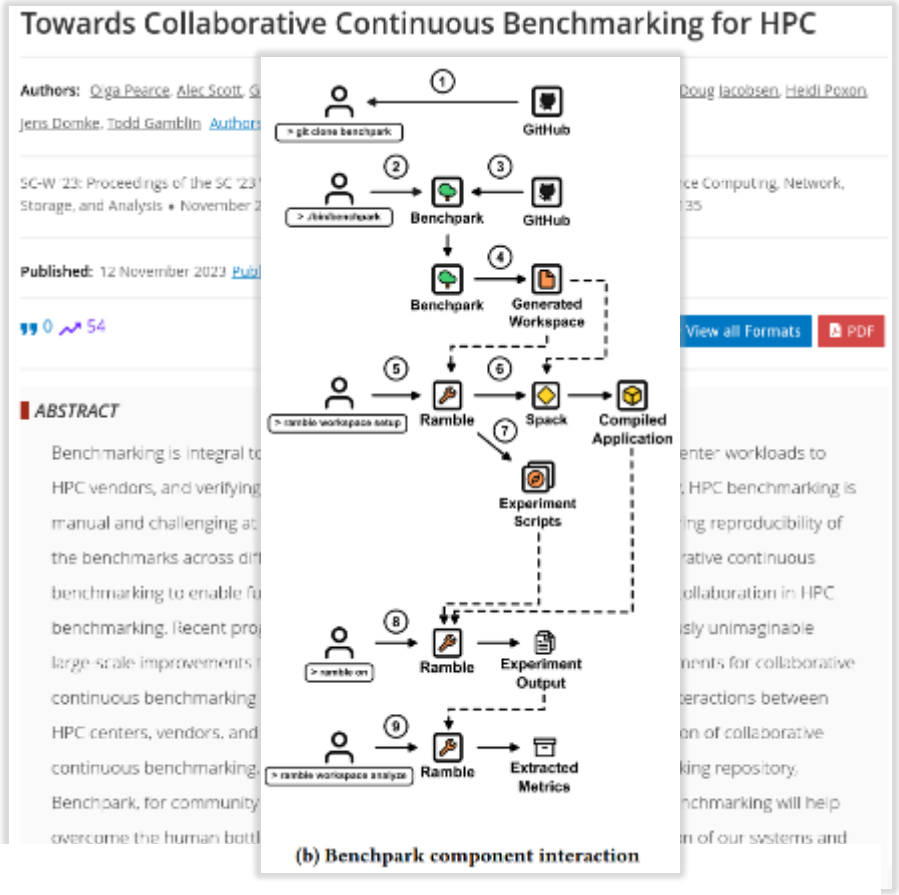
- Phase 4 (FY2029-30) FugakuNEXT deployment & operations

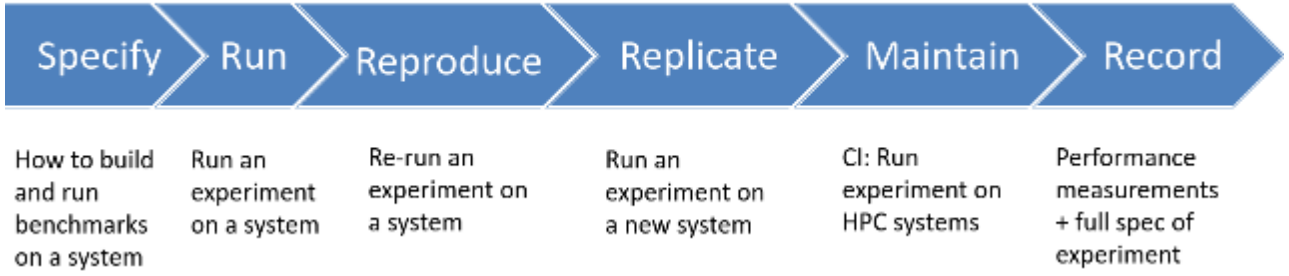
BenchPark: DoE and RIKEN Collaboration on CI / CD / CB [Olga Pierce @ LLNL, Jens Domke @ R-CCS, …]

- **Collaborative Continuous Benchmarking**
  - Designed to **minimize human effort** by maximizing reuse and supporting easy collaboration between HPC centers, vendors, and researchers
  - **Components**: Source code, Build instr., Inputs, Run instructions, Exp. evaluation, and CI testing
  - **Security** considerations: Hubcast for mirroring code, Jacamar for GiLab runners with authorized account
- **Current status:**
  - Know systems (10): Fugaku, CSC LUMI, JSC Juwels, CSCS Piz Daint, various LLNL (ElCapitan) and LANL
  - 20 HPC benchmarks; incl. 3 from R-CCS (QWS, GENESIS, SALMON)

Paper: dl.acm.org/doi/10.1145/3624062.3624135

Repo: github.com/LLNL/benchpark





AMD vs NVIDIA Accelerator Specifications

	VR200 NVL144	VR300 NVL576	MI400	MI400 vs VR200
Peak TDP	1,700W	2,800W	1,800W	1.1x
BF16 Dense TFLOP/s	8,333	16,667	10,000	1.2x
FP8 Dense TFLOP/s	16,666	33,334	20,000	1.2x
FP6 Dense TFLOP/s	16,666	33,334	40,000	2.4x
FP4 Dense TFLOP/s	50,000	50,000	40,000	0.8x
Supported FP4 Dtypes	OCP MX4, NVFP4	OCP MX4, NVFP4	OCP MX4	
Memory Bandwidth	13.0 TByte/s	32.0 TByte/s	19.6 TByte/s	1.5x
Memory Capacity	288 GB	1,024 GB	432 GB	1.5x
Scale Up World Size	72	144	72	
Scale Up Bandwidth (Uni-di)	1,800 GByte/s	1,800 GByte/s	1,800 GByte/s	1.0x
Scale Out Bandwidth (Uni-di)	1,600 Gbit/s	1,600 Gbit/s	2,400 Gbit/s	1.5x
Cooling	DLC	DLC	DLC	

1. The VR200 NVL144 has 144 compute chiplets across 72 logical GPUs.
2. The VR300 NVL576 has 576 compute chiplets across 144 logical GPUs.

Source: SemiAnalysis Estimates

Not much advance BW/W c.f. Fugaku (~50GB/W)

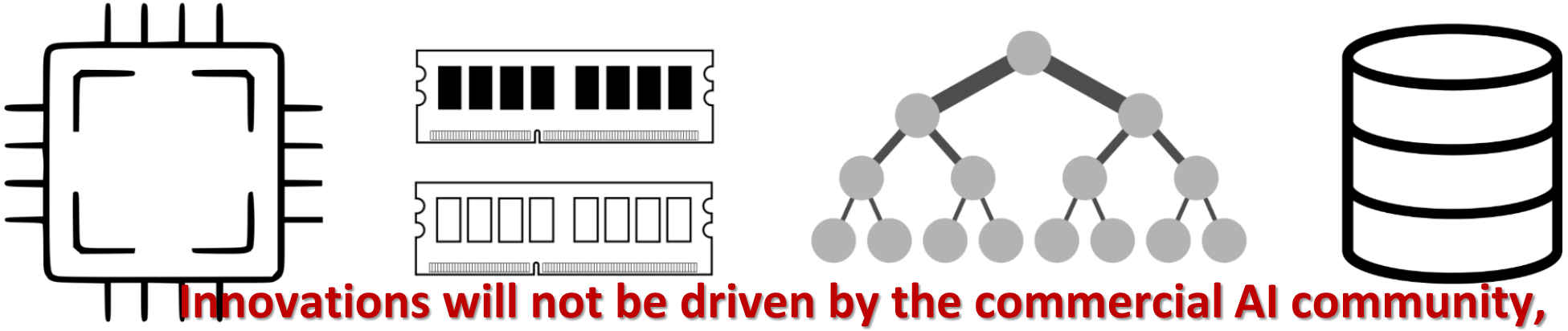
Future HBM Roadmap not very favorable

↓

Innovations in memory

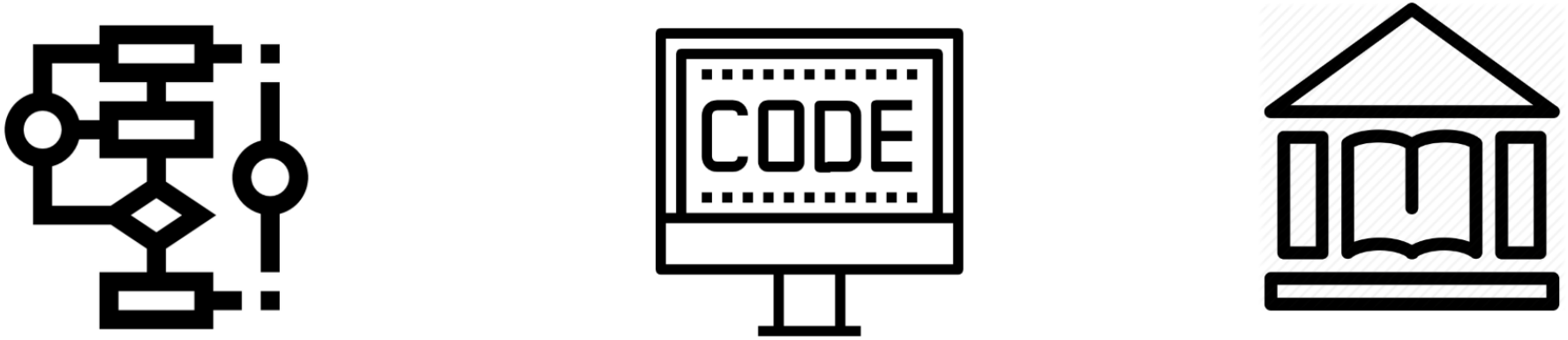
Divergence of HPC and AI (Convergence)

Compute **Memory** **Network** **Storage**



Innovations will not be driven by the commercial AI community, but rather the HPC(/AI) community has to lead

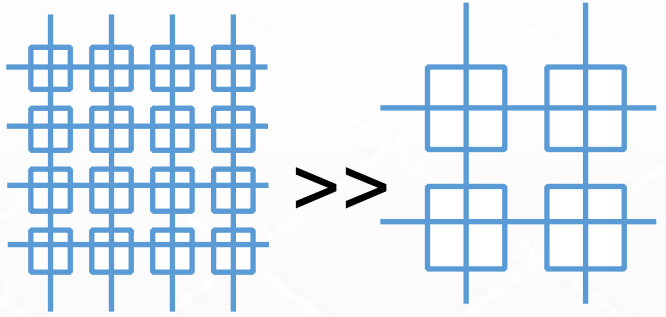
Algorithms **Programming** **Libraries (Sys SW)**



What about Dense Linear Algebra?

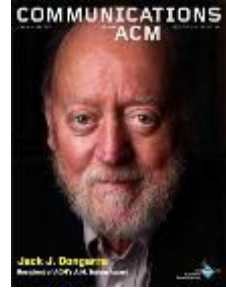
Precision Depending Analysis – what and how matrix engines provide good ROI relative to their silicon occupancy?

- Energy = compute (multipliers, volume) + data movement (between units, surface)
  - Low precision – low surface:volume, optimize to minimize data movement, matrix engines to minimize wire distance
  - High precision – high surface:volume, data transfer less problem, performance & energy gain small, dark silicon of unused multipliers wasteful, **wide vectors sufficient.**
- 4~16 bit apps: Deep Learning/AI training
- 19~ (TF32) ~ 32 bit apps: DL/AI, molecular dynamics, higher order methods (**mixed precision**)
- 64 bit apps: first-principle material science eg DFT => **Emulation of "64 bit" apps with "Ozaki Scheme II" => with 1/20 slowdown we expect effective 10 Exaflops from 200 INT8 ExaOps "Zettascale" AI machine (20x Fugaku)**



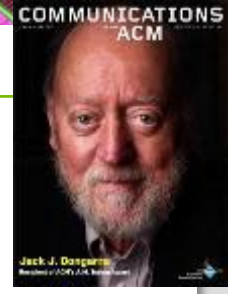
Low precision MM	High precision MM
Low volume (compute) : surface (comm) ratio	high volume (compute) : surface (comm) ratio
Matrix units help to reduce data transfer energy	Vector units may be sufficient as benefit of matrix may be low
Very wire energy efficient	NOT very energy efficient cf vectors

One Idea for Using Mixed Precision Goes Something Like This…



- Exploit lower arithmetic as much as possible.
  - Especially for the bulk of the computation
- Correct or update the solution with selective use of higher floating point arithmetic to provide a “refined results” (more accurate).
- Intuitively:
  - Compute a 32 bit result,
  - Calculate a correction to 32 bit result using selected higher precision (64 bit) and,
  - Perform the update of the 32 bit results with the correction using high precision (64 bit).

Slides courtesy Jack Dongarra, U-Tennessee



Leveraging Mixed Precision for Linear Algebra

Idea: use low precision to compute the expensive flops (LU $O(n^3)$) and then iteratively refine ($O(n^2)$) the solution in order to achieve the FP64 arithmetic

Iterative refinement for dense systems, $Ax = b$, can work this way.

L U = lu(A)	lower precision	$O(n^3)$
x = U\(L\b)	lower precision	$O(n^2)$
r = b – Ax (with original A)	FP64 precision	$O(n^2)$
WHILE || r || not small enough		
1. find a correction "z" to adjust x that satisfy Az=r		
solving Az=r could be done by either:		
➢ GMRes preconditioned by the LU to solve Az=r Iterative Refinement GMRes	lower precision	$O(n^2)$
2. x = x + z	FP64 precision	$O(n^1)$
3. r = b – Ax (with original A)	FP64 precision	$O(n^2)$
END		

Higham and Carson showed can solve the inner problem with iterative method and not infect the solution with the conditioning of the original matrix.

Slides courtesy Jack Dongarra, U-Tennessee

Originally motivated by the Sony PlayStation
SP peak 205 Gflop/s, DP peak 15 Gflop/s

J. Langou, et al., Exploiting the Performance of 32 bit fl-pt Arithmetic in Obtaining 64 bit Accuracy, in: Proc. of SC06

E. Carson & N. Higham, "Accelerating the Solution of Linear Systems by Iterative Refinement in Three Precisions *SIAM J. Sci. Comput.*, 40(2), A817–A847.

HPL-MxP Top 10 for June 2025

Source: Top500 Presentation, June 2025, Erich Strohmaier

Rank	Site	Computer	Cores	HPL Rmax (Eflop/s)	TOP500 Rank	HPL-MxP (Eflop/s)	Speedup
1	DOE/NNSA/ANL **USA**	**El Capitan**, HPE Cray 255a, AMD 4th Gen EPYC 24C 1.8 GHz, AMD Instinct MI300A, Slingshot-11	11,039,616	1.742	1	**16.7**	9.6
2	DOE/SC/ANL **USA**	**Aurora**, HPE Cray EX, Intel Max 9470 52C, 2.4 GHz, Intel GPU MAX, Slingshot-11	8,159,232	1.012	3	**11.6**	11.5
3	DOE/SC/ORNL **USA**	**Frontier**, HPE Cray EX235a, AMD Zen-3 (Milan) 64C 2GHz, AMD MI250X, Slingshot-11	8,560,640	1.353	2	**11.4**	8.4
4	AIST **Japan**	**ABCI 3.0**, HPE Cray XD670, Xeon Platinum 8558 48C 2.1GHz, NVIDIA H200, Infiniband NDR200	479,232	0.145	15	**2.36**	**16.3**
5	EuroHPC/CSC **Finland**	**LUMI**, HPE Cray EX235a, AMD Zen-3 (Milan) 64C 2GHz, AMD MI250X, Slingshot-11	2,752,704	0.380	8	**2.35**	6.2
6	RIKEN Center for Comput. Science, **Japan**	**Fugaku**, Fujitsu A64FX 48C 2.2GHz, Tofu D	7,630,848	0.442	6	**2.00**	4.5
7	EuroHPC/CINECA **Italy**	**Leonardo**, BullSequana XH2000, Xeon 8358 32C 2.6GHz, NVIDIA A100, QR NVIDIA HDR100 IB	1,824,768	0.241	9	**1.80**	7.5
8	CII, Institute of Science **Japan**	**TSUBAME 4**, HPE Cray XD665, AMD EPYC 9654 96C 2.4GHz, NVIDIA H100, Mellanox NDR200	172,800	0.035	47	**0.64**	**16.2**
9	NVIDIA **USA**	**Selene**, DGX SuperPOD, AMD EPYC 7742 64C 2.25 GHz, Mellanox HDR, NVIDIA A100	555,520	0.063	23	**0.63**	9.9
10	DOE/SC/LBNL/NERSC **USA**	**Perlmutter**, HPE Cray EX235n, AMD EPYC 7763 64C 2.45 GHz, Slingshot-10, NVIDIA A100	761,856	0.079	19	**0.59**	7.5

(New!) Ozaki Scheme II

Ultra-fast emulation of matrix multiplication using INT8 Tensor Cores

Advancing AI-oriented architectures
toward the evolution of
traditional scientific computing

Katsuhisa Ozaki

Yuki Uchino

Toshiyuki Imamura

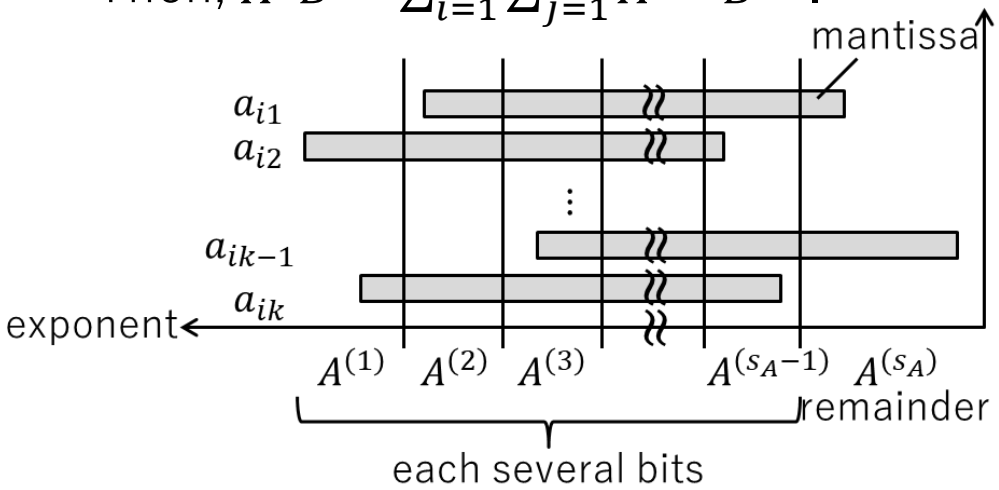
Ozaki-scheme:

- Error-free transformation of AB for $A \in \mathbb{R}^{k\times m}$ & $B \in \mathbb{R}^{k\times n}$.
- Simply, split A & B into sub-matrices such that

$$A = A^{(1)} + \cdots + A^{(s_A-1)} + A^{(s_A)},$$
$$B = B^{(1)} + \cdots + B^{(s_B-1)} + B^{(s_B)},$$

$$\begin{cases} {A^{(i)}}^T B^{(j)} = \mathrm{fl}\left({A^{(i)}}^T B^{(j)}\right), \text{for } \forall i,j \\ \max_p \left\lceil \log_2 |{A^{(i)}}_{pq}| \right\rceil \geq \max_p \left\lceil \log_2 |{A^{(i+1)}}_{pq}| \right\rceil + d, \text{ for } \forall i,q \\ \max_p \left\lceil \log_2 |{B^{(j)}}_{pq}| \right\rceil \geq \max_p \left\lceil \log_2 |{B^{(j+1)}}_{pq}| \right\rceil + d, \text{ for } \forall j,q \\ 2d + \log_2 k \leq \mathrm{bitsizeof(Acc)} \end{cases}$$

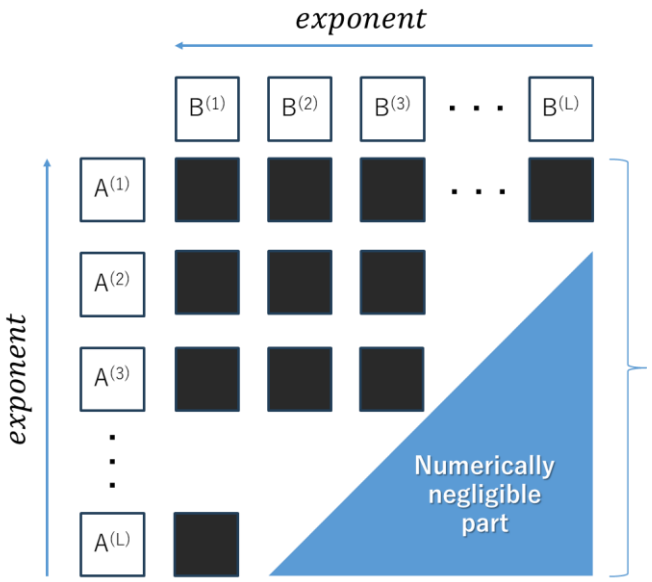
- Then, $A^T B = \sum_{i=1}^{s_A} \sum_{j=1}^{s_B} {A^{(i)}}^T B^{(j)}$.



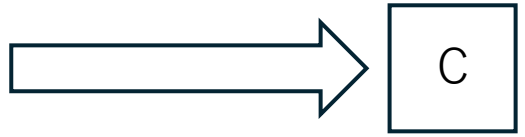
$$A^T = \mathrm{diag}(\mu) * \{A^{(1)}; A^{(2)}; A^{(3)} \ldots\}^T \left\{\left(2^{-0d}, 2^{-d}, 2^{-2d}, \ldots\right)^T \otimes I\right\}$$

$$B = \left\{\left(2^{-0d}, 2^{-d}, 2^{-2d}, \ldots\right) \otimes I\right\}\{B^{(1)}; B^{(2)}; B^{(3)} \ldots\} * \mathrm{diag}(\nu)$$

$$C = A^T B = \mathrm{diag}(\mu) * \{A;\}^T * ([d]^T \otimes I)([d] \otimes I) * \{B;\} * \mathrm{diag}(\nu)$$
$$= \sum_{ij} \mathrm{diag}(\mu)\, {A^{(i)}}^T 2^{-(i-1)d} 2^{-(j-1)d} B^{(j)} \mathrm{diag}(\nu)$$



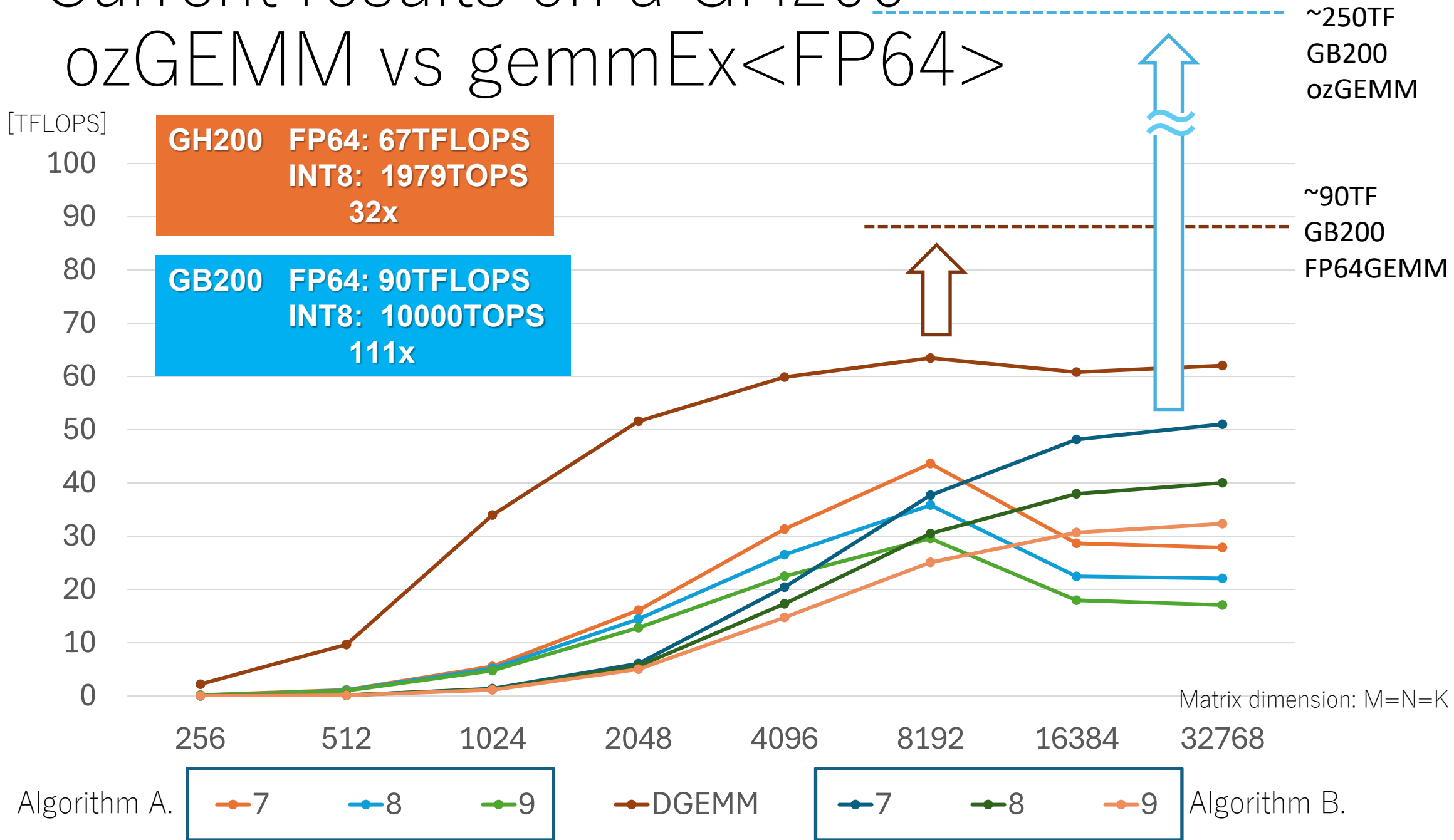
Re-scale the row/column-wise alignment of the exponent part and accumulate onto C



$$C \mathrel{+}= \mathrm{diag}(\mu) \times \{{A^{(i)}}^T B^{(j)}\} \times \mathrm{diag}(\nu) \times 2^{-(i+j-2)d}$$

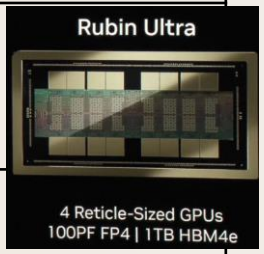
INT8GEMM by TensorCores/MatrixEngines

Current results on a GH200
ozGEMM vs gemmEx<FP64>



Throughput test (FP64 emulation)

	RTX4090	GH200	Rubin 300 Ultra (estimate by S. Matsuoka)
DGEMM (cuBLAS)	0.62 TFLOPS	60.9 TFLOPS	??? 100+ TFLOPS per 4 GPU die package extrapolation from Blackwell
Ozaki Scheme I	5.84 TFLOPS	34.5 TFLOPS	
Ozaki Scheme II (new)	**7.4–9.8** TFLOPS	**66.9–80.2** TFLOPS	**800~1300** TFLOPS per 4 GPU die package



With Ozaki Scheme II,
emulation outperforms native FP64
even on data center GPUs.

Consumer-grade GPU → RTX4090: INT8TC: FP64 = 512:1
DATA Center GPU → GH200: INT8TC: FP64TC = 29.5:1
DATA Center GPU → B200: INT8TC: FP64TC = **112.5:1**

Code is available at RIKEN's GitHub presented by **Dr. Uchino**
https://github.com/RIKEN-RCCS/accelerator_for_ozIMMU
https://github.com/RIKEN-RCCS/GEMMul8

Acceleration of Quantum Chemistry using Combinatios of Emulation (Ozaki) & Mixed Precision utilizing AI-Centric GPUs

Reducing Numerical Precision Requirements in Quantum Chemistry Calculations

William Dawson,*,† Jens Domke,† Takahito Nakajima,† and Katsuhisa Ozaki‡

†*RIKEN Center for Computational Science, Kobe, Japan*

‡*Shibaura Institute of Technology, Saitama, Japan*

E-mail: william.dawson@riken.jp

Abstract

The abundant demand for deep learning compute resources has created a renaissance in low precision hardware. Going forward, it will be essential for simulation software to run on this new generation of machines without sacrificing scientific fidelity. In this paper, we examine the precision requirements of a representative kernel from quantum chemistry calculations: calculation of the single particle density matrix from a given mean field Hamiltonian (i.e. Hartree-Fock or Density Functional Theory) represented in an LCAO basis. We find that double precision affords an unnecessarily high level of precision, leading to optimization opportunities. We show how an approximation built from an error-free matrix multiplication transformation can be used to potentially accelerate this kernel on future hardware. Our results provide a road map for adapting quantum chemistry software for the next generation of High Performance Computing platforms.

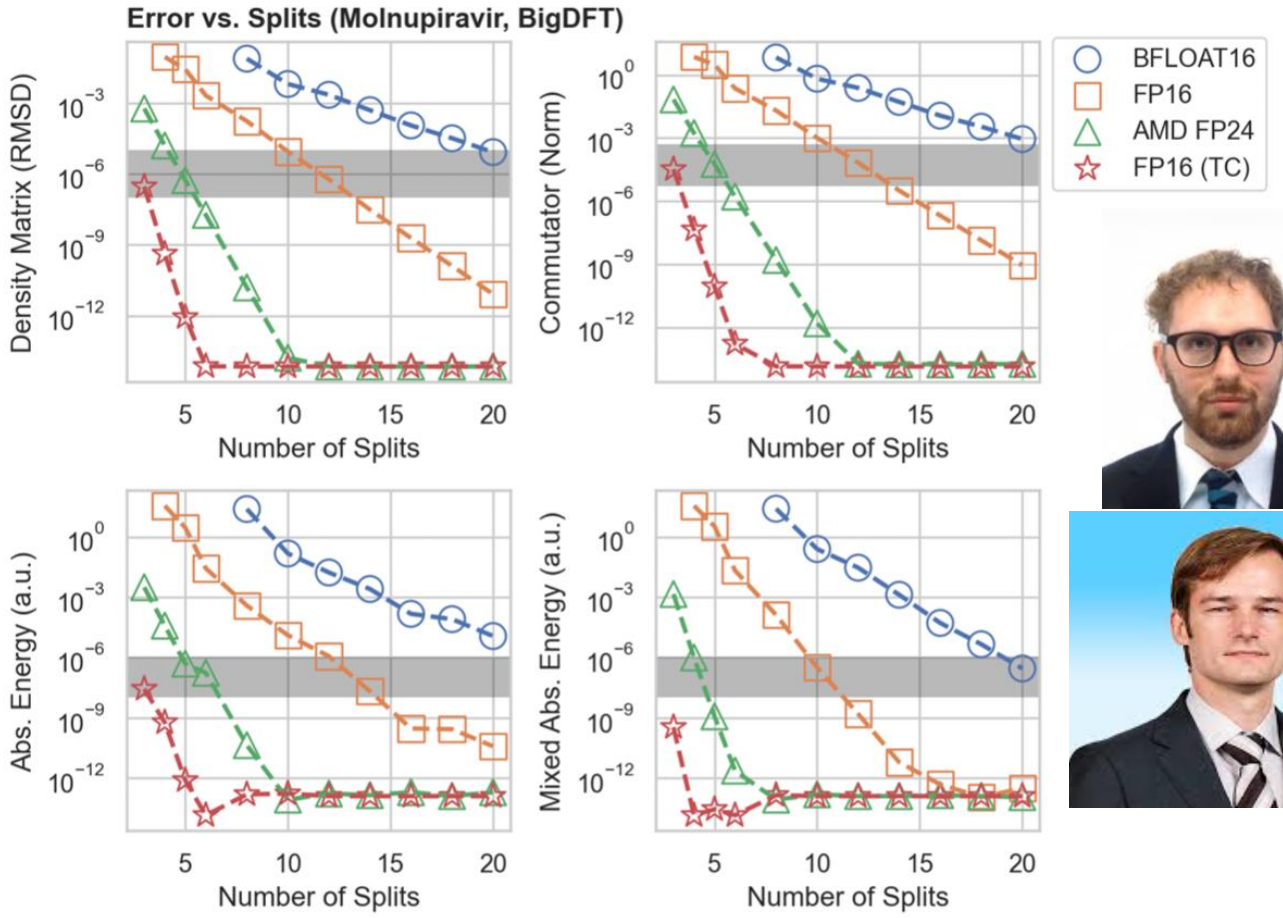
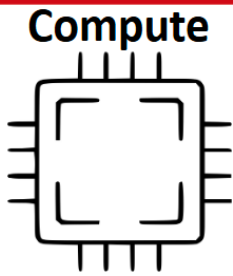
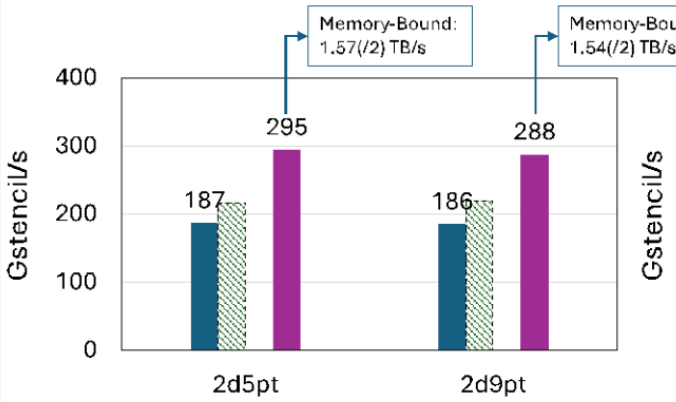


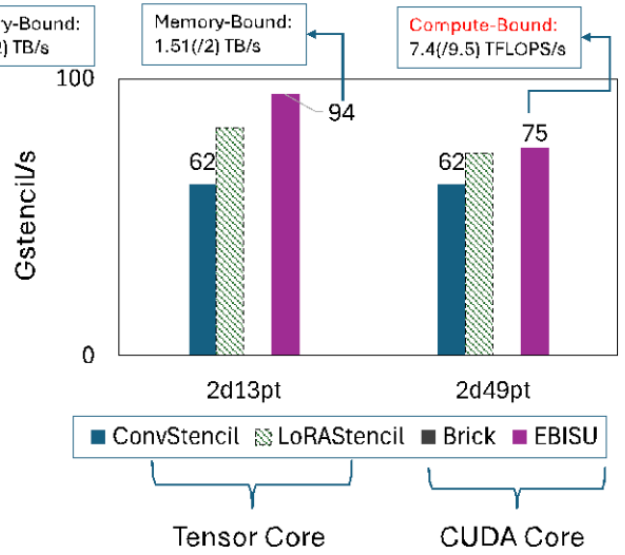
Figure 7: Errors when using purification vs. the number of splits used in the Ozaki scheme applied to the Molnupiravir / BigDFT system (matrix dimension 147 × 147). We examine the error for four different floating point types: BFLOAT16 (8, 7), FP16 (5, 10), AMD FP24

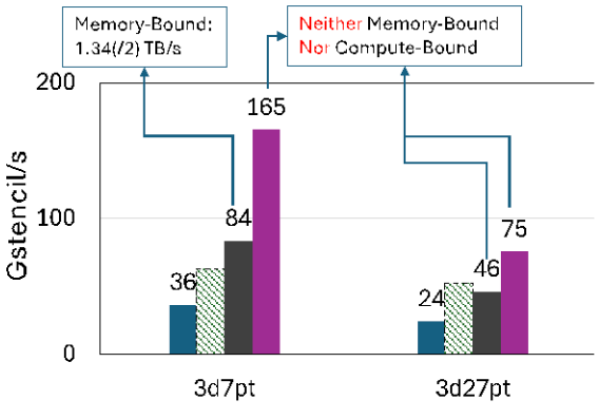
Memory-bound Codes

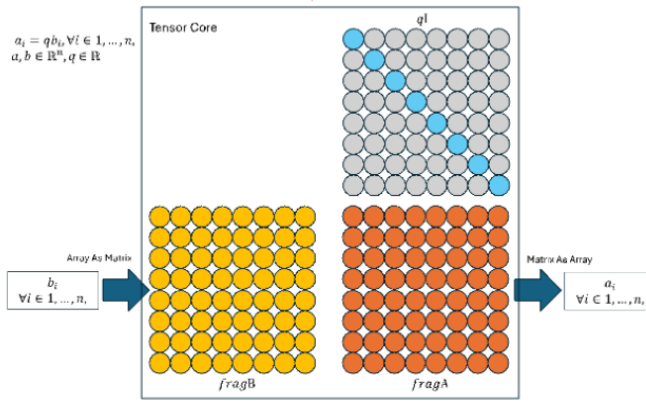


- Memory-bound code: what does low precision do there?
  - Simple answer: “reduces the memory traffic!”
  - Complex answer*: “low precision is mostly on matrix engines”
    - Lack of support for L1 and L2 BLAS: inefficient use of matrix engines









* Zhang et al., Can Tensor Cores Benefit Memory-Bound Kernels? , Under review (can be shared)

Use of INT8 Tensor Cores in explicit voxel finite-element wave simulation

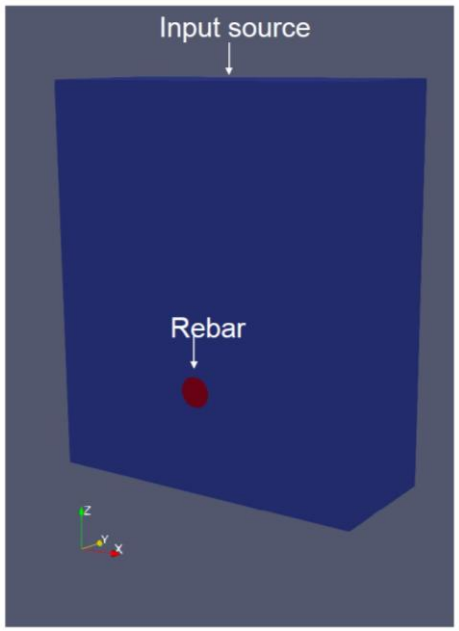

- Developed an algorithm that guarantees FP64-equivalent accuracy even when using integer operations within the **explicit** method, and accelerated simulations with INT8 Tensor Cores
  - Convert element matrix-vector product $\boldsymbol{K}_e\boldsymbol{u}_e$ into products of integer-component matrix and FP64 vectors:
    - $\boldsymbol{K}_e\boldsymbol{u}_e = \frac{\kappa\, ds}{256}\boldsymbol{K}_e^{int}\overline{\boldsymbol{u}}_e + \frac{ds}{3G}\boldsymbol{u}_e$ [**red indicate integer matrix/vector**, **black indicate FP64 matrix/vector**]
  - Convert the main computation part $\boldsymbol{K}_e^{int}\overline{\boldsymbol{u}}_e$ to M sets of integer-valued matrix-vector products $\boldsymbol{K}_e^{int}\overline{\boldsymbol{u}}_{e\mathrm{int}(i)}$ and compute with INT8 Tensor Cores (s, a are FP64 coefficients):
    - $\boldsymbol{K}_e^{int}\overline{\boldsymbol{u}}_e = s\sum_{i=1}^{M} a^i \boldsymbol{K}_e^{int}\overline{\boldsymbol{u}}_{e\mathrm{int}(i)}$
  - Furthermore, data conversion cost is reduced by hierarchical data conversion ($\overline{\boldsymbol{u}}_e \rightarrow \overline{\boldsymbol{u}}_{e\mathrm{int64}(i)} \rightarrow \overline{\boldsymbol{u}}_{e\mathrm{int8}(i,j)}$)
  - FP64-equivalent accuracy obtained by use of $M = 8$ INT8 stages

Computation type	Fraction bits	Value
FP128	112	**507813.690592559616827867910192902549**
FP64	52	**507813.6905925**632
FP32	23	**507813**.750
INT8 ($M = 4$)	28	**507813**.7802133318
INT8 ($M = 8$)	56	**507813.690592559**5

Tsuyoshi Ichimura, Kohei Fujita, Muneo Hori, Maddegedara Lalith: Low-ordered Orthogonal Voxel Finite Element with INT8 Tensor Cores for GPU-based Explicit Elastic Wave Propagation Analysis, *International Conference on Computational Science 2024*

Use of INT8 Tensor Cores in explicit voxel finite-element wave simulation

- Compare the time required to obtain equivalent accuracy@A100 PCIe GPU
  - Use of INT8 Tensor Cores lead to 43.3/9.62 = 4.5-fold speedup in sparse matrix-vector product part and 3.4-fold speedup of total simulation while attaining FP64-equivalent accuracy
  - By use of orthogonal voxel elements with less numerical dispersion in the INT8 Tensor Core accelerated simulation, we can use larger elements compared to standard voxel finite-element, enabling a total of **17-fold speedup**



	Computation type	ds (mm)	Elapsed time (s) Time-step loop	Matrix-vector product
Orthogonal voxel FEM	OVFEM FP64	2.0	48.3	43.3 [2.1 TFLOPS]
	OVFEM FP32	2.0	18.9	15.5 [5.9 TFLOPS]
	TCOVFEM INT8 ($M = 8$)	2.0	14.2	9.62 [64.4 TOPS]
Standard voxel FEM	VFEM FP64	1.2	242.8	-

Tsuyoshi Ichimura, Kohei Fujita, Muneo Hori, Maddegedara Lalith: Low-ordered Orthogonal Voxel Finite Element with INT8 Tensor Cores for GPU-based Explicit Elastic Wave Propagation Analysis, *International Conference on Computational Science 2024*

Integer-arithmetic based sparse linear solver

T. Iwashita (KU, HU, RIKEN), K. Suzuki(KU, HU), T. Fukaya(KU, HU)
KU: Kyoto Univ., HU: Hokkaido Univ.

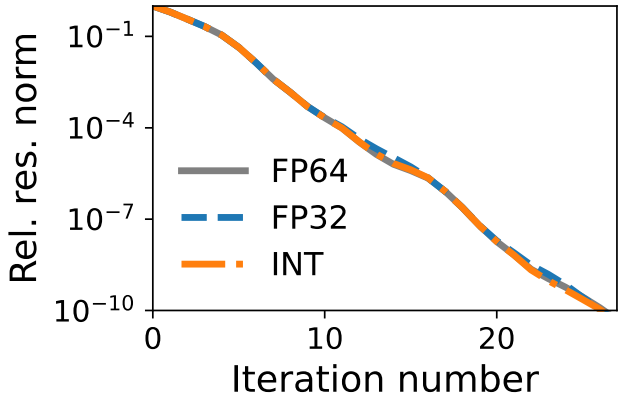
An integer-arithmetic based AMG-FGMRES solver was developed.

- The accuracy of the solution vector is identical to the FP64-based solver.
- Most of computations are performed using integer arithmetic instructions.

Approach

- Iterative refinement method
  - Outer solver: only checking the accuracy of solution
  - Inner solver: an integer arithmetic iterative solver
- Initial scaling of the linear system to be solved
- Division of coefficient matrix to matrices with a different range
- Shift operations for avoiding overflow and underflow
  - Automatic shift amount setting
  - Maximize the accuracy using the property of GMRES

atmosmodd from SuiteSparse



On convergence rate is comparable to FP64, FP32 solvers.

The time to solution is reduced to 2/3 of the FP64 based solver on CPU (GPU version forthcoming)

Papers

K. Suzuki et al., ACM TOMS, 2025 (https://doi.org/10.1145/3704726)
T. Iwashita et al., ScalA20, 2020 (https://doi.org/10.1109/ScalA51936.2020.00006)

京都大学

RAPTOR: Practical Numerical Profiling of Scientific Apps [Domke, Wahib, et. al. SC25]

Motivation:

- Historically, easy choice: **FP32 enough** (→ speedup & lower bandwidth requirements), **or is FP64 necessary**?
- Some numerical methods more amenable to systematic reduction of precision (usually guided by robust mathematical foundation for maintaining accuracy)
- **AI** induced HW trends:
  - **low-precision** units (TF32, BF16, fp16, posits, …) for vector and tensor operations
  - **FP64** capabilities stagnate or are **reduced**

➔ **R&D Question: how to lower precision under lack of mathematical foundation?**

Collaborators: Faveo Hoerold, Ivan Ivanov, Akash Dhruv, William Moses, Anshu Dubey, Mohamed Wahib, Jens Domke

Towards 100x or 'Zettascale' HPC Performance for FugakuNEXT

- **Simulation Workloads ~100x**
  - Raw HW Performance Gain: 10x ~ 20x
  - Mixed precision or emulation: 2x ~ 8x
  - Surrogates / PINN: 10x ~ 25x
  - Total: 100x, some apps 200x ~ 1000x or more over Fugaku
    => 100x or even 'Zettascale'
- **Raw AI HW performance in Zettascale (> 100x)**
  - Low precision, sparsity, new models…
  - Expect 'Zettascale' AI performance
- **With 40MW Limit (not GigaW e.g., hyperscalars)**

Overview of our QC-supercomputer hybrid platform

IBM System 2
"IBM Kobe"
156 qubits
June 2025

Quantiniumm
H1-2
20 qubits => 56 qubits

Wako Campus

QC
Ion-Trap qubits

Quantinuum > 20 qubits Feb, 2025

Riken RQC 'A'
QC 64 qubits

R-CCS (Kobe)

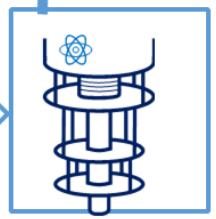
PC/Server

GPU Systems

2130 GPUs
GB200 NVL4
AI+QCHPC

Supercomputer
(Fugaku)

Fast
Low-latency
Network (Infiniband)



QC
Superconductive
qubit

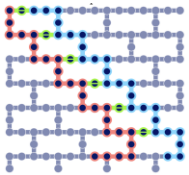
Internet
(SINET)

Supercomputer
Osaka U.

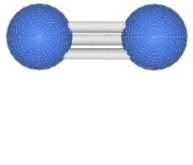
Supercomputer
U. Tokyo

N_2 : Bond breaking on large basis set

Fugaku

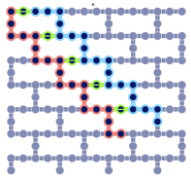


58 qubits

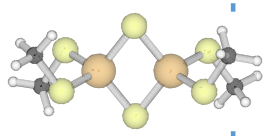


Fe_2S_2: Precision many-body physics

Fugaku

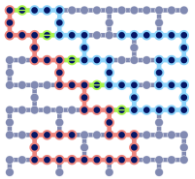


45 qubits

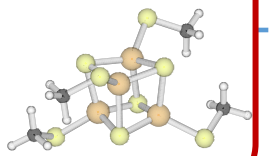


Fe_4S_4: Pushing hardware capabilities

Fugaku



77 qubits



SCA/HPCAsia 2026

Everything with HPC - AI, Cloud, QC and Future Society

Date January 26-29, 2026

Venue Osaka International Convention Center (Osaka, Japan)

acm In-Cooperation

sighpc

- 6 Keynotes – HPC, AI, QC-HPC
- 28 Workshops
- 20 Tutorials
- 101 full papers submitted
- 20 BoFs submitted
- Over 10 Invited Tracks
- ~100 International Exhibitors
- ~3000 participants