Post-K: A Game Changing Supercomputer for Convergence of HPC and Big Data / AI

Satoshi Matsuoka
Director, Riken Center for Computational Science /
Professor, Tokyo Institute of Technology

ADAC Presentation @ ORNL

20190325

Apr 1 2018 Became Director of Riken-CCS:
Science, of Computing, by Computing, and for Computing

Riken Center for Computational Science（R-CCS）
World Leading HPC Research, active collaborations w/Universities, national labs, & Industry

Sci. of Computing

Foundational research on computing in high performance for K, Post-K, and beyond towards the "Post-Moore" era, including future high performance architectures, new computing and programming models, system software, large scale systems modeling, big data analytics, and scalable artificial intelligence / machine learning

Sci. by Computing

Breakthrough Science & Technology using high performance computing capabilities of K, Post-K and beyond to address the issues of high public concern, in areas such as life sciences, climate & environment, disaster prediction & prevention, advanced manufacturing, applications of machine learning for Society 5.0.

High Resolution, High Fidelity Analysis & Simulation

Mutual Synergy

Sci. for Computing

Novel Future High Performance Computing Architectures & Algorithms

New Materials & Electronic Devices e.g., Photonics, Neuromorphics, Quantum, Reconfigurable

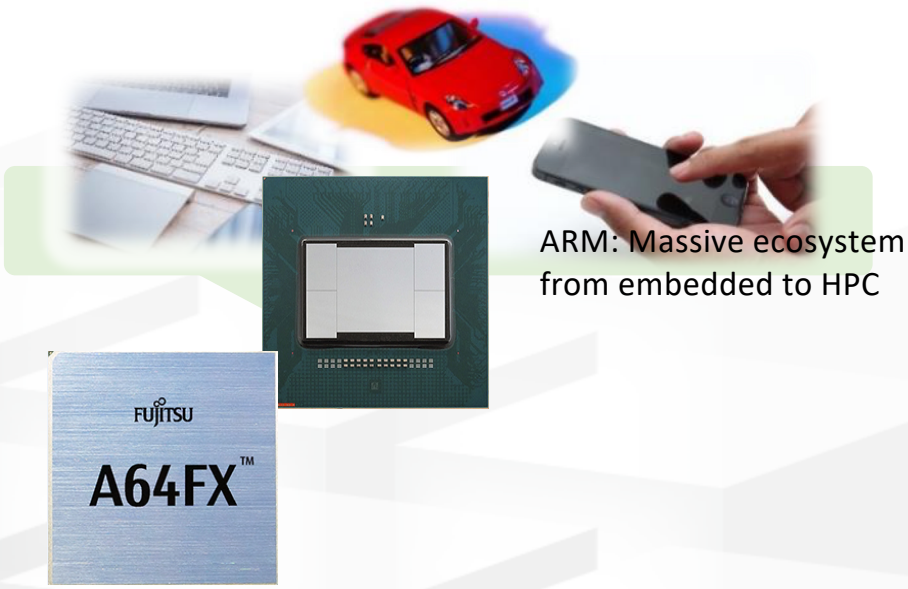
Post-K: The Game Changer

1. **Heritage of the K-Computer, HP in simulation via extensive Co-Design**
- High performance: up to x100 performance of K in real applications
- Multitudes of Scientific Breakthroughs via Post-K application programs
- Simultaneous high performance and ease-of-programming

2. New Technology Innovations of Post-K

- **High Performance, esp. via high memory BW**
  Performance boost by "factors" c.f. mainstream CPUs in many HPC & Society5.0 apps via BW & Vector acceleration
- **Very Green e.g. extreme power efficiency**
  Ultra Power efficient design & various power control knobs
- **Arm Global Ecosystem & SVE contribution**
  Top CPU in ARM Ecosystem of 21 billion chips/year, SVE co-design and world's first implementation by Fujitsu
- **High Perf. on Society5.0 apps incl. AI**
  Architectural features for high perf on Society 5.0 apps based on Big Data, AI/ML, CAE/EDA, Blockchain security, etc.

Global leadership not just in the machine & apps, but as cutting edge IT

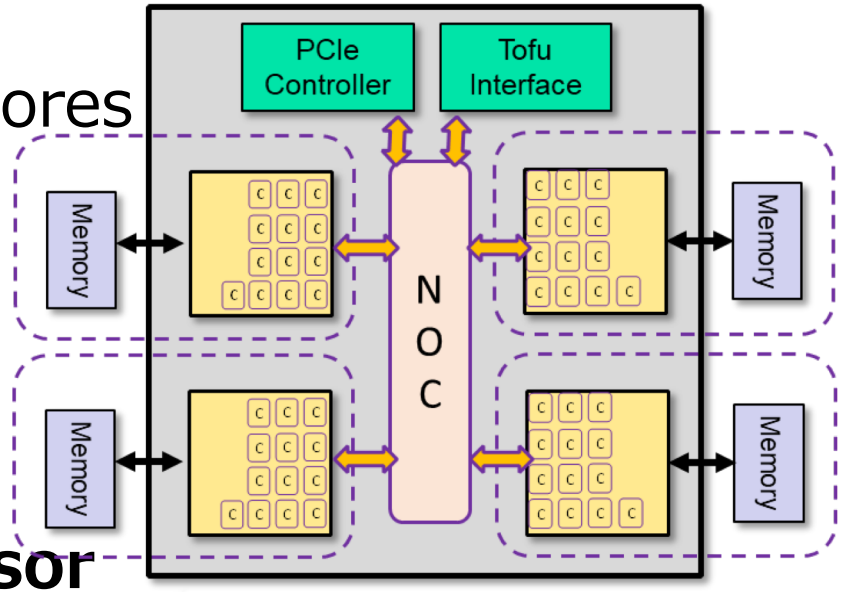


ARM: Massive ecosystem from embedded to HPC

Technology not just limited to Post-K, but into societal IT infrastructures e.g. Clouds

Post K A64fx Processor is…

- **an Many-Core ARM CPU…**
  - 48 compute cores + 2 or 4 assistant (OS) cores
  - Brand new core design
  - Near Xeon-Class Integer performance core
  - ARM V8 --- 64bit ARM ecosystem
  - Tofu-D + PCIe 3 external connection



- **…but also an accelerated GPU-like processor**
  - SVE 512 bit vector extensions (ARM & Fujitsu)
    - Integer (1, 2, 4, 8 bytes) + Float (16, 32, 64 bytes)
  - Cache + scratchpad-like local memory (sector cache)
  - HBM2 on package memory – Massive Mem BW (Bytes/DPF ~0.4)
    - Streaming memory access, strided access, scatter/gather etc.
  - Intra-chip barrier synch. and other memory enhancing features

- **GPU-like High performance in HPC, AI/Big Data, Auto Driving…**

“Post-K” Chronology

(Disclaimer: below includes speculative schedules and subject to change)

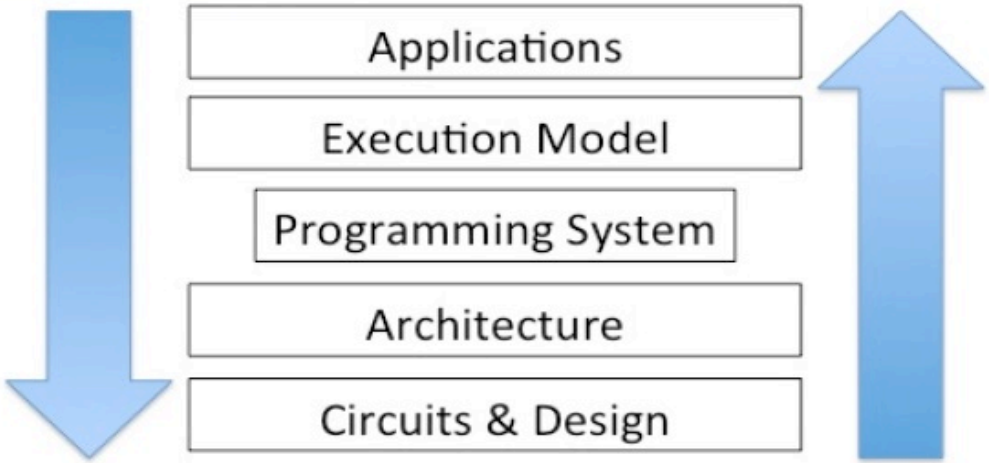
- 1H2019 “Post-K” manufacturing budget approval by the Diet, actual manufacturing commences
- Apr 2019 R-CCS lead research activities on next-gen architectures will commence => whitepaper to be written by Winter
- Aug 2019 End of K-Computer operations
- 4Q2019~1Q2020 “Post-K” installation starts
- 1H2020 “Post-K” preproduction operation starts
- 2020~2021 “Post-K” production operation starts (hopefully)
- And of course we move on…

Watch for announcements on “Post-K” technology commercialization by Fujitsu and its partner vendors RSN

Co-design for Post-K

(slides by Mitsuhisa Sato Team Leader of Architecture Development Team)
Deputy project leader, FLAGSHIP 2020 project
Deputy Director, RIKEN Center for Computational Science (R-CCS)

Analysis of applications to devise the most efficient solutions



Issues and opportunities to exploit

Started in 2009 (before K)

Richard F. BARRETT, et.al. “On the Role of Co-design in High Performance Computing”, *Transition of HPC Towards Exascale Computing*

Co-design from Apps to Architecture

- **Architectural Parameters to be determined**
  - #SIMD, SIMD length, #core, #NUMA node, O3 resources, specialized hardware
  - cache (size and bandwidth), memory technologies
  - Chip die-size, power consumption
  - Interconnect
- **We have selected a set of target applications**
- **Performance estimation tool**
  - Performance projection using Fujitsu FX100 execution profile to a set of arch. parameters.
- **Co-design Methodology (at early design phase)**
  1. Setting set of system parameters
  2. Tuning target applications under the system parameters
  3. Evaluating execution time using prediction tools
  4. Identifying hardware bottlenecks and changing the set of system parameters

Target applications representatives of almost all our applications in terms of computational methods and communication patterns in order to design architectural features.

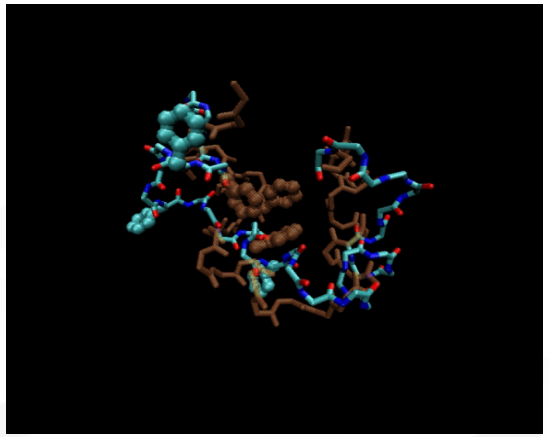
	Target Application	
	Program	**Brief description**
①	GENESIS	MD for proteins
②	Genomon	Genome processing (Genome alignment)
③	GAMERA	Earthquake simulator (FEM in unstructured & structured grid)
④	NICAM+LETK	Weather prediction system using Big data (structured grid stencil & ensemble Kalman filter)
⑤	NTChem	molecular electronic (structure calculation)
⑥	FFB	Large Eddy Simulation (unstructured grid)
⑦	RSDFT	an ab-initio program (density functional theory)
⑧	Adventure	Computational Mechanics System for Large Scale Analysis and Design (unstructured grid)
⑨	CCS-QCD	Lattice QCD simulation (structured grid Monte Carlo)

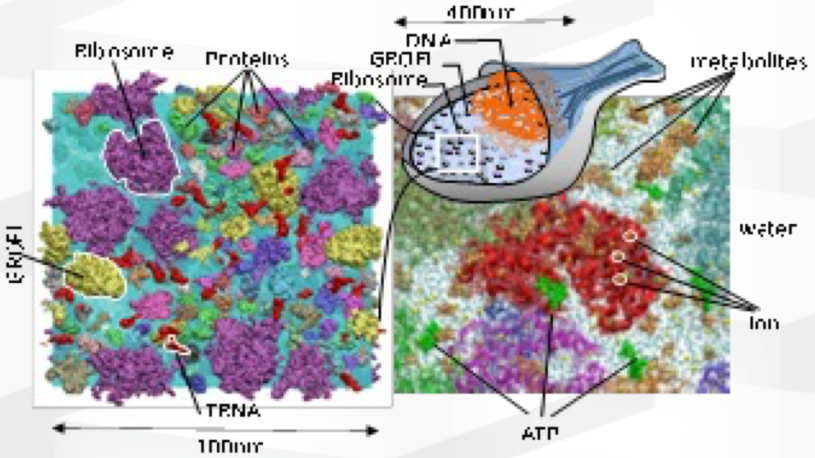
Genesis MD: proteins in a cell environment

Protein simulation before K

- Simulation of a protein in isolation

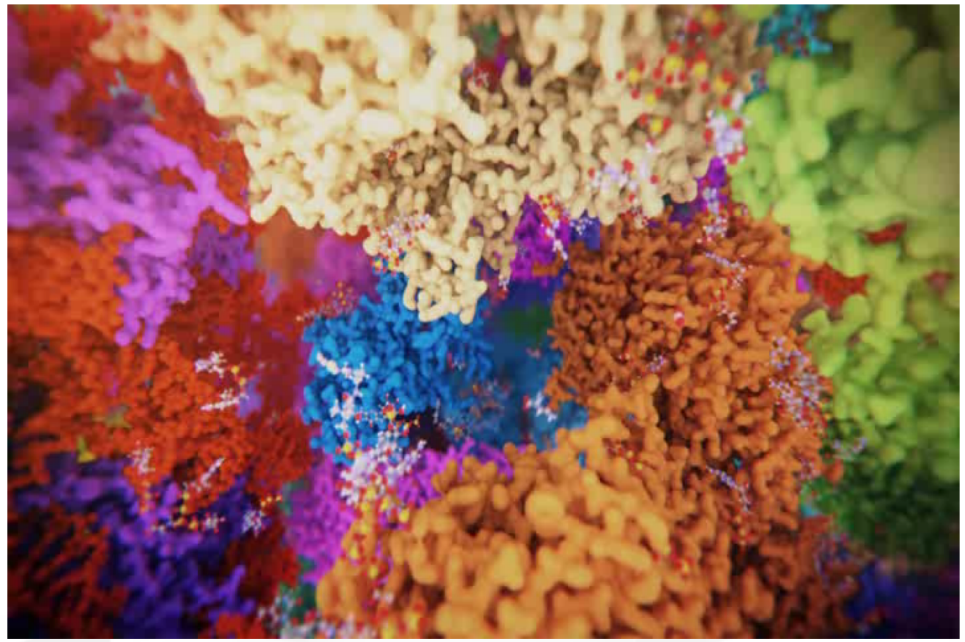
  Folding simulation of Villin, a small protein with 36 amino acids





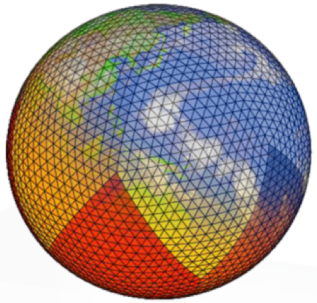
Protein simulation with K

- all atom simulation of a cell interior
- cytoplasm of Mycoplasma genitalium

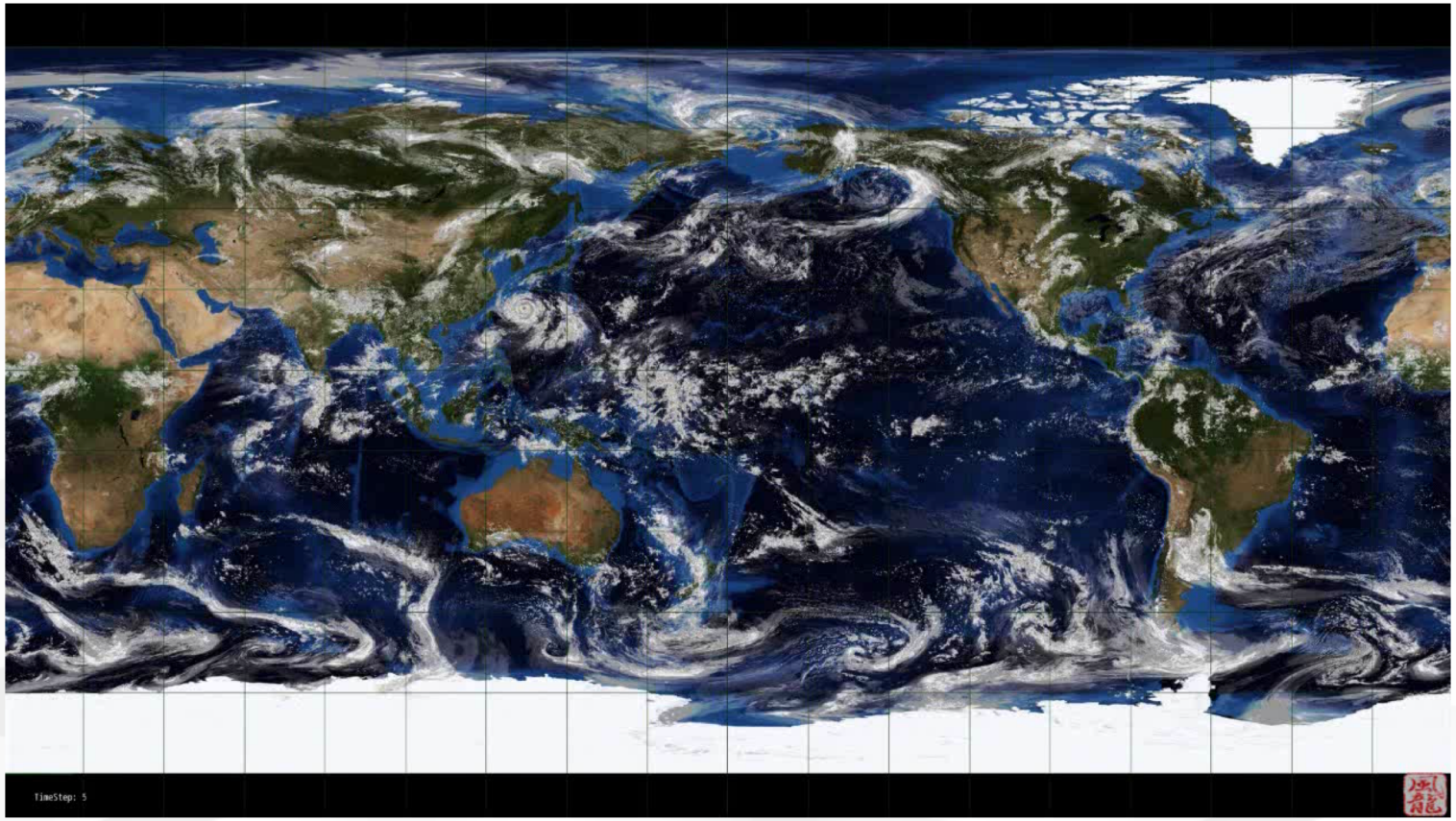


NICAM: Global Climate Simulation

- Global cloud resolving model **with 0.87 km-mesh** which allows resolution of cumulus clouds
- Month-long forecasts of Madden-Julian oscillations in the tropics is realized.

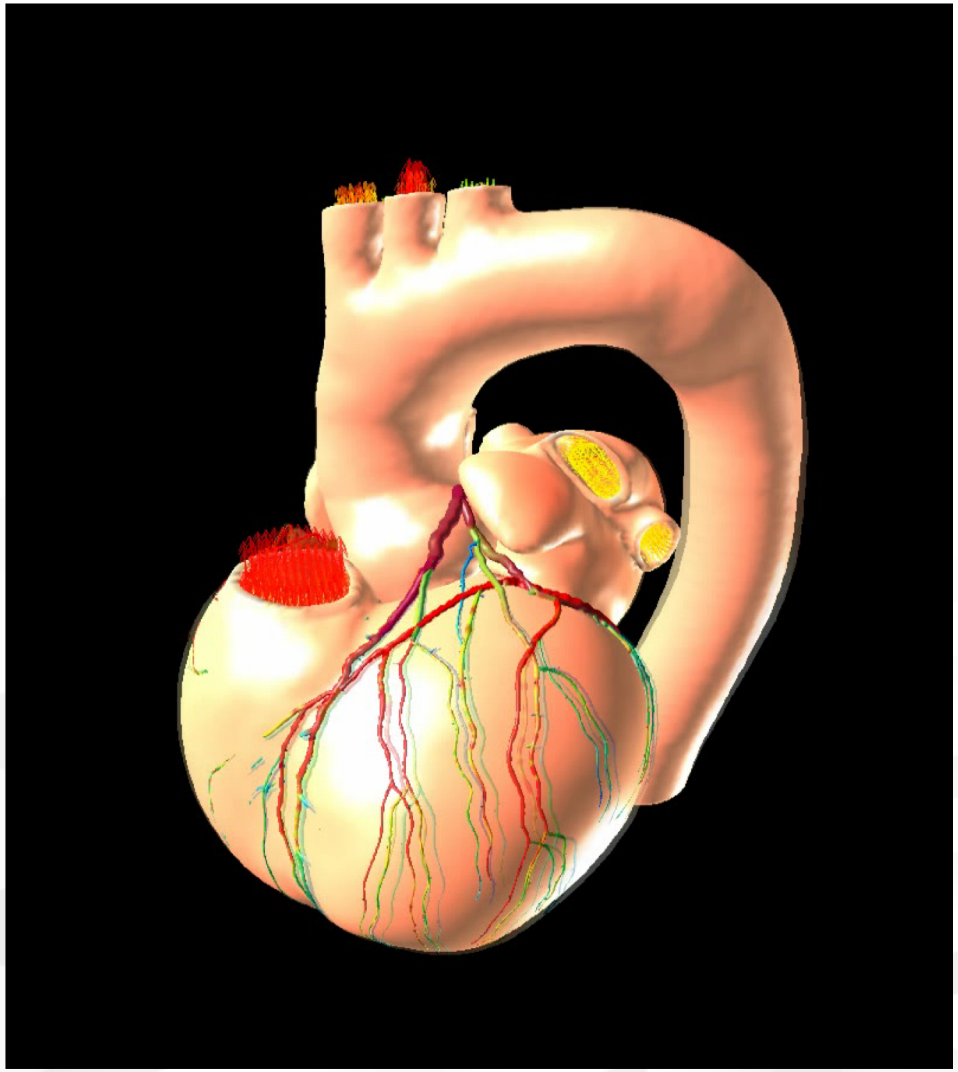


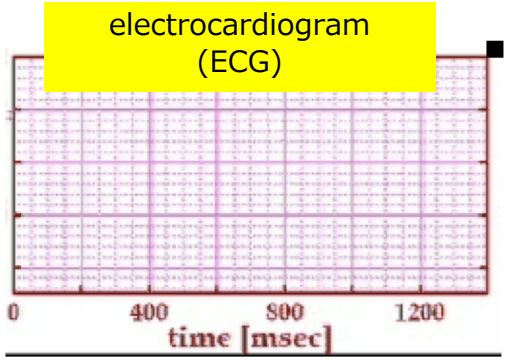
Global cloud resolving model



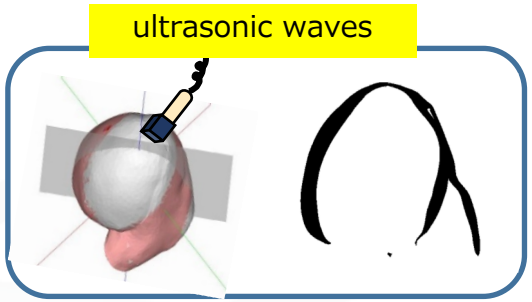
Miyamoto et al (2013) , Geophys. Res. Lett., 40, 4922–4926, doi:10.1002/grl.50944.

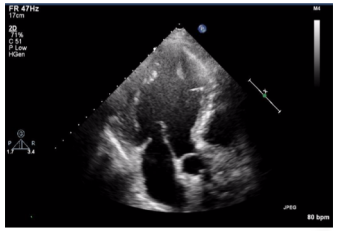
Heart Simulator





- Multi-scale simulator of heart starting from molecules and building up cells, tissues, and heart



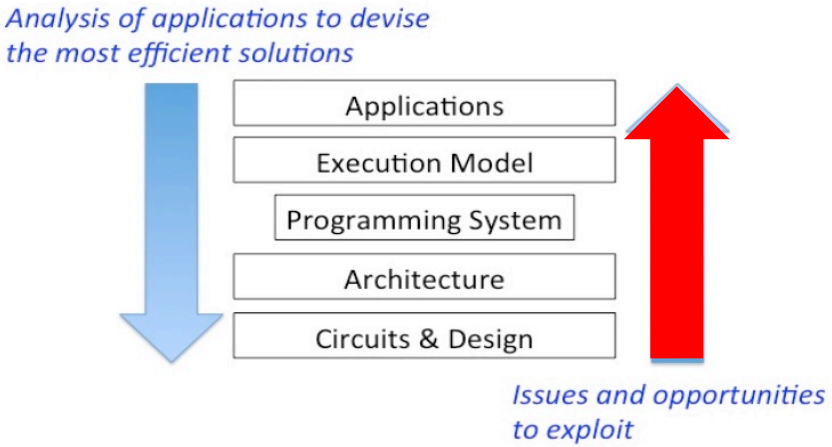


- Heartbeat, blood ejection, coronary circulation are simulated consistently.
- Applications explored
  - congenital heart diseases
  - Screening for drug-induced irregular heartbeat risk

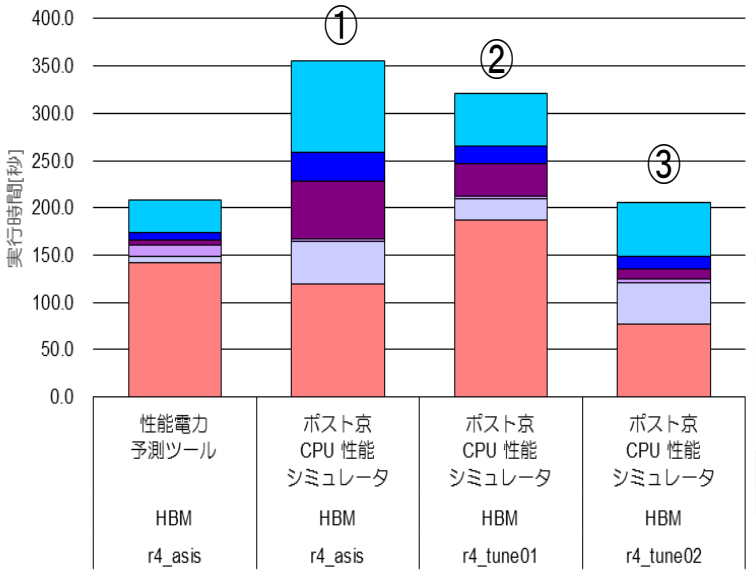
UT-Heart, Inc., Fujitsu Limited

Co-design of Apps for Architecture

- **Tools for performance tuning**
  - Performance estimation tool
    - Performance projection using Fujitsu FX100 execution profile
    - Gives “target” performance
  - **Post-K processor simulator**
    - **Based on gem5, O3, cycle-level simulation**
    - **Very slow, so limited to kernel-level evaluation**



- **Co-design of apps**
  - 1. Estimate “target” performance using performance estimation tool
  - 2. Extract kernel code for simulator
  - 3. Measure exec time using simulator
  - 4. Feed-back to code optimization
  - 5. Feed-back to compiler

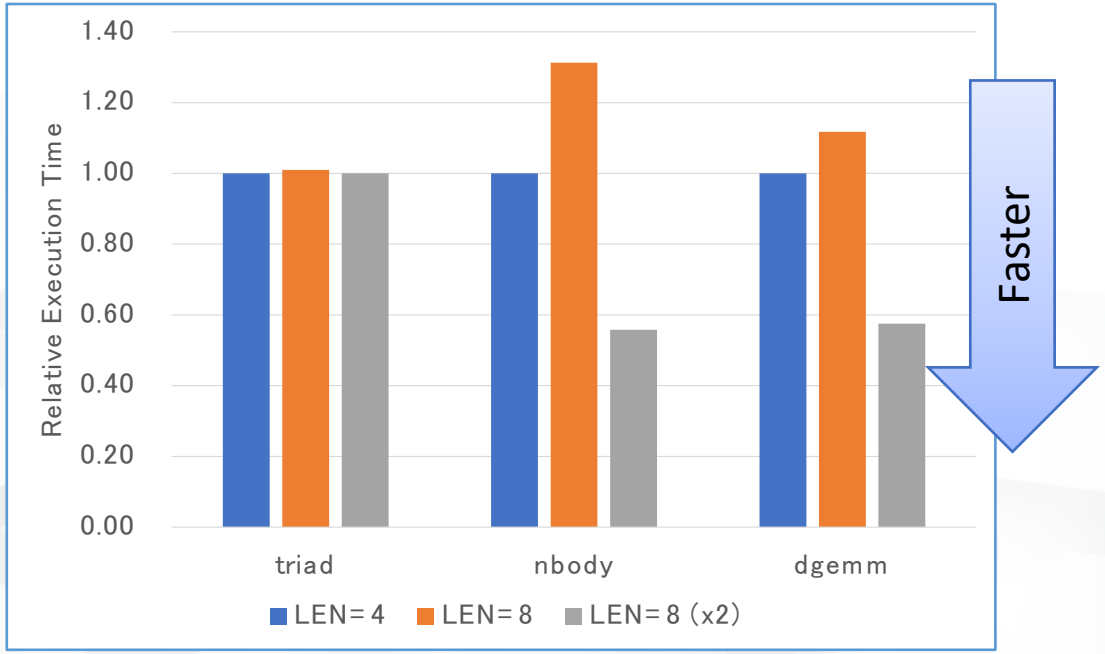


ARM for HPC - Co-design Opportunities

- **ARM SVE Vector Length Agnostic feature is very interesting, since we can examine vector performance using the same binary.**
- **We have investigated how to improve the performance of SVE keeping hardware-resource the same. (in "Rev-A" paper)**
  - ex. "512 bits SVE x 2 pipes" vs. "1024 bits SVE x 1 pipe"
  - Evaluation of **Performance and Power** (in "coolchips" paper) by using our gem-5 simulator (with "white" parameter) and ARM compiler.
  - Conclusion: Wide vector size over FPU element size will improve performance if there are enough rename registers and the utilization of FPU has room for improvement.

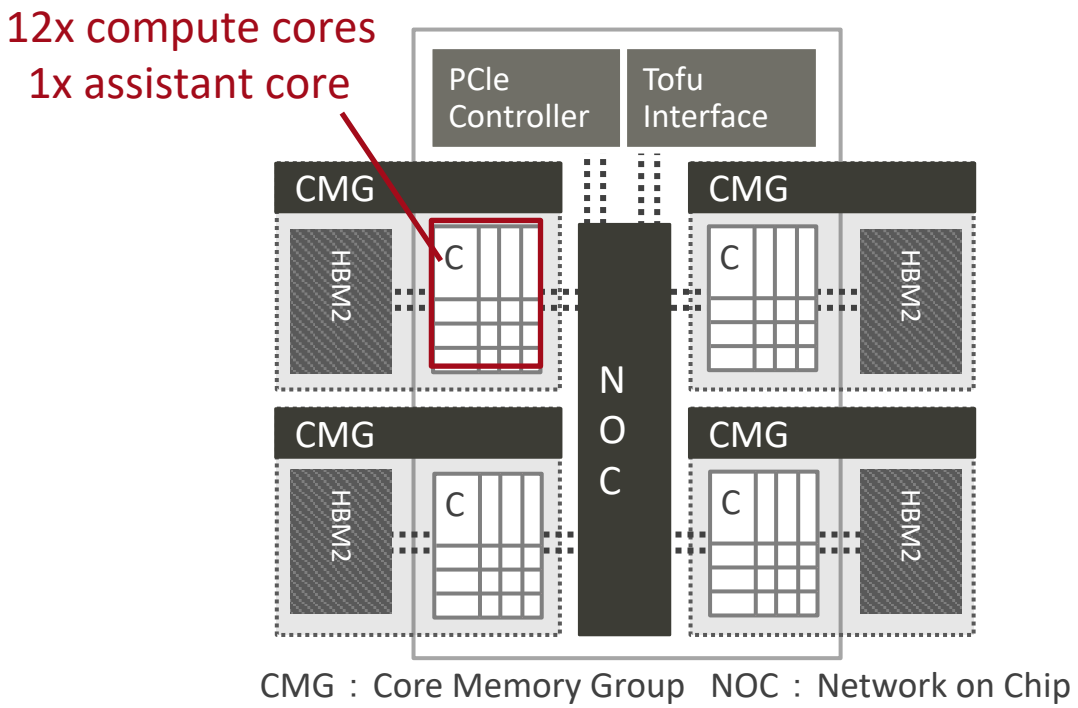
Note that these researches are not relevant to "post-K" architecture.

- Y. Kodama, T. Oajima and M. Sato. "Preliminary Performance Evaluation of Application Kernels Using ARM SVE with Multiple Vector Lengths", In Re-Emergence of Vector Architectures Workshop (Rev-A) in 2017 IEEE International Conference on Cluster Computing, pp. 677-684, Sep. 2017.
- T. Odajima, Y. Kodama and M. Sato, "Power Performance Analysis of ARM Scalable Vector Extension", In IEEE Symposium on Low-Power and High-Speed Chips and Systems (COOL Chips 21), Apr. 2018



A64FX: Summary

- Arm SVE, high performance and high efficiency
  - DP performance 2.7+ TFLOPS, >90%@DGEMM
  - Memory BW 1024 GB/s, >80%@STREAM Triad

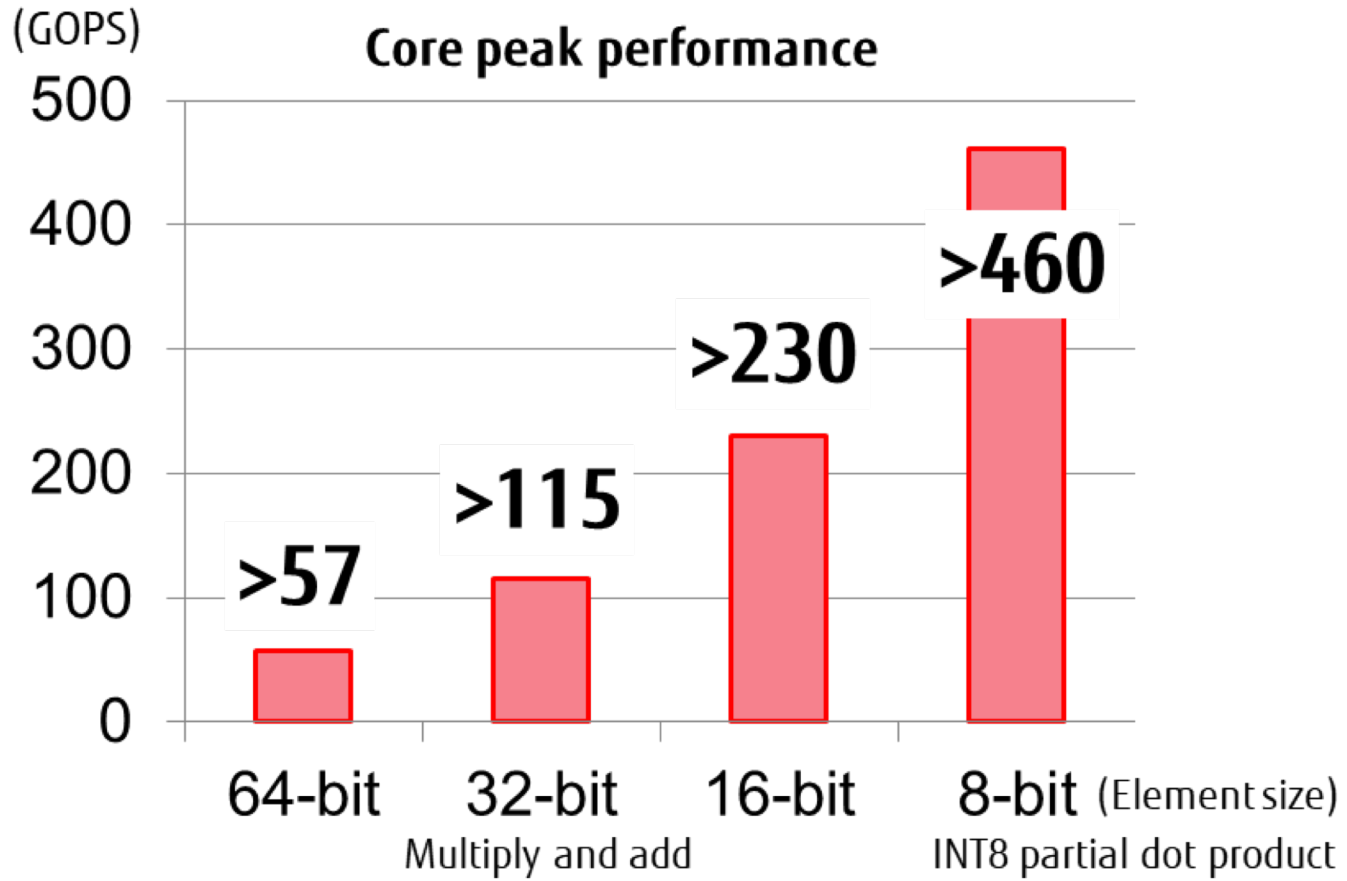


CMG : Core Memory Group NOC : Network on Chip

	A64FX
ISA (Base, extension)	Armv8.2-A, SVE
Process technology	7 nm
Peak DP performance	2.7+ TFLOPS
SIMD width	512-bit
# of cores	48 + 4
Memory capacity	32 GiB (HBM2 x4)
Memory peak bandwidth	1024 GB/s
PCIe	Gen3 16 lanes
High speed interconnect	TofuD integrated

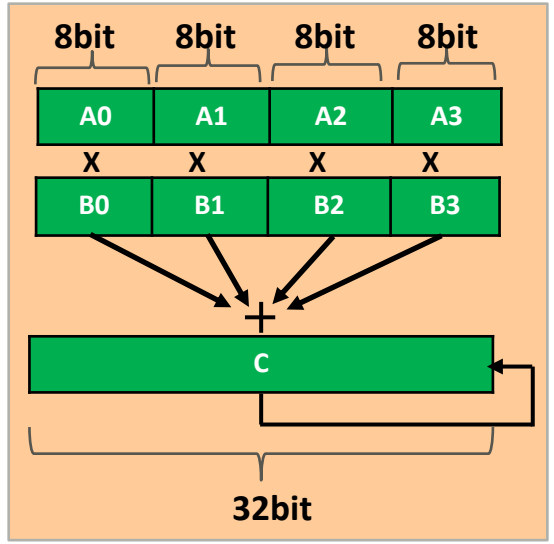
A64FX technologies: Core performance

- High calc. throughput of Fujitsu's original CPU core w/ SVE
  - 512-bit wide SIMD x 2 pipelines and new integer functions



INT8 partial dot product

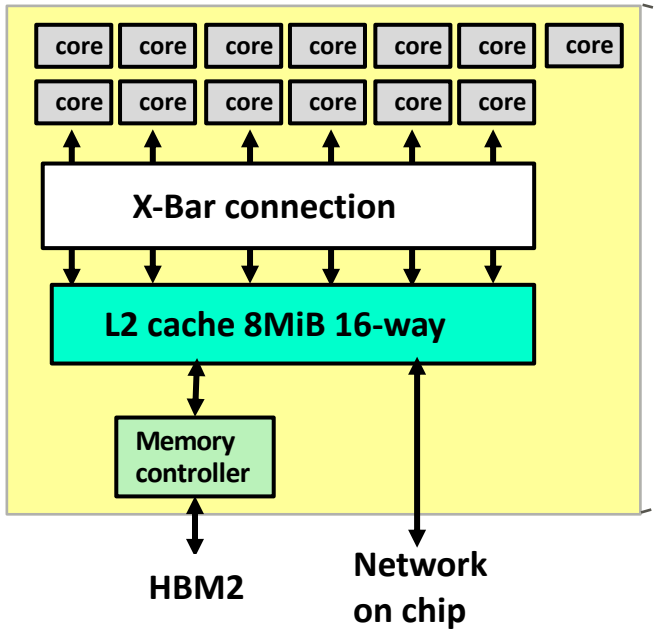
$C = \Sigma\ (Ai \times Bi) + C$



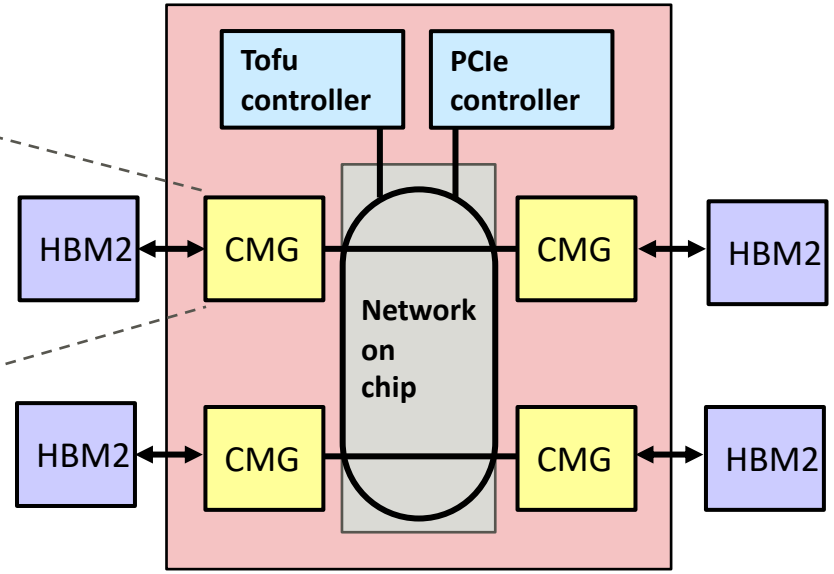
A64FX technologies: Scalable architecture

- Core Memory Group (CMG)
  - 12 compute cores for computing and an assistant core for OS daemon, I/O, etc.
  - Shared L2 cache
  - Dedicated memory controller
- Four CMGs maintain cache coherence w/ on-chip directory
  - Threads binding within a CMG allows linear speed up of cores' performance

CMG configuration



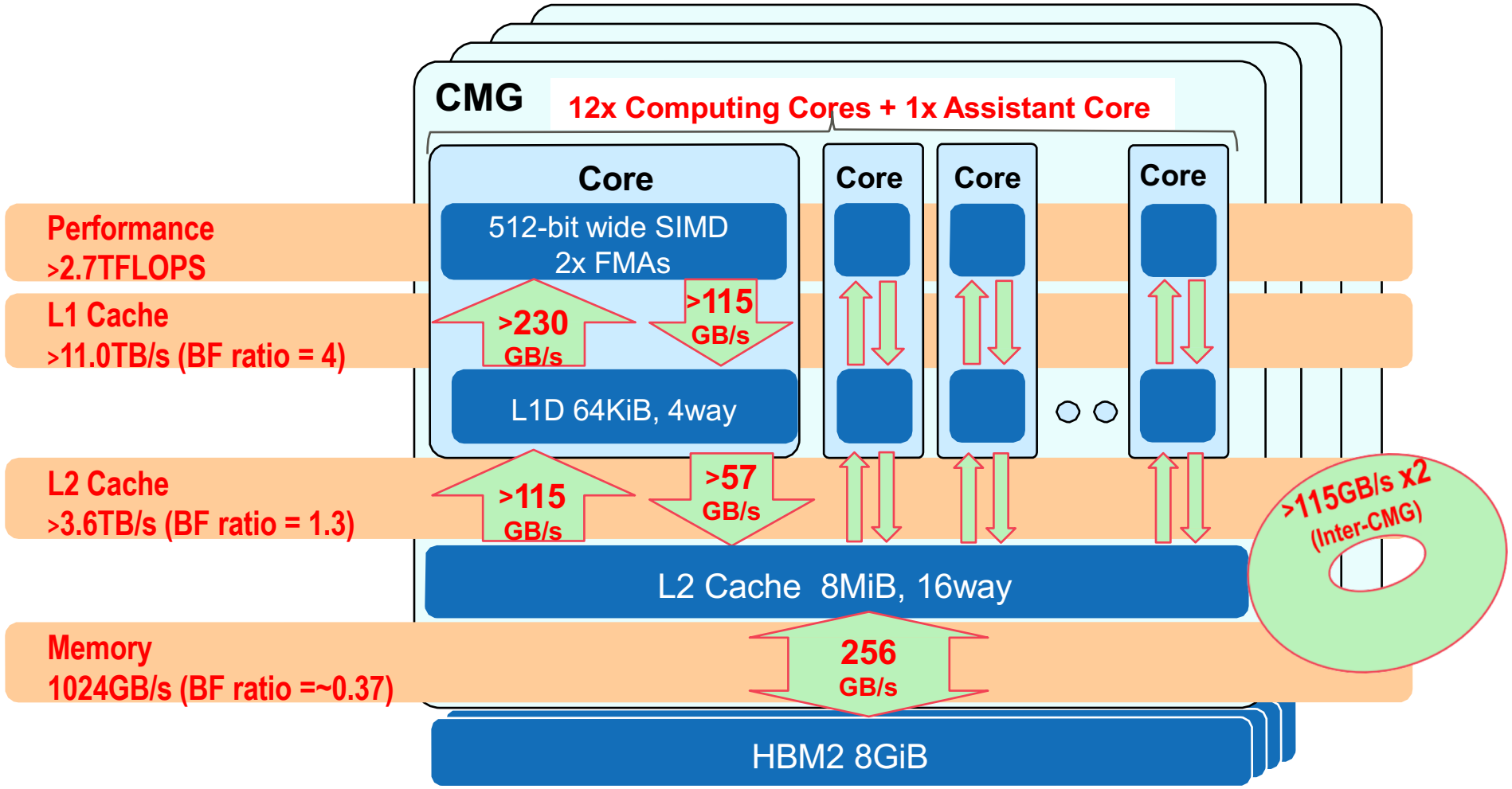
A64FX chip configuration



High Bandwidth

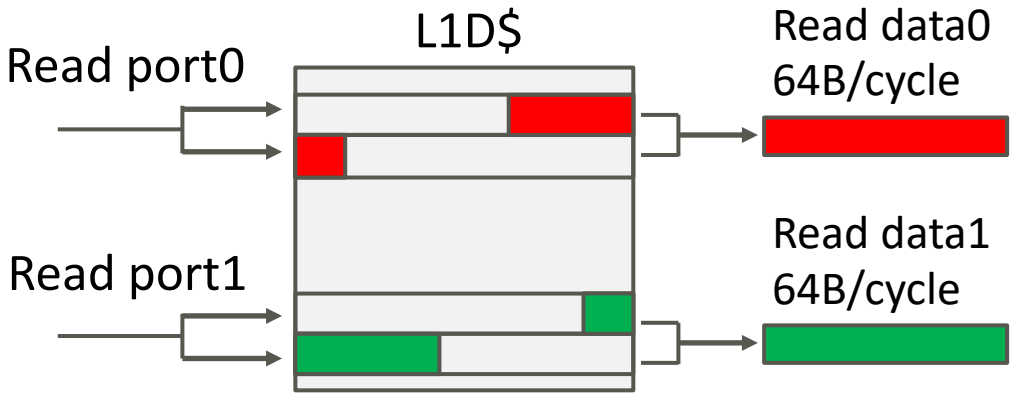
FUJITSU

- Extremely high bandwidth in caches and memory
  - A64FX has out-of-order mechanisms in cores, caches and memory controllers. It maximizes the capability of each layer's bandwidth

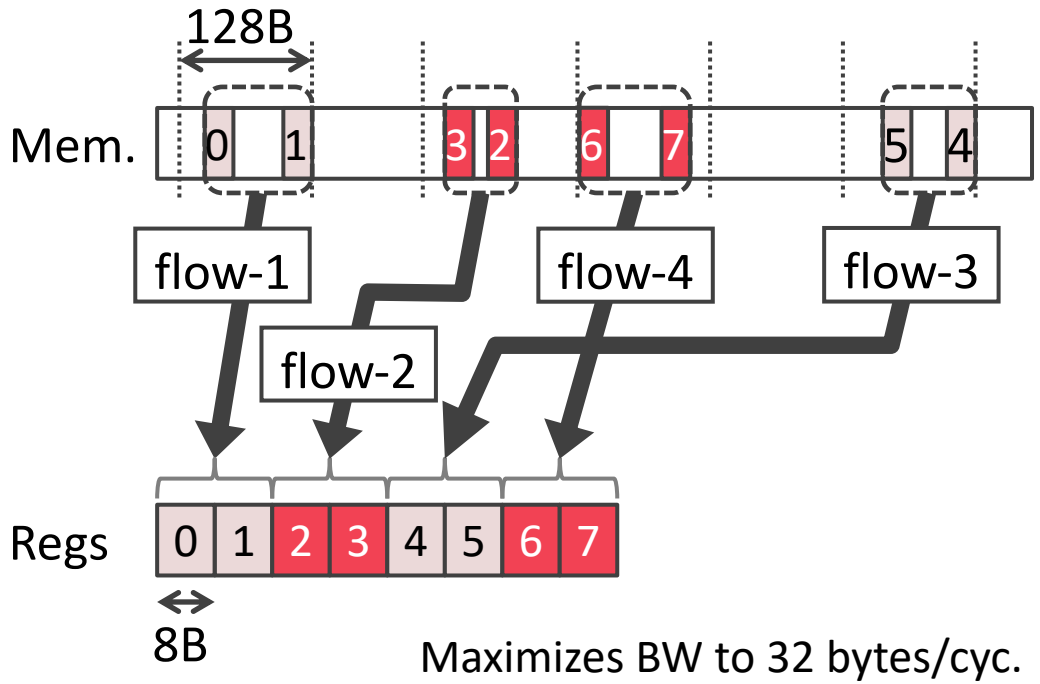


A64FX: L1D cache uncompromised BW

- 128B/cycle sustained BW even for unaligned SIMD load



- “Combined Gather” doubles gather (indirect) load’s data throughput, when target elements are within a “128-byte aligned block” for a pair of two regs, even & odd



Maximizes BW to 32 bytes/cyc.

A64FX: Power monitor and analyzer

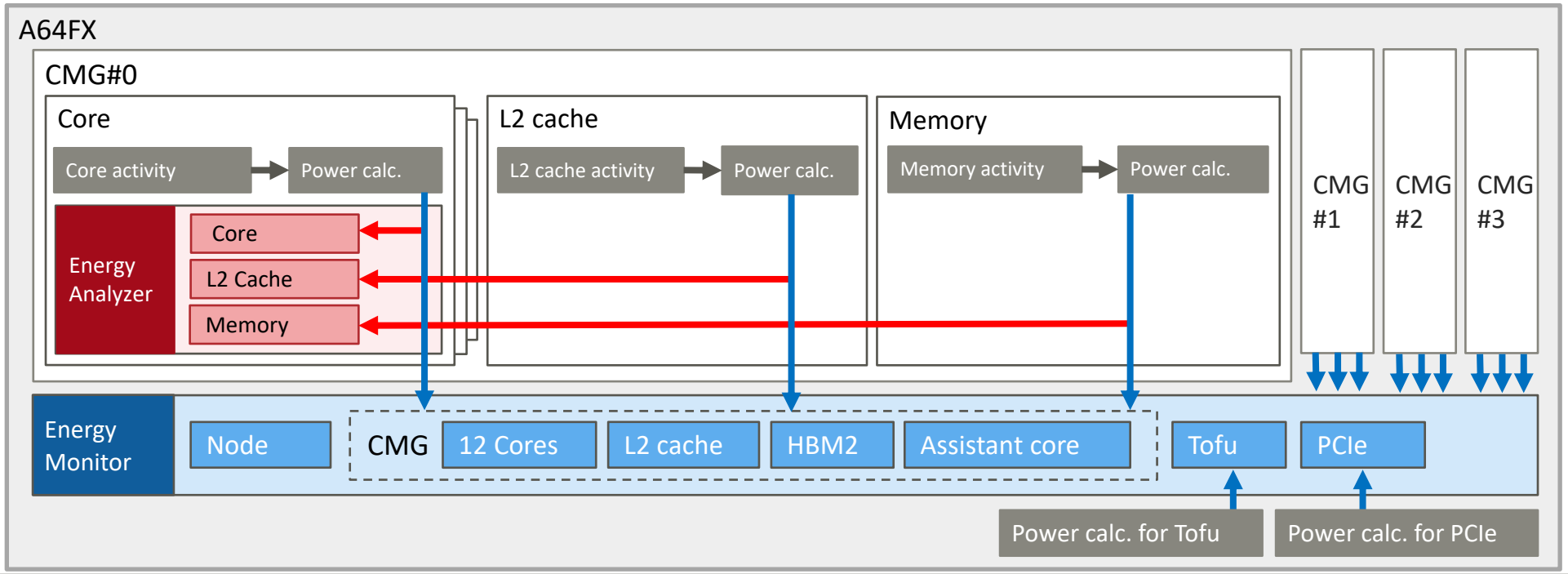
FUJITSU

- Energy monitor (per chip)
  - Node power via Power API*1 (~msec)
  - Averaged power of a node, CMG (cores, L2 cache, memory) etc.

*1: Sandia National Laboratory

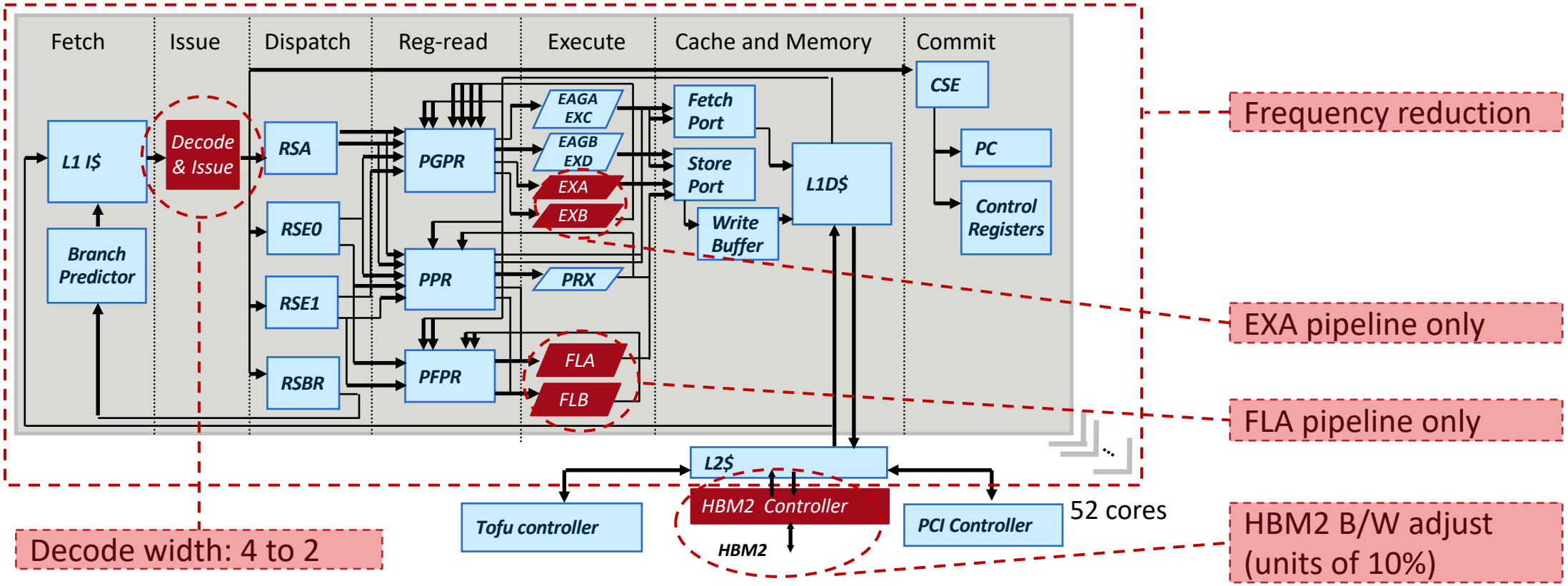
- Energy analyzer (per core)
  - Power profiler via PAPI*2 (~nsec)
  - Fine grained power analysis of a core, L2 cache, and memory

*2: Performance Application Programming Interface



A64FX: Power Knobs to reduce power consumption

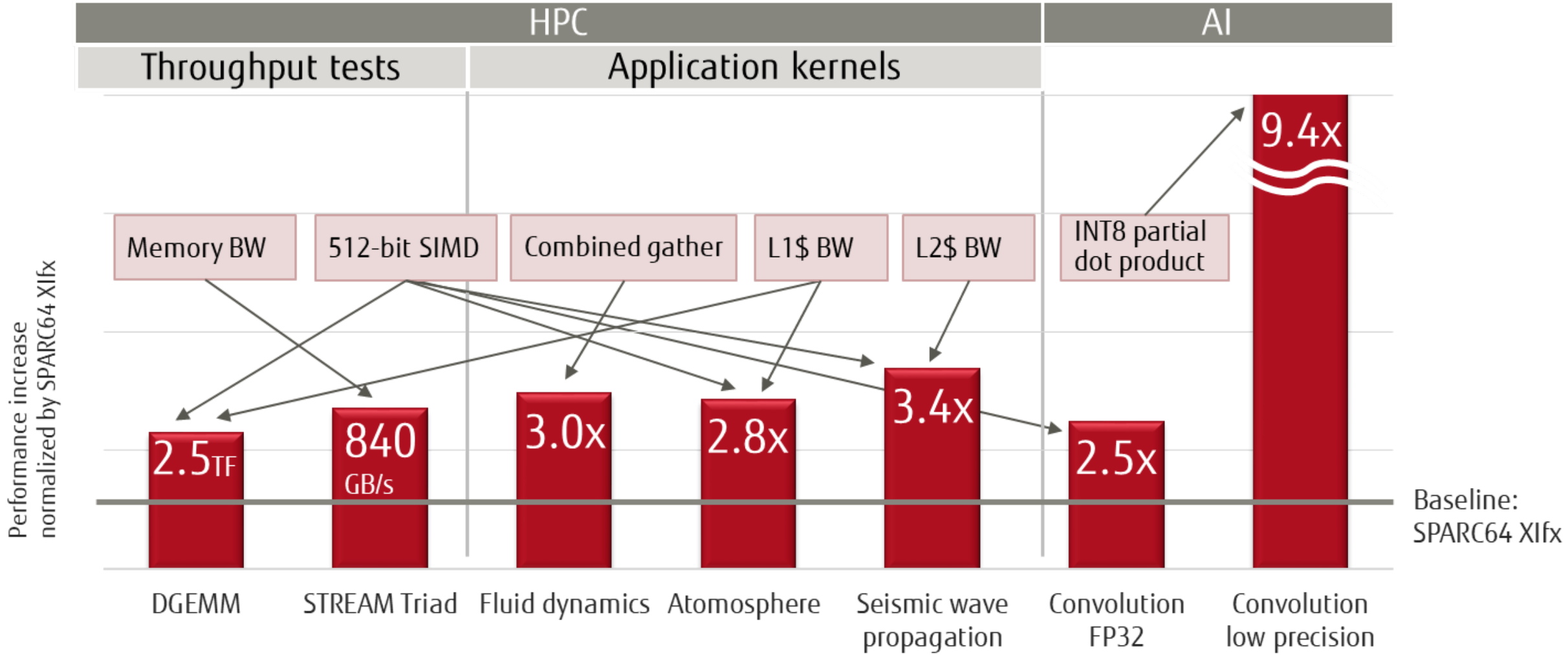
- "Power knob" limits units' activity via user APIs
- Performance/W can be optimized by utilizing Power knobs, Energy monitor & analyzer



Preliminary performance evaluation results

FUJITSU

- Over 2.5x faster in HPC & AI benchmarks than SPARC64 XIfx



Post-K A64fx A0 (ES) performance

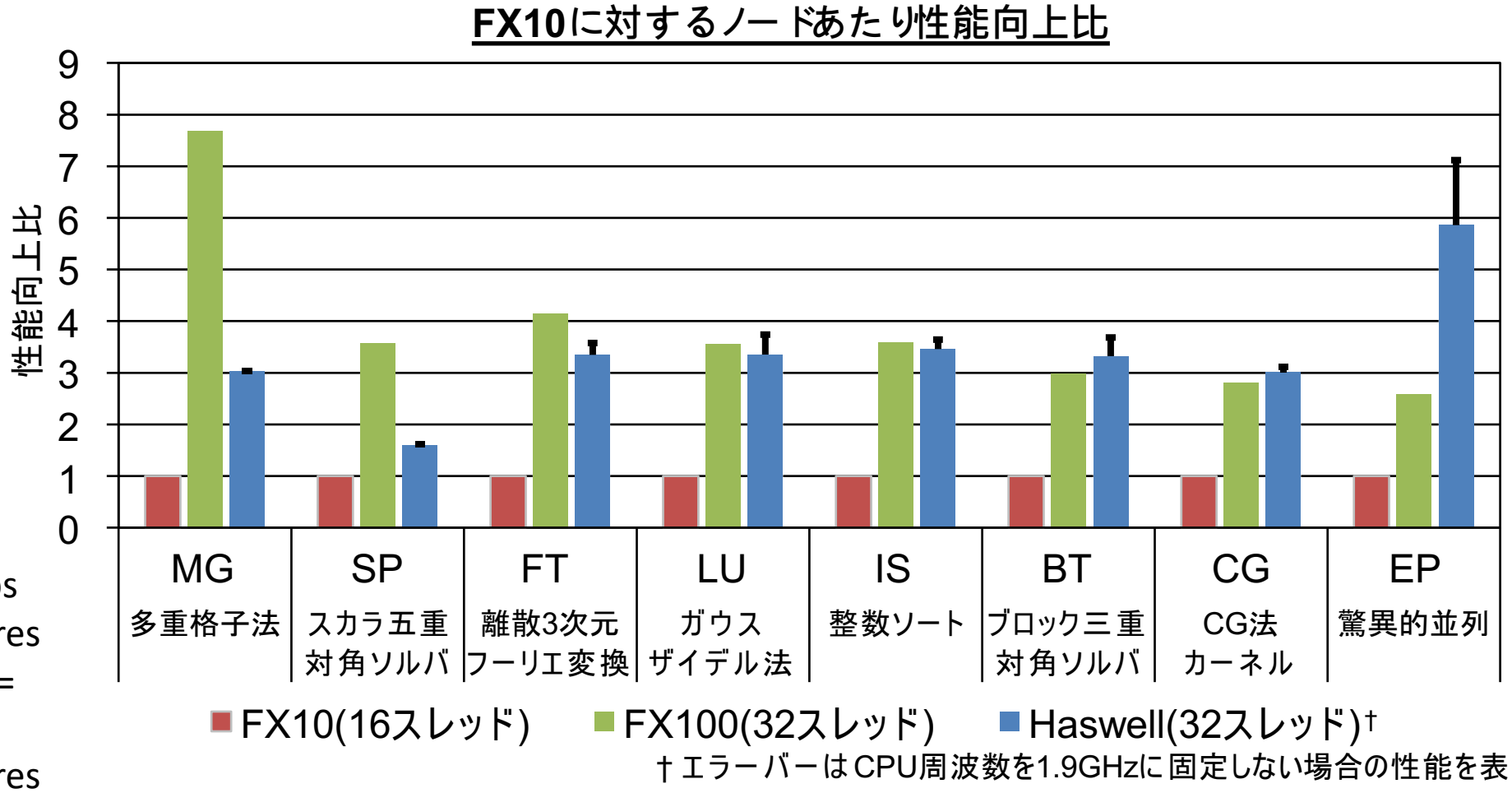
	Performance / CPU					Machine Performance (HPC)		
	Peak TF (DFP)	Peak Mem. BW	Stream Triad	Theoretical B/F	DGEMM Efficiency	Linpack Efficiency	GF/W	Network BW Per Chip
Post-K A64fx (A0 Eng. Sample)	2.764/ 3.072	1024GB/s	840GB/s	0.37/ 0.33	94 %	87.7 %	>15	TOFU-D 40.8GB/s (6.8x 6)
Intel KNL	3.0464	600GB/s	490GB/s	0.20	66%	54.4 %	4.9	12.5 GB/s
Intel Skylake	1.6128	127.8GB/s	97 GB/s	0.08	80 %	66.7 %	4.5	6.2GB/s
NVIDIA V100 (DGX-2)	7.8	900 GB/s	855GB/s	0.12		76 %	15.113	160GB/s 6.2GB/s

NAS Parallel Benchmark of FX100

FUJITSU

[Slide by Ikuo Miyoshi, Fujitsu, SSKen2015]

■ OpenMP版を用いてノードあたり演算性能を評価



Note: Haswell:
1 node = 2 chips
32 threads&cores
FX100: 1 node =
1 chip
32 threads&cores

FX100は、FX10比平均3.9倍、Haswell比平均1.3倍のノードあたり演算性能

使用コード: NAS Parallel Benchmarks Ver. 3.3.1 OpenMP版 クラスC

Fiber (Post-K) MiniApp on FX100

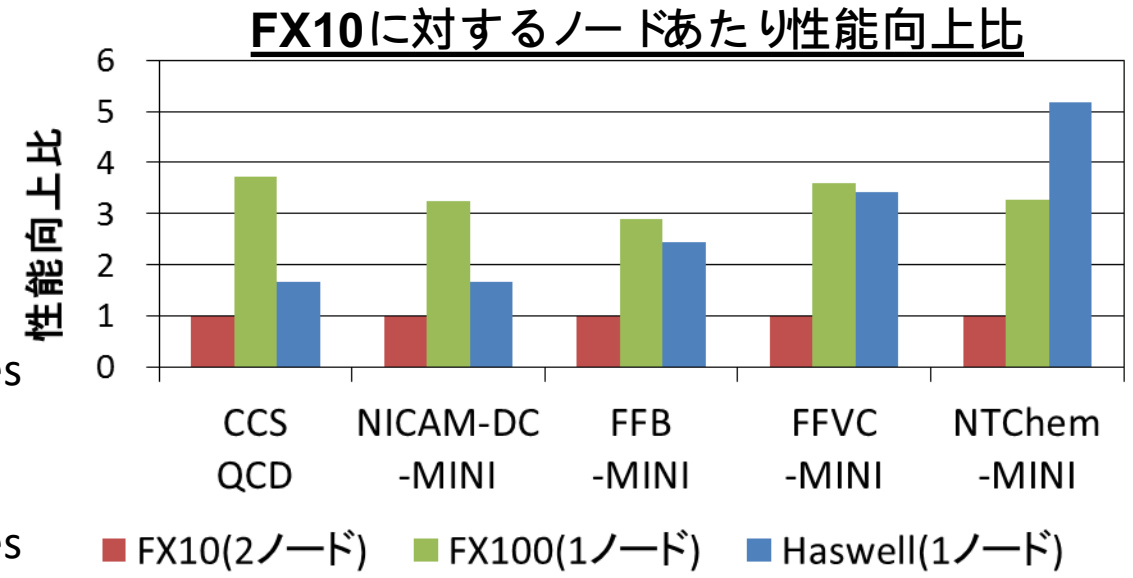
FUJITSU

[Slide by Ikuo Miyoshi, Fujitsu, SSKen2015]

■ 評価条件

アプリ名	問題サイズ	評価区間	スレッド数×プロセス数 (左からFX10、FX100、Haswell)		
CCS QCD	32×32×32×32	BiCGStab	16t×2p	32t×1p	16t×2p
NICAM-DC-MINI	gl05rl00z80pe10	Dynamics	3t×10p		
FFB-MINI	1,048,576要素	MAIN_LOOP	1t×32p	8t×4p	1t×32p
FFVC-MINI	256×256×256	Total	4t×8p		16t×2p
NTChem-MINI	taxol	RIMP2_Driver	1t×32p	16t×2p	2t×16p

■ 測定結果



FX100は、FX10比平均3.3倍、Haswell比平均1.4倍の ノードあたり性能

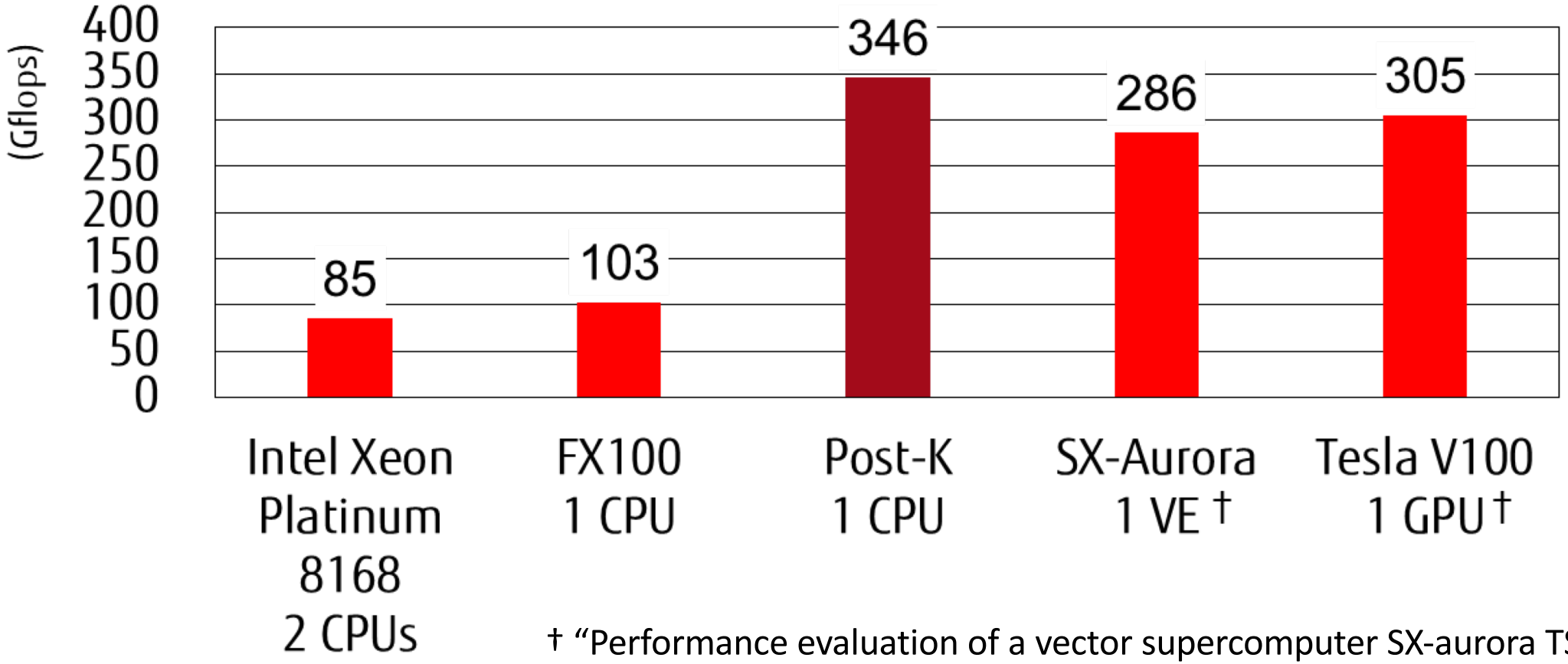
Note: Haswell:
1 node = 2 chips
32 threads&cores
FX100: 1 node =
1 chip
32 threads&cores

注) FFVCには 開発版コンパイラ、 NTChemには 開発版数学ライブラリを使用。 QCDではセクタキャッシュ利用、
FFBではループリローリングのコード変更を実施

Post-K performance evaluation

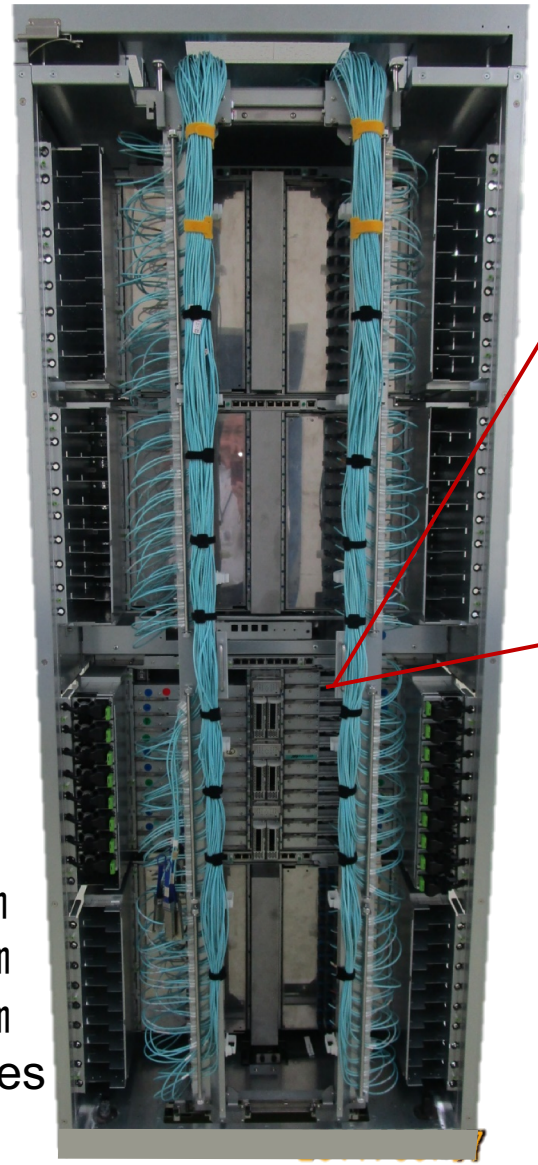
FUJITSU

- Himeno Benchmark (Fortran90)

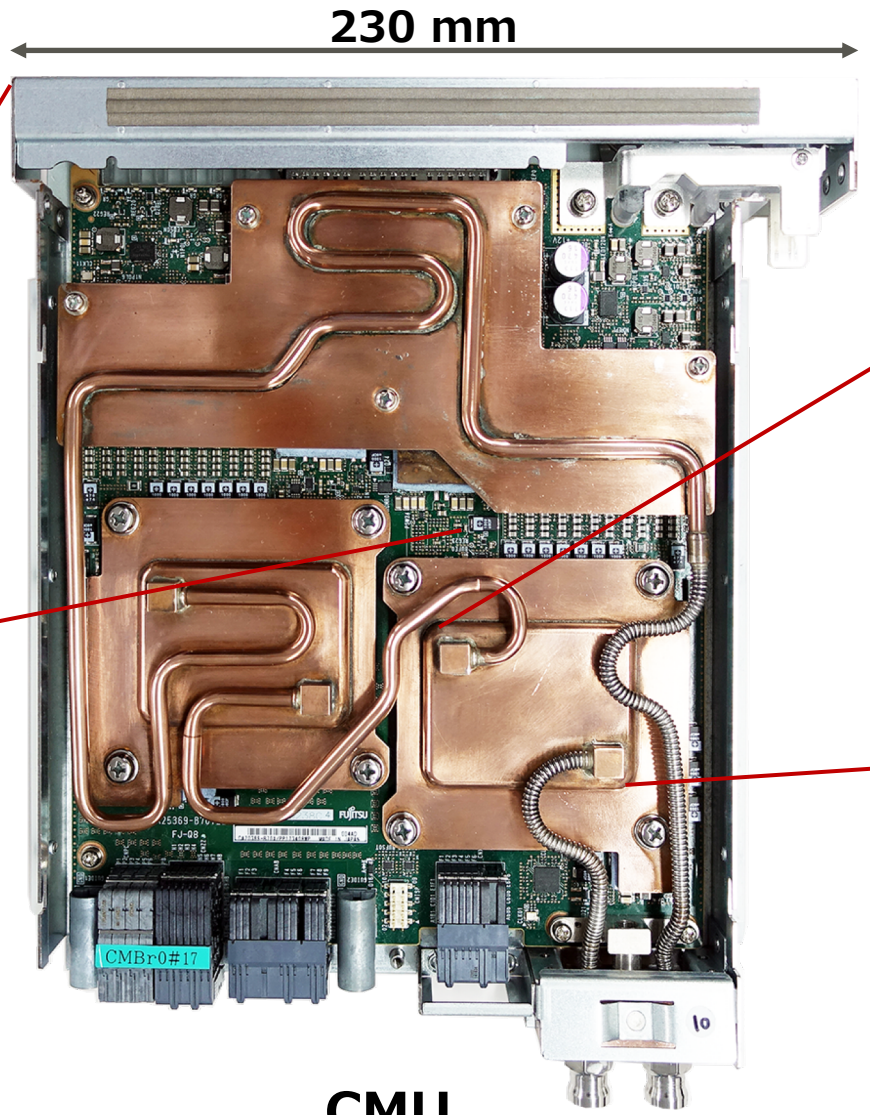


† "Performance evaluation of a vector supercomputer SX-aurora TSUBASA", SC18, https://dl.acm.org/citation.cfm?id=3291728

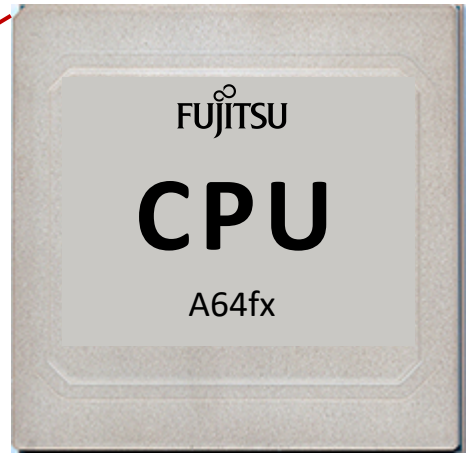
Post-K Chassis, PCB (w/DLC), and CPU Package



W 800mm
D1400mm
H2000mm
384 nodes



CMU

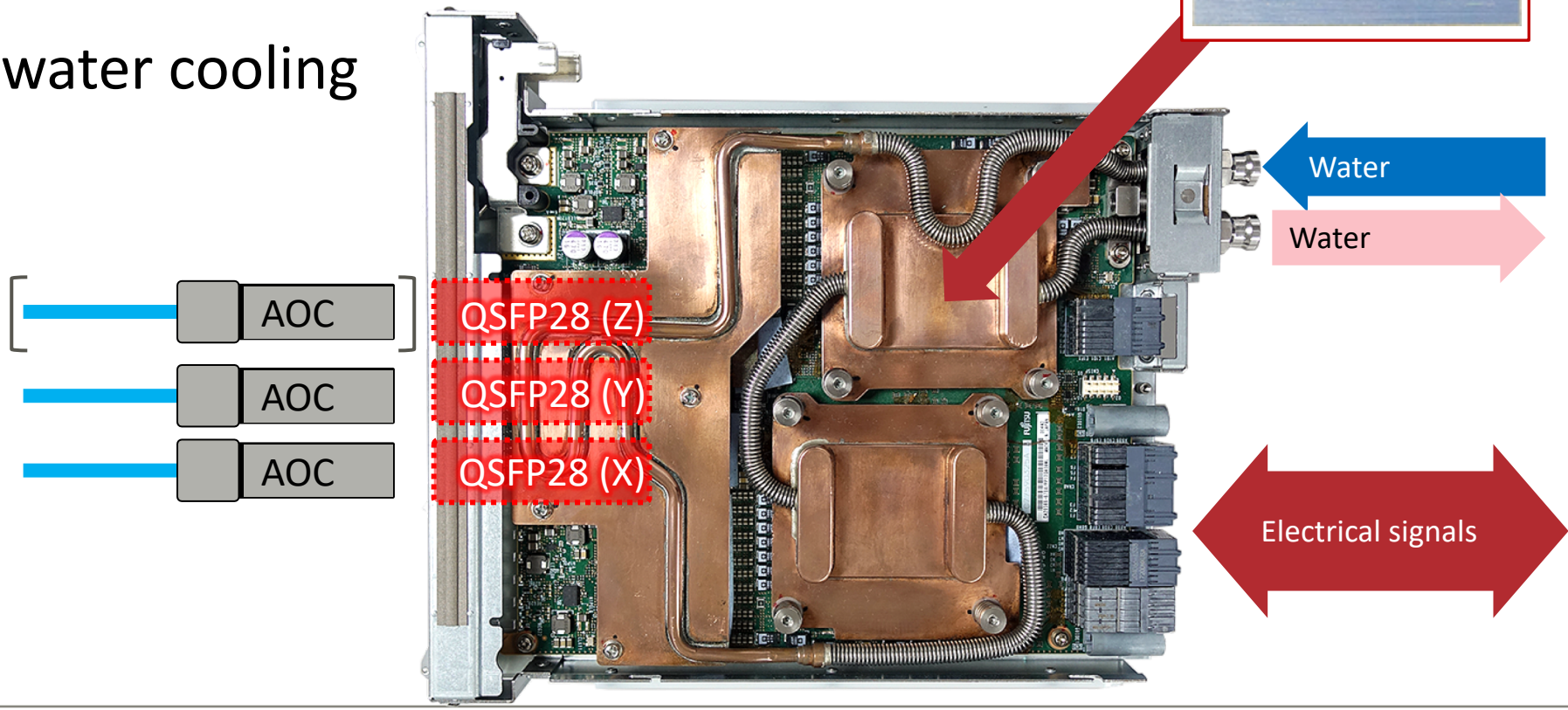


CPU Package

A0 Chip Booted in June
Undergoing Tests

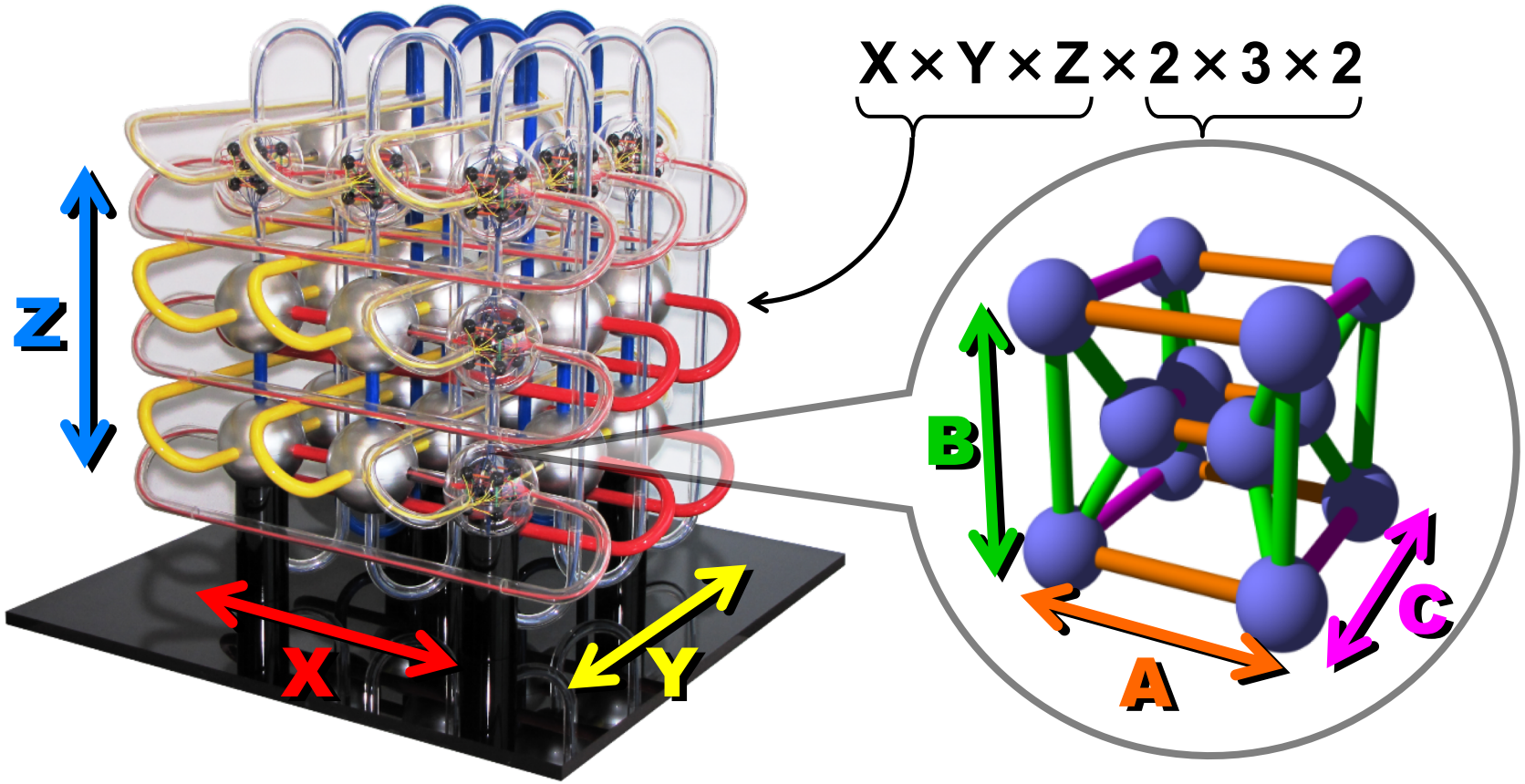
CMU: CPU Memory Unit

- A64FX CPU x2 (Two independent nodes)
- QSFP28 x3 for Active Optical Cables
- Single-side blind mate connectors of signals & water
- ~100% direct water cooling



TOFU-D 6D Mesh/Torus Network

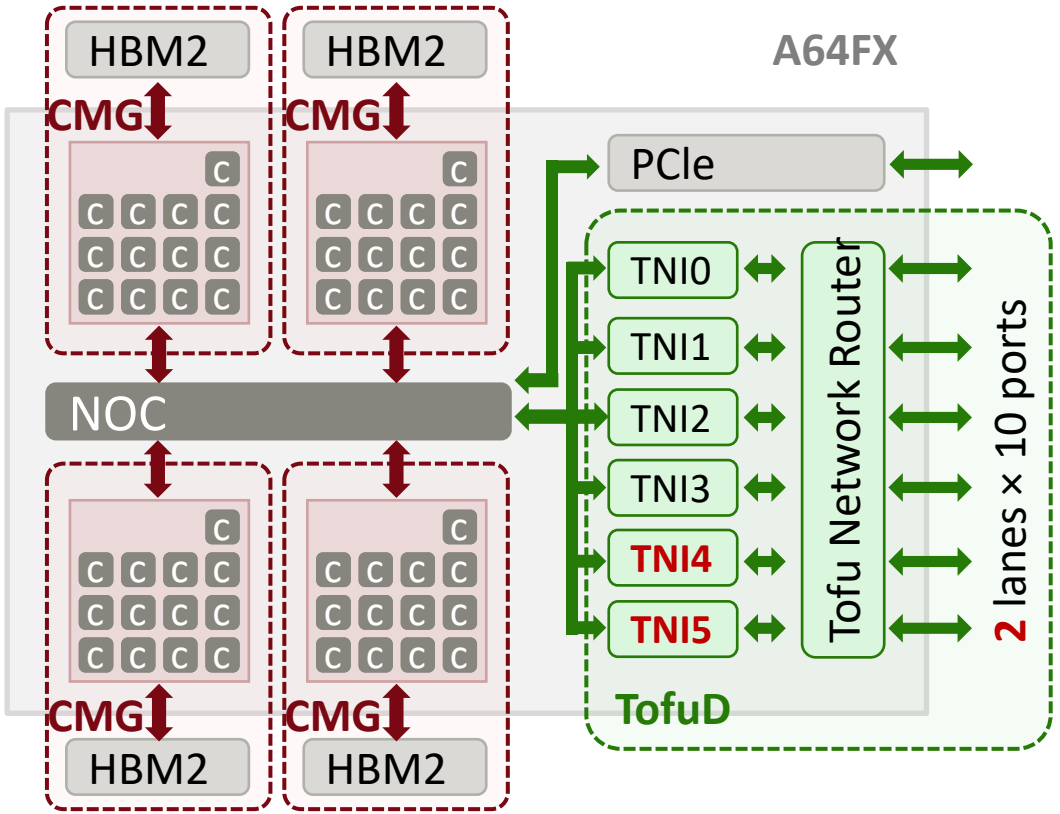
- Six coordinate axes: X, Y, Z, A, B, C
  - X, Y, Z: the size varies according to the system configuration
  - A, B, C: the size is fixed to 2 × 3 × 2
- Tofu stands for “torus fusion”: (X, Y, Z) × (A, B, C)



A64FX: Tofu interconnect D

- Integrated w/ rich resources
  - Increased TNIs achieves higher injection BW & flexible comm. patterns
  - Increased barrier resources allow flexible collective comm. algorithms
- Memory bypassing achieves low latency
  - Direct descriptor & cache injection

	TofuD spec
Port bandwidth	6.8 GB/s
Injection bandwidth	40.8 GB/s
	Measured
Put throughput	6.35 GB/s
Ping-pong latency	0.49~0.54 µs



Put Latencies

FUJITSU

- 8B Put transfer between nodes on the same board
  - The low-latency features were used

	Communication settings	Latency
Tofu1	Descriptor on main memory	1.15 µs
	Direct Descriptor	0.91 µs
Tofu2	Cache injection OFF	0.87 µs
	Cache injection ON	0.71 µs
TofuD	To/From far CMGs	0.54 µs
	To/From near CMGs	0.49 µs

0.20 µs

0.22 µs

- Tofu2 reduced the Put latency by 0.20 µs from that of Tofu1
  - The cache injection feature contributed to this reduction
- TofuD reduced the Put latency by 0.22 µs from that of Tofu2

Injection Rates per Node

FUJITSU

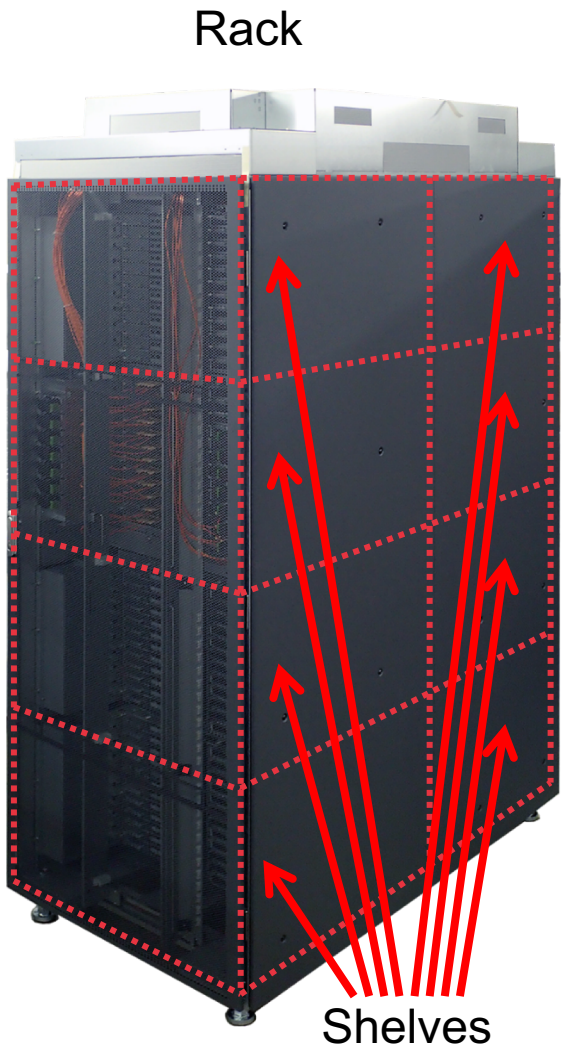
- Simultaneous Put transfers to multiple nearest-neighbor nodes
  - Tofu1 and Tofu2 used 4 TNIs, and TofuD used 6 TNIs

	Injection rate	Efficiency
Tofu1 (K)	15.0 GB/s	77 %
Tofu1 (FX10)	17.6 GB/s	88 %
Tofu2	45.8 GB/s	92 %
TofuD	38.1 GB/s	93 %

- The injection rate of TofuD was approximately 83% that of Tofu2
- The efficiencies of Tofu1 were lower than 90%
  - Because of a bottleneck in the bus that connects CPU and ICC
- The efficiencies of Tofu2 and TofuD exceeded 90 %
  - Integration into the processor chip removed the bottleneck

Packaging – Rack Structure of Post-K

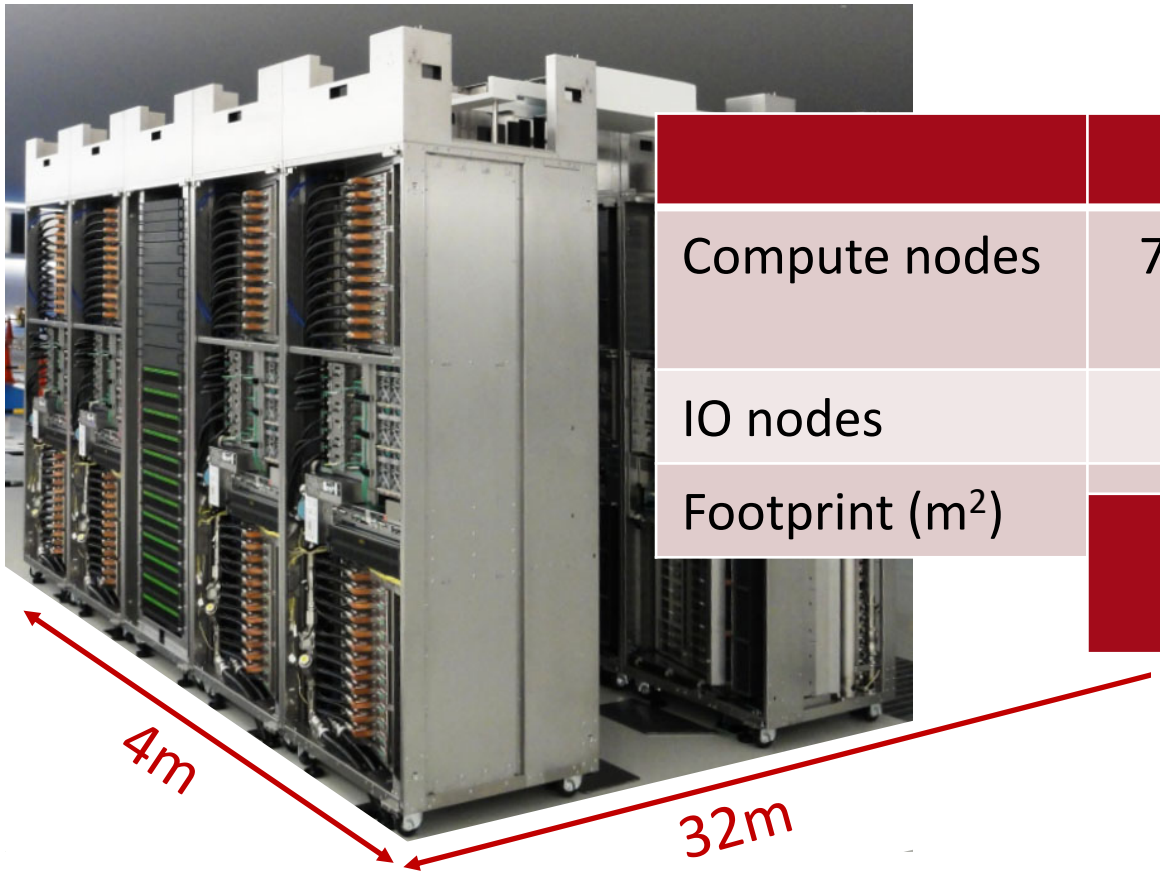
- Rack
  - 8 shelves
  - 192 CMUs or 384 CPUs
- Shelf
  - 24 CMUs or 48 CPUs
  - X × Y × Z × A × B × C = 1 × 1 × 4 × 2 × 3 × 2
- Top or bottom half of rack
  - 4 shelves
  - X × Y × Z × A × B × C = 2 × 2 × 4 × 2 × 3 × 2



1 Peta FLOPS by K computer & Post-K

FUJITSU

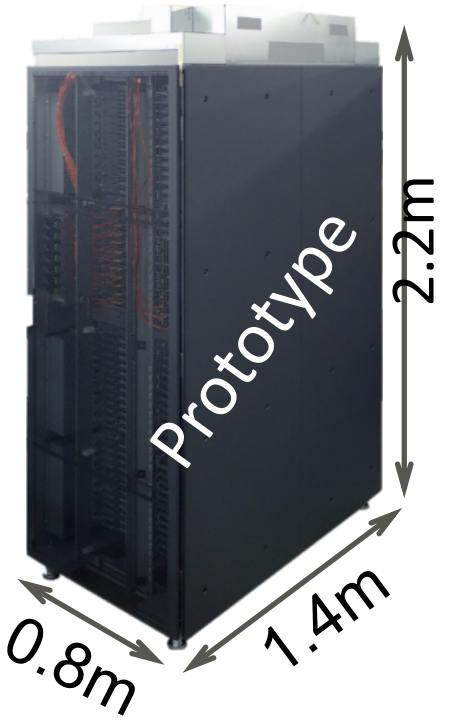
- K computer
  - 80x compute racks & 20x disk racks



- Post-K
  - 1x rack w/ SSDs

	K computer	Post-K
Compute nodes	7,680(=96x80)	384
IO nodes	4,80(=6x80)	
Footprint (m^2)		
	SPARC Linux	**Arm Linux**

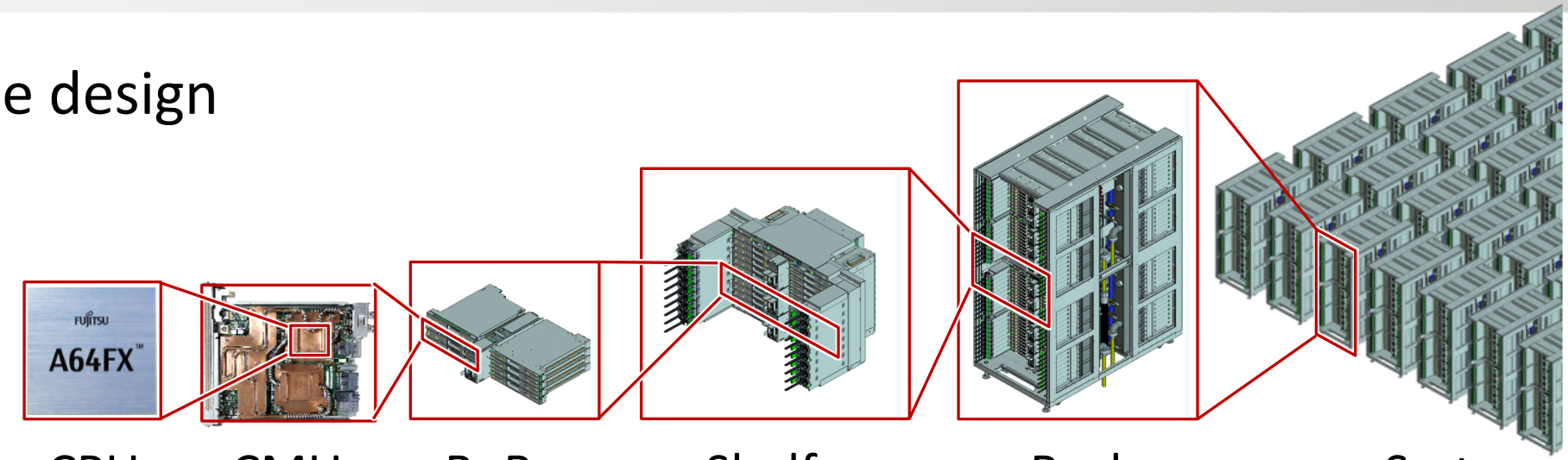
as system software in collaboration with Open Source Community



Post-K system configuration

FUJITSU

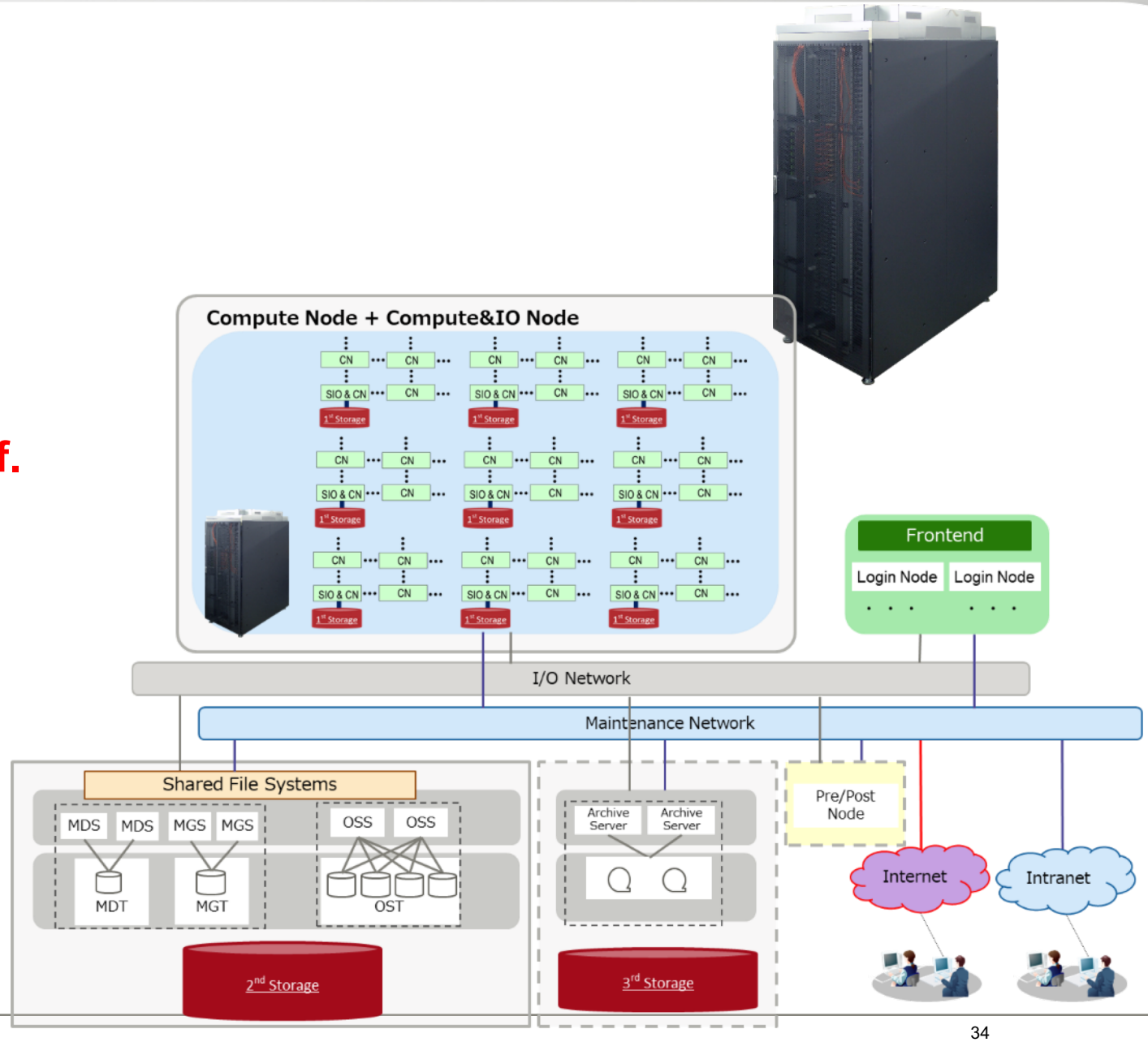
- Scalable design



Unit	# of nodes	Description
CPU	1	Single socket node with HBM2 & Tofu interconnect D
CMU	2	CPU Memory Unit: 2x CPU
BoB	16	Bunch of Blades: 8x CMU
Shelf	48	3x BoB
Rack	384	8x Shelf

Overview of Post-K System & Storage

- **Compute Node, Compute + I/O Node connected by TOFU-D**
- **3-level hierarchical storage**
  - 1st Layer: GFS Cache + Temp FS
  - 2nd Layer: Lustre-based GFS
  - 3rd Layer: Off-site Cloud Storage
- **Full Machine Spec**
  - **>150,000 nodes, ~8 million High Perf. Arm v8.2 Cores**
  - **> 150PB/s memory BW**
  - **> 400 racks**
  - **~40 MegaWatts Machine+IDC PUE ~ 1.1 High Pressure DLC**
  - **~= 15~30 million state-of-the art competing CPU Cores for HPC workloads (both dense and sparse problems)**



Post-K Programming Environment

FUJITSU

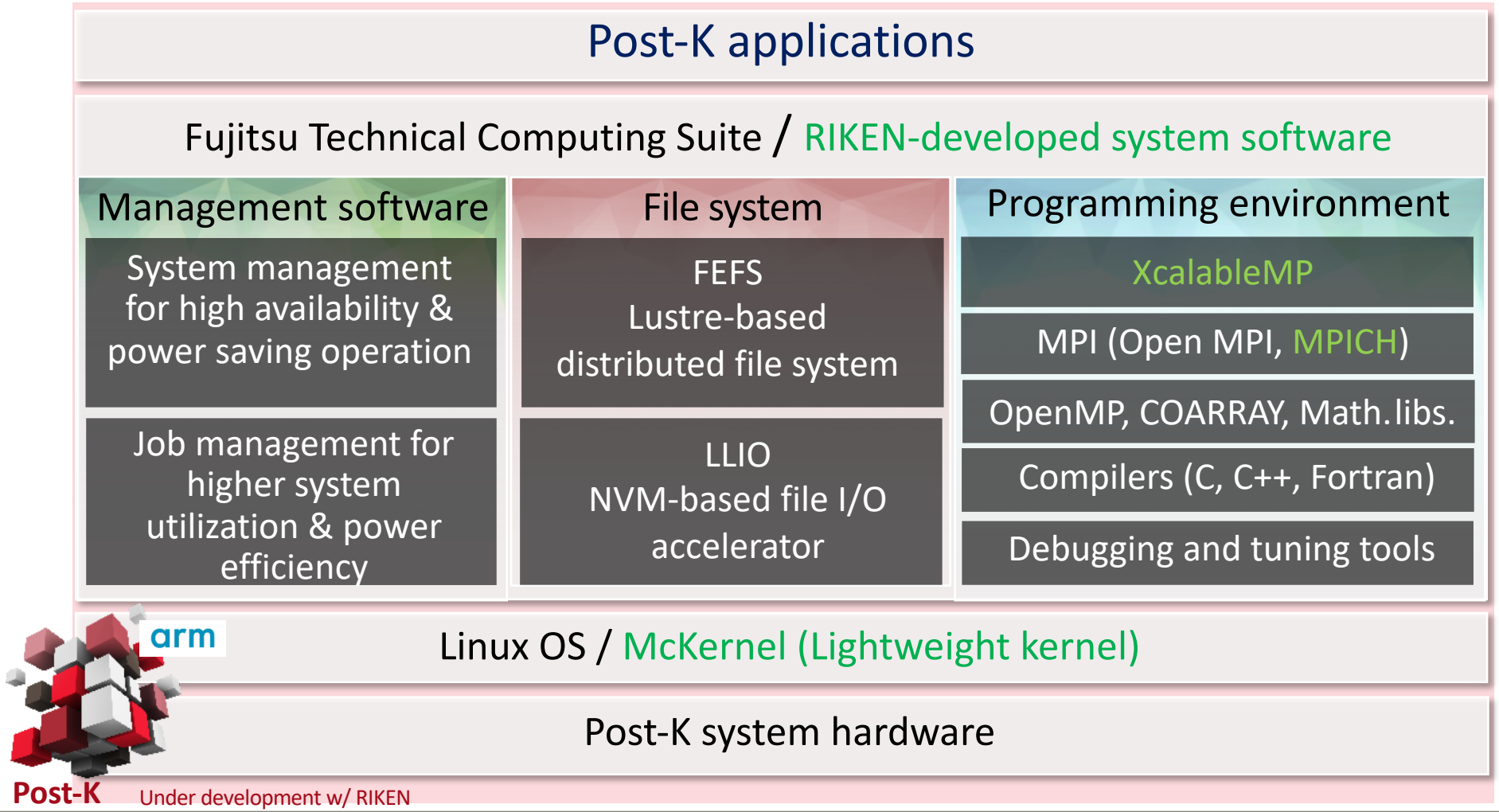
- **Programing Languages and Compilers provided by Fujitsu**
  - Fortran2008 & Fortran2018 subset
  - C11 & GNU and Clang extensions
  - C++14 & C++17 subset and GNU and Clang extensions
  - OpenMP 4.5 & OpenMP 5.0 subset
  - Java
- **Parallel Programming Language & Domain Specific Library provided by RIKEN**
  - XcalableMP
  - FDPS (Framework for Developing Particle Simulator)
- **Process/Thread Library provided by RIKEN**
  - PiP (Process in Process)

GCC and LLVM will be also available

- Script Languages provided by Linux distributor
  - E.g., Python+NumPy, SciPy
- Communication Libraries
  - MPI 3.1 & MPI4.0 subset
    - Open MPI base (Fujitsu), MPICH (RIKEN)
  - Low-level Communication Libraries
    - uTofu (Fujitsu), LLC(RIKEN)
- File I/O Libraries provided by RIKEN
  - Lustre
  - pnetCDF, DTF, FTAR
- Math Libraries
  - BLAS, LAPACK, ScaLAPACK, SSL II (Fujitsu)
  - EigenEXA, Batched BLAS (RIKEN)
- Programming Tools provided by Fujitsu
  - Profiler, Debugger, GUI
- NEW: Containers (Singularity) and other Cloud APIs
- NEW: AI software stacks (w/ARM)

Post-K system software

- RIKEN and Fujitsu are developing a software stack for Post-K



OSS Application Porting @ Arm HPC Users Group

FUJITSU

(http://arm-hpc.gitlab.io/)

Application	Lang.	GCC	LLVM	Arm	Fujitsu
LAMMPS	C++	Modified	Modified	Modified	Modified
GROMACS	C	Modified	Modified	Modified	Modified
GAMESS*	Fortran	Modified	Modified	Modified	Modified
OpenFOAM	C++	Modified	Modified	Modified	Modified
NAMD	C++	Modified	Modified	Modified	Modified
WRF	Fortran	Modified	Modified	Modified	Modified
Quantum ESPRESSO	Fortran	Ok in as is	Ok in as is	Ok in as is	Modified
NWChem	Fortran	Ok in as is	Modified	Modified	Modified
ABINIT	Fortran	Modified	Modified	Modified	Modified
CP2K	Fortran	Ok in as is	Issues found	Issues found	Modified
NEST*	C++	Ok in as is	Modified	Modified	Modified
BLAST*	C++	Ok in as is	Modified	Modified	Modified

OSS Application Porting @ Arm HPC Users Group

FUJITSU

(http://arm-hpc.gitlab.io/)

Application	Lang.	GCC	LLVM	Arm	Fujitsu
LAMMPS	C++	Modified	Modified	Modified	Modified
GROMACS	C	Modified	Modified	Modified	Modified
GAMESS*	Fortran	Modified	Modified	Modified	Modified
OpenFOAM					odified
NAMD					odified
WRF					odified
Quantum ESPRESSO	Fortran	Ok in as is	Ok in as is	Ok in as is	Modified
NWChem	Fortran	Ok in as is	Modified	Modified	ongoing
ABINIT	Fortran	Modified	Modified	Modified	Modified
CP2K	Fortran	Ok in as is	Issues found	Issues found	ongoing
NEST*	C++	Ok in as is	Modified	Modified	Modified
BLAST*	C++	Ok in as is	Modified	Modified	Modified

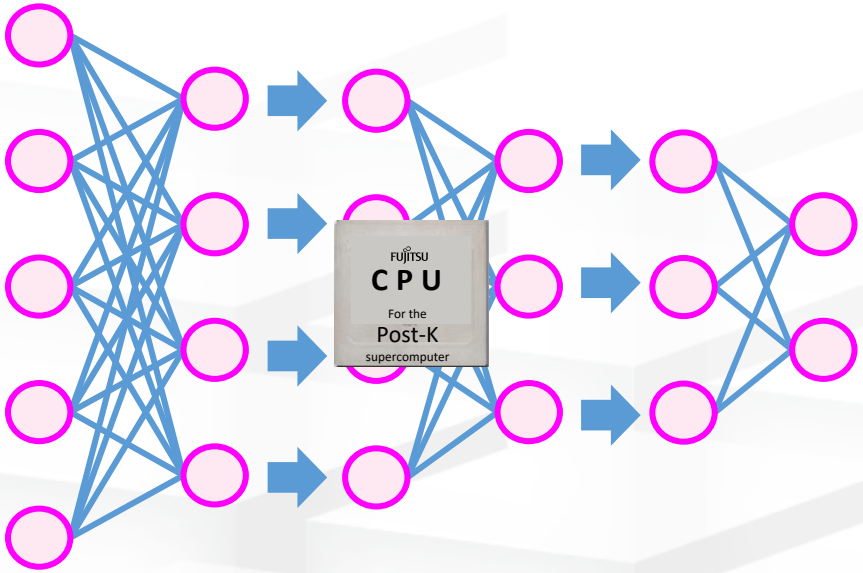
Twelve primary OSS applications are listed and being tested in the Users Group for each compilers, collaboratively w/ Arm

Massive Scale Deep Learning on Post-K

Post-K Processor

- ◆High perf FP16&Int8
- ◆**High mem BW for convolution**
- ◆**Built-in scalable Tofu network**

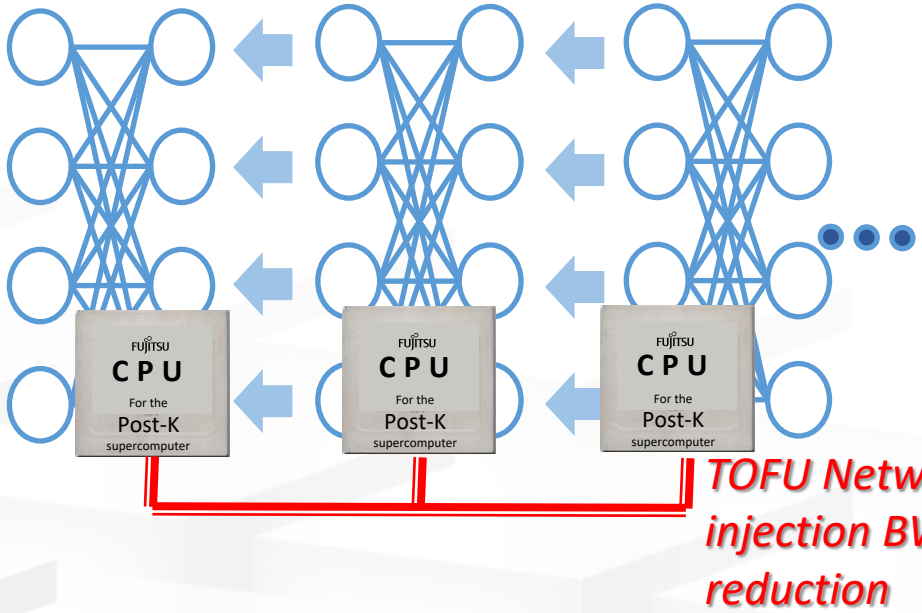
High Performance DNN Convolution



Low Precision ALU + High Memory Bandwidth + Advanced Combining of Convolution Algorithms (FFT+Winograd+GEMM)

Unprecedened DL scalability

High Performance and Ultra-Scalable Network for massive scaling model & data parallelism



Unprecedented Scalability of Data/

“Post-K” Naming

until April 8, 2019 5 pm JST

FUJITSU

- Foreign submissions welcome
- Requirements for the post-K’s name are:
  - The name should preferably express the idea that RIKEN is a world-class research institute operating a state-of-the–art supercomputer.
  - The name should be attractive not only to Japanese speakers but to people around the world.

Call for proposals for the name of the post-K

The RIKEN Center for Computational Science (R-CCS) is calling for proposals for the name of the successor to the K computer (often referred to as the post-K computer), which has been under development with the target to start providing shared use service around the year 2021.

https://www.r-ccs.riken.jp/en/topics/naming.html