

The Supercomputer “Fugaku” and Arm-SVE enabled A64FX processor for energy-efficiency and sustained application performance

Mitsuhisa Sato Team Leader of Architecture Development Team

Deputy project leader, FLAGSHIP 2020 project

Deputy Director, RIKEN Center for Computational Science (R-CCS)

Professor (Cooperative Graduate School Program), University of Tsukuba

FLAGSHIP2020 Project “Fugaku”

□ Missions

- Building the Japanese national flagship supercomputer “Fugaku “(a.k.a post K), and
- Developing wide range of HPC applications, running on Fugaku, in order to solve social and science issues in Japan (**application development projects was over at the end of march, 2020**)

□ Overview of Fugaku architecture

Node: Manycore architecture

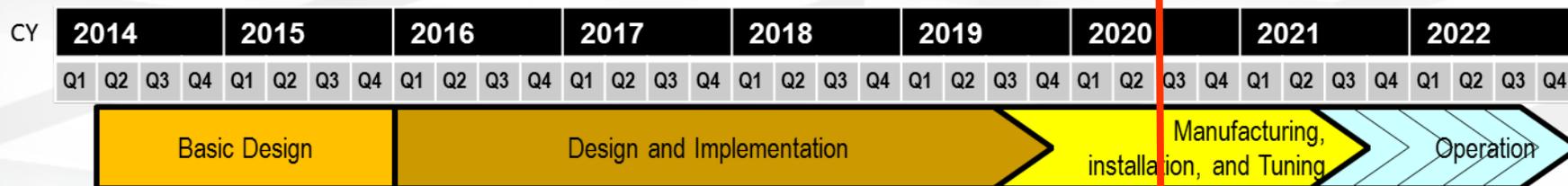
- Armv8-A + SVE (Scalable Vector Extension)
- SIMD Length: 512 bits
- # of Cores: 48 + (2/4 for OS) (> 3.0 TF / 48 core)
- Co-design with application developers and high memory bandwidth utilizing **on-package stacked memory (HBM2) 1 TB/s B/W**
- **Low power : 15GF/W (dgemm)**

Network: TofuD

- Chip-Integrated NIC, 6D mesh/torus Interconnect

□ Status and Update

- March 2019: The Name of the system was decided as “Fugaku”
- Aug. 2019: The K computer decommissioned, stopped the services and shutdown (removed from the computer room)
- Oct 2019: access to the test chips was started.
- **Nov. 2019: Fujitsu announce FX1000 and FX700, and business with Cray.**
- **Nov 2019: Fugaku clock frequency will be 2.0GHz and boost to 2.2 GHz.**
- **Nov 2019: Green 500 1st position!**
- Oct-Nov 2019: MEXT announced the Fugaku “early access program” to begin around Q2/CY2020
- **Dec 2019: Delivery and Installation of “Fugaku” was started.**
- **May 2020: Delivery completed**
- **June 2020: 1st in Top500, HPCG, Graph 500, HPL-AI at ISC2020**



No.1 in Green500 at SC19!

Announce from
Fujitsu at SC19



Green500, Nov. 2019

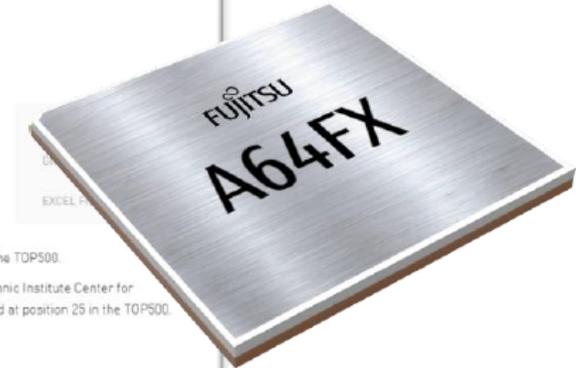
A64FX prototype –
Fujitsu A64FX 48C 2GHz
ranked **#1** on the list

768x general purpose A64FX
CPU w/o accelerators

- 1.9995 PFLOPS @ HPL, 84.75%
- 16.876 GF/W
- Power quality level 2

The Green500 website screenshot shows the November 2019 list. The top entry is the Fujitsu A64FX prototype, ranked #1. The table below shows the top 5 systems.

Rank	Rank	System	Cores	Rmax (TFlop/s)	Power (kW)	Power Efficiency (GFlops/watts)
1	159	A64FX prototype - Fujitsu A64FX, Fujitsu A64FX 48C 2GHz, Tofu interconnect D, Fujitsu Fujitsu Numazu Plant Japan	36,864	1,999.5	118	16.876
2	420	NA-1 - ZettaScaler-2.2, Xeon D-1571 16C 1.3GHz, Infiniband EDR, PEZY-SC2 700Mhz, PEZY Computing / Exascale Inc, PEZY Computing K.K. Japan	1,271,040	1,303.2	80	16.256
3	24	AIMOS - IBM Power System AC922, IBM POWER9 20C 3.43GHz, Dual-rail Mellanox EDR Infiniband, NVIDIA Volta GV100, IBM Rensselaer Polytechnic Institute Center for Computational Innovations (CCI) United States	130,000	8,045.0	510	15.771
4	373	Satori - IBM Power System AC922, IBM POWER9 20C 2.40GHz, Infiniband EDR, NVIDIA Tesla V100 SXM2, IBM MIT/MGHPC Holyoke, MA United States	23,040	1,464.0	94	15.374
5	1	Summit - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148,600.0	10,096	14.719



FUJITSU

Fugaku won 1st position in 4 benchmarks!

Benchmark	1st	Score	Unit	2nd	Score	1 st / 2 nd
TOP500 (LINPACK)	Fugaku	415.5	PFLOPS	Summit (US)	148.6	2.80
HPCG	Fugaku	13.4	PFLOPS	Summit (US)	2.93	4.57
HPL-AI	Fugaku	1.42	EFLOPS	Summit (US)	0.55	2.58
Graph500	Fugaku	70,980	GTEPS	太湖之光 TaihuLight (China)	23,756	2.99

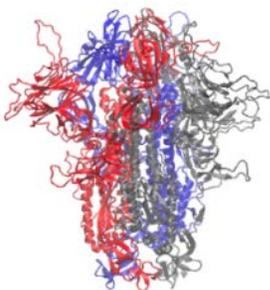
2 to 4 times faster in every benchmark!

MEXT Fugaku Program: Fight Against COVID19

Fugaku resources made available a year ahead of general production
(more research topics under international solicitation)

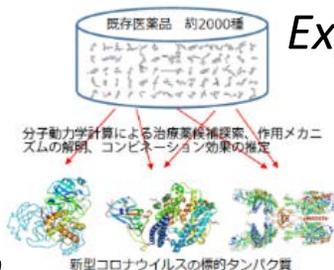
Medical-Pharma

Prediction of conformational dynamics of proteins on the surface of SARS-Cov-2



GENESIS MD to interpolate unknown experimentally undetectable dynamic behavior of spike proteins, whose static behavior has been identified via Cryo-EM

(Yuji Sugita, RIKEN)



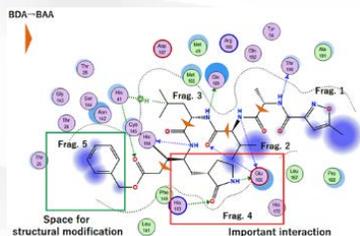
Exploring new drug candidates for COVID-19

Large-scale MD to search & identify therapeutic drug candidates showing high affinity for COVID-19 target proteins from 2000 existing drugs

(Yasushi Okuno, RIKEN / Kyoto University)



Fragment molecular orbital calculations for COVID-19 proteins



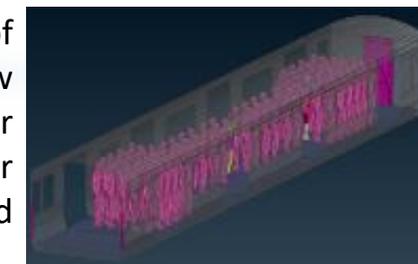
Large-scale, detailed interaction analysis of COVID-19 using Fragment Molecular Orbital (FMO) calculations using ABINIT-MP

(Yuji Mochizuki, Rikkyo University)

Societal-Epidemiology

Prediction and Countermeasure for Virus Droplet Infection under the Indoor Environment

Massive parallel simulation of droplet scattering with airflow and heat transfer under indoor environment such as commuter trains, offices, classrooms, and hospital rooms



(Makoto Tsubokura, RIKEN / Kobe University)

Simulation analysis of pandemic phenomena

Combining simulations & analytics of disease propagation w/contact tracing apps, economic effects of lockdown, and reflections social media, for effective mitigation policies



(Nobuyasu Ito, RIKEN)

Target applications for co-design

- Typical benchmarks such as HPL (dgemm) and stream
- Target applications provided by each application projects (“9 priority issues”)

Target applications are 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)

KPIs on Fugaku development in FLAGSHIP 2020 project

3 KPIs (key performance indicator) were defined for Fugaku development

● 1. Extreme Power-Efficient System

- Maximum performance under Power consumption of 30 - 40MW (for system)
- Approx. 15 GF/W (dgemm) confirmed by the prototype CPU => 1st in Green 500 !!!

● 2. Effective performance of target applications

- It is expected to exceed 100 times higher than the K computer's performance in some applications
- 125 times faster in GENESIS (MD application), 120 times faster in NICAM+LETKF (climate simulation and data assimilation) were estimated

● 3. Ease-of-use system for wide-range of users

- Co-design with application developers
- Shared memory system with high-bandwidth on-package memory must make existing OpenMP-MPI program ported easily.
- No programming effort for accelerators such as GPUs is required.

Target Application's Performance

● Performance Targets

- 100 times faster than K for some applications (tuning included)
- 30 to 40 MW power consumption

<https://postk-web.r-ccs.riken.jp/perf.html>

□ Predicted Performance of 9 Target Applications

As of 2019/05/14

Area	Priority Issue	Performance Speedup over K	Application	Brief description
Health and longevity	1. Innovative computing infrastructure for drug discovery	x125+	GENESIS	MD for proteins
	2. Personalized and preventive medicine using big data	x8+	Genomon	Genome processing (Genome alignment)
Disaster prevention and Environment	3. Integrated simulation systems induced by earthquake and tsunami	x45+	GAMERA	Earthquake simulator (FEM in unstructured & structured grid)
	4. Meteorological and global environmental prediction using big data	x120+	NICAM+ LETKF	Weather prediction system using Big data (structured grid stencil & ensemble Kalman filter)
Energy issue	5. New technologies for energy creation, conversion / storage, and use	x40+	NTChem	Molecular electronic (structure calculation)
	6. Accelerated development of innovative clean energy systems	x35+	Adventure	Computational Mechanics System for Large Scale Analysis and Design (unstructured grid)
Industrial competitiveness enhancement	7. Creation of new functional devices and high-performance materials	x30+	RSDFT	Ab-initio program (density functional theory)
	8. Development of innovative design and production processes	x25+	FFB	Large Eddy Simulation (unstructured grid)
Basic science	9. Elucidation of the fundamental laws and evolution of the universe	x25+	LQCD	Lattice QCD simulation (structured grid Monte Carlo)

CPU Architecture: A64FX

- **Armv8.2-A (AArch64 only) + SVE (Scalable Vector Extension)**

- FP64/FP32/FP16 (<https://developer.arm.com/products/architecture/a-profile/docs>)

- **SVE 512-bit wide SIMD**

- **# of Cores: 48 + (2/4 for OS)**

- Co-design with application developers and high memory bandwidth utilizing **on-package stacked memory: HBM2(32GiB)**
- Leading-edge Si-technology (7nm FinFET), **low power logic design (approx. 15 GF/W (dgemm))**, and **power-controlling knobs**

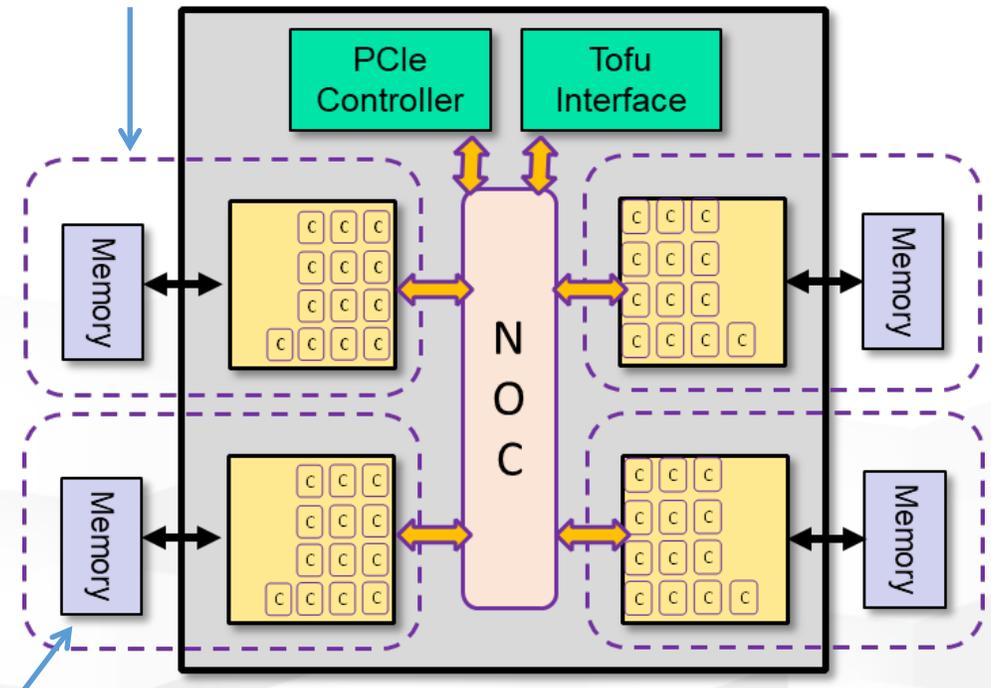
- PCIe Gen3 16 lanes

- Peak performance

- 3.0 TFLOPS@2GHz (>90% @ dgemm)
- Memory B/W 1024GB/s (>80% stream)
- Byte per Flops: approx. 0.4

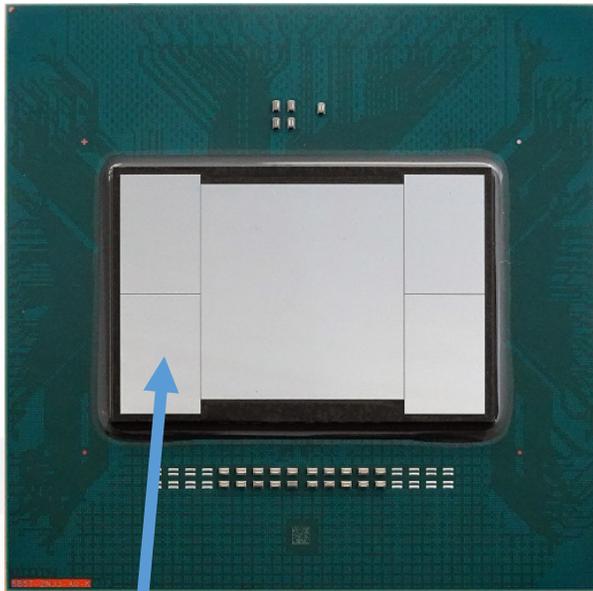
- ◆ “Common” programming model will be to run each MPI process on a NUMA node (CMG) with OpenMP-MPI hybrid programming.
- ◆ 48 threads OpenMP is also supported.

CMG(Core-Memory-Group): NUMA node
12+1 core

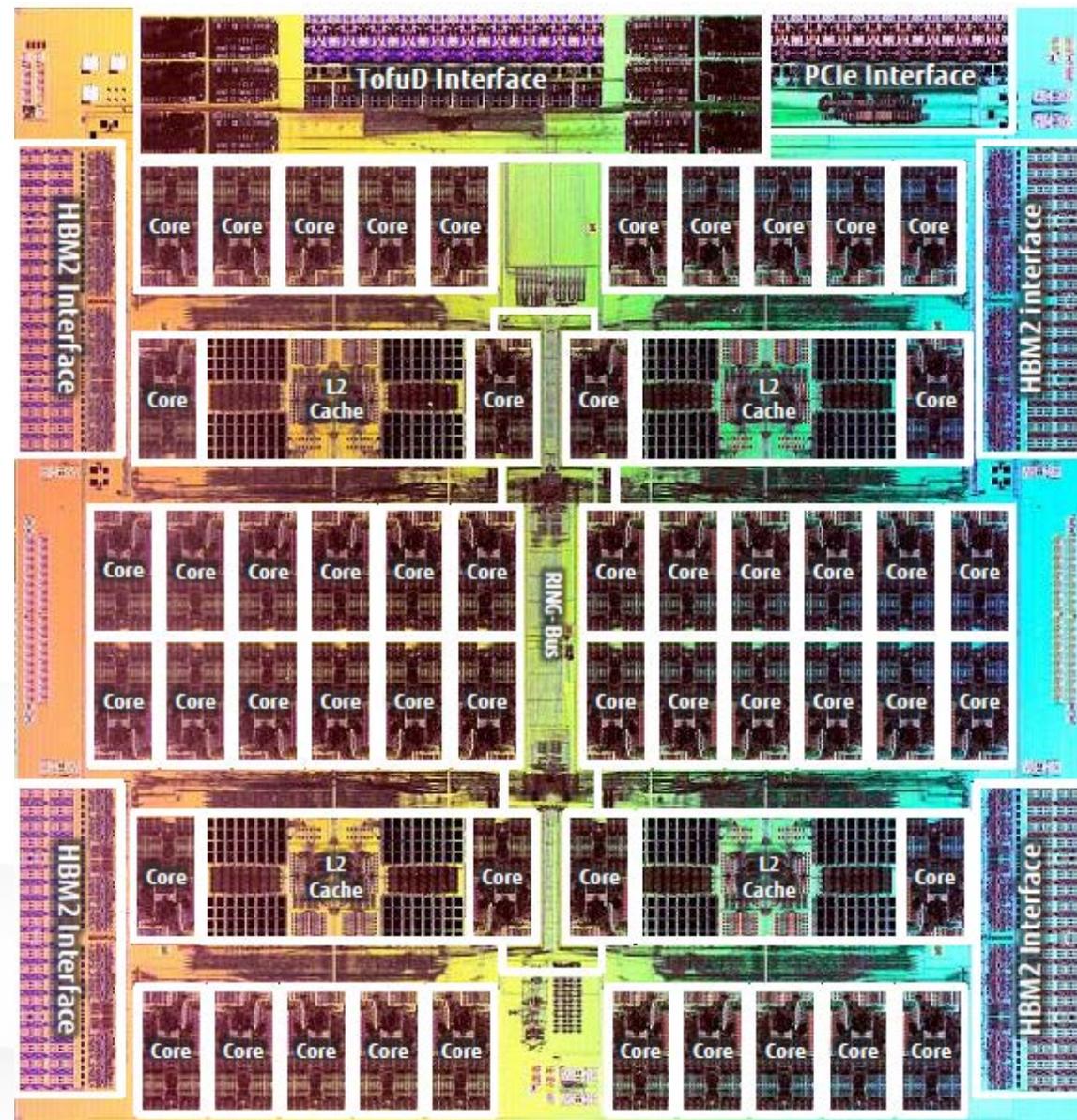


HBM2: 8GiB

- TSMC 7nm FinFET
- CoWoS technologies for HBM2



HBM2



ARM v8 Scalable Vector Extension (SVE)

- **SVE is a complementary extension that does not replace NEON, and was developed specifically for vectorization of HPC scientific workloads.**
- **The new features and the benefits of SVE comparing to NEON**
 - **Scalable vector length (VL)** : Increased parallelism while allowing implementation choice of VL
 - **VL agnostic (VLA) programming**: Supports a programming paradigm of write-once, run-anywhere scalable vector code
 - **Gather-load & Scatter-store**: Enables vectorization of complex data structures with non-linear access patterns
 - **Per-lane predication**: Enables vectorization of complex, nested control code containing side effects and avoidance of loop heads and tails (particularly for VLA)
 - **Predicate-driven loop control and management**: Reduces vectorization overhead relative to scalar code
 - **Vector partitioning and SW managed speculation**: Permits vectorization of uncounted loops with data-dependent exits
 - **Extended integer and floating-point horizontal reductions**: Allows vectorization of more types of reducible loop-carried dependencies
 - **Scalarized intra-vector sub-loops**: Supports vectorization of loops containing complex loop-carried dependencies

SVE example

DAXPY (scalar)

```
// -----  
//      subroutine daxpy(x,y,a,n)  
//      real*8 x(n),y(n),a  
//      do i = 1,n  
//          y(i) = a*x(i) + y(i)  
//      enddo  
// -----  
// x0 = &x[0], x1 = &y[0], x2 = &a, x3 = &n  
daxpy_  
    ldrsw    x3, [x3]           // x3=*n  
    mov     x4, #0             // x4=i=0  
    ldr     d0, [x2]           // d0=*a  
    b      .latch  
.loop:  
    ldr     d1, [x0,x4,1s1 3]   // d1=x[i]  
    ldr     d2, [x1,x4,1s1 3]   // d2=y[i]  
    fmadd  d2, d1, d0, d2       // d2+=x[i]*a  
    str     d2, [x1,x4,1s1 3]   // y[i]=d2  
    add    x4, x4, #1           // i+=1  
.latch:  
    cmp    x4, x3               // i < n  
    b.lt  .loop                 // more to do?  
    ret
```

DAXPY (SVE)

```
// -----  
//      subroutine daxpy(x,y,a,n)  
//      real*8 x(n),y(n),a  
//      do i = 1,n  
//          y(i) = a*x(i) + y(i)  
//      enddo  
// -----  
// x0 = &x[0], x1 = &y[0], x2 = &a, x3 = &n  
daxpy_  
    ldrsw    x3, [x3]           // x3=*n  
    mov     x4, #0             // x4=i=0  
    whilelt p0.d, x4, x3       // p0=while(i++<n)  
    ldldr    z0.d, p0/z, [x2]   // p0:z0=bcast(*a)  
.loop:  
    ldld    z1.d, p0/z, [x0,x4,1s1 3] // p0:z1=x[i]  
    ldld    z2.d, p0/z, [x1,x4,1s1 3] // p0:z2=y[i]  
    fmla    z2.d, p0/m, z1.d, z0.d // p0?z2+=x[i]*a  
    stld    z2.d, p0, [x1,x4,1s1 3] // p0?y[i]=z2  
    incd    x4                  // i+=(VL/64)  
.latch:  
    whilelt p0.d, x4, x3       // p0=while(i++<n)  
    b.first .loop              // more to do?  
    ret
```

Make predicate mask

SIMD with mask

- Compact code for SVE as scalar loop
- OpenMP SIMD directive is expected to help the SVE programming

- **Leading-edge Si-technology (7nm FinFET)**
- **Low power logic design (15 GF/W @ dgemm)**
- **A64FX provides power management function called “Power Knob”**
 - FL pipeline usage: FLA only, EX pipeline usage : EXA only, Frequency reduction ...
 - User program can change “Power Knob” for power optimization
 - “Energy monitor” facility enables chip-level power monitoring and detailed power analysis of applications

- **“Eco-mode” : FLA only with lower “stand-by” power for ALUs**
 - Reduce the power-consumption for memory intensive apps.
 - 4 apps out of 9 target applications select “eco-mode” for the max performance under the limitation of our power capacity (Even using HBM2!)

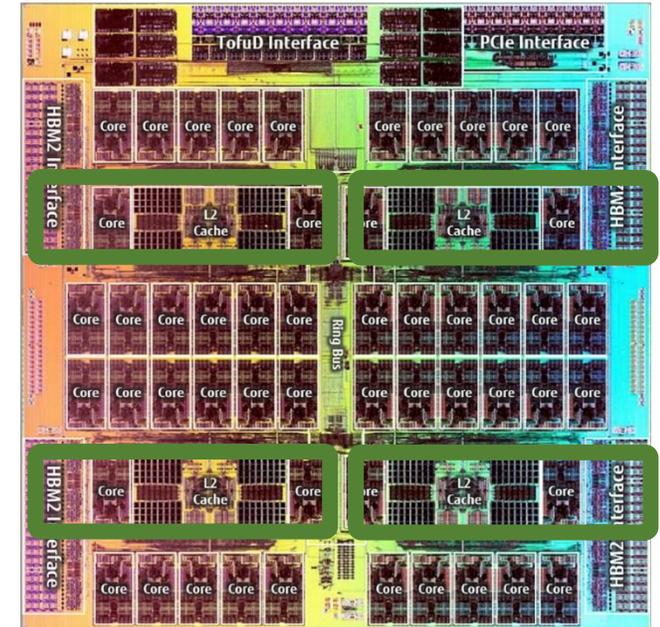
- **Retention mode: power state for de-activation of CPU with keeping network alive**
 - Large reduction of system power-consumption at idle time

CPU A64FX



Architecture	Armv8.2-A SVE (512 bit SIMD)	
Core	48 cores for compute and 2/4 for OS activities	
	Normal: 2.0 GHz	DP: 3.072 TF, SP: 6.144 TF, HP: 12.288 TF
	Boost: 2.2 GHz	DP: 3.3792TF, SP: 6.7584 TF, HP: 13.5168 TF
Cache L1	64 KiB, 4 way, 230+ GB/s(load), 115+ GB/s (store)	
Cache L2	CMG(NUMA): 8 MiB, 16way	
	Node: 3.6+ TB/s Core: 115+ GB/s (load), 57+ GB/s (store)	
Memory	HBM2 32 GiB, 1024 GB/s	
Interconnect	TofuD (28 Gbps x 2 lane x 10 port)	
I/O	PCIe Gen3 x 16 lane	
Technology	7nm FinFET	

4 NUMA Nodes



Performance

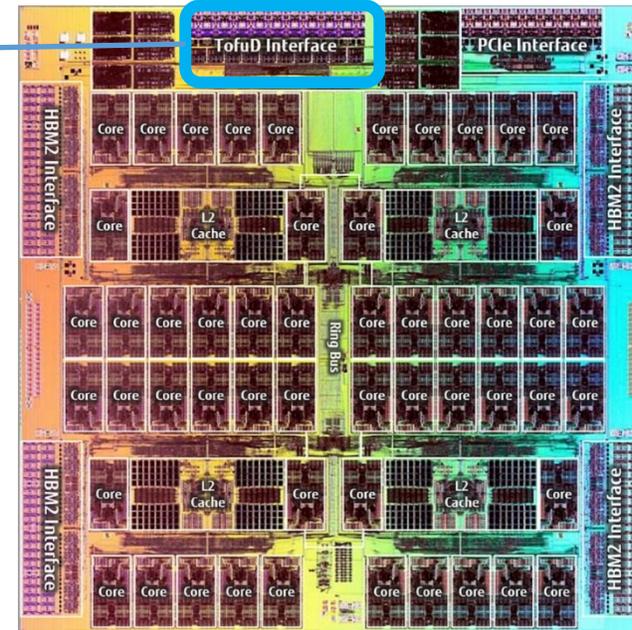
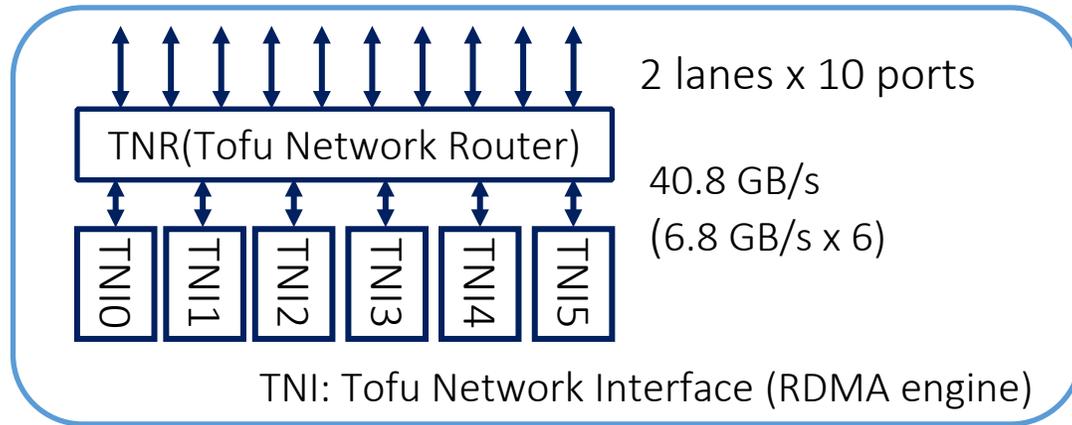
Stream triad: 830+ GB/s

Dgemm: 2.5+ TF (90+% efficiency)

ref. Toshio Yoshida, "Fujitsu High Performance CPU for the Post-K Computer,"
IEEE Hot Chips: A Symposium on High Performance Chips, San Jose, August 21, 2018.

Courtesy of FUJITSU LIMITED

TofuD Interconnect

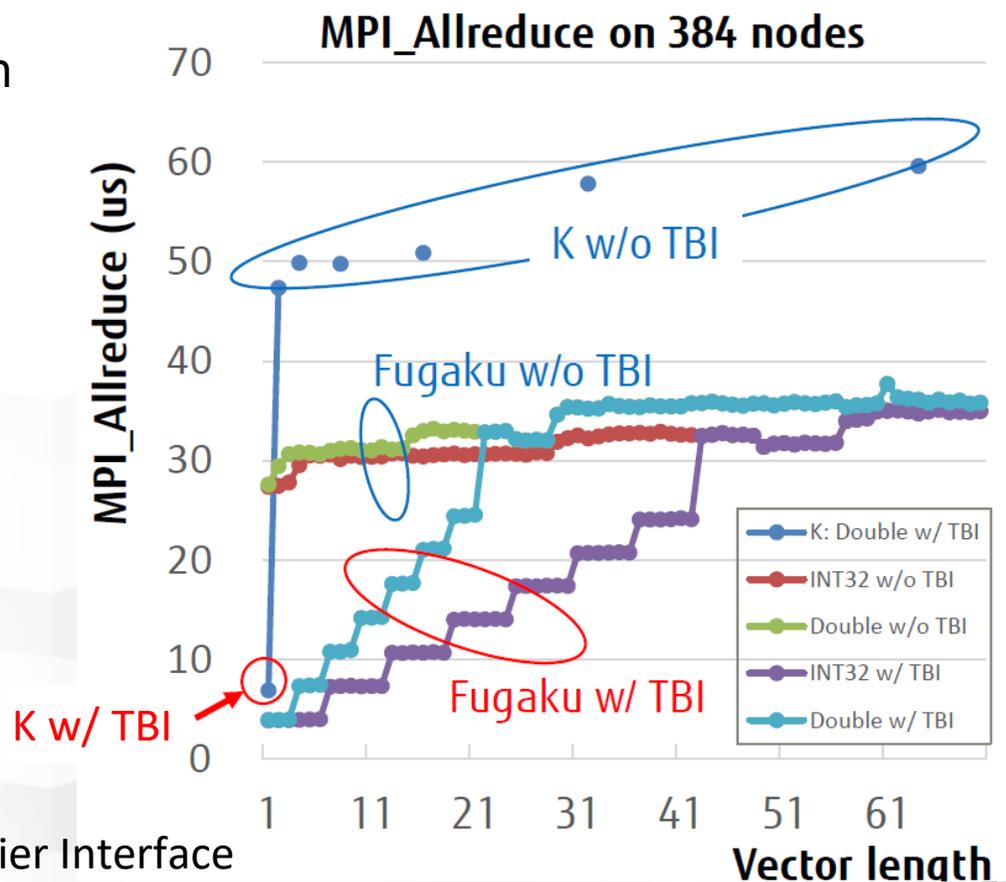


- 6 RDMA Engines
- Hardware barrier support
- Network operation offloading capability

8B Put latency	0.49 – 0.54 usec
1MiB Put throughput	6.35 GB/s

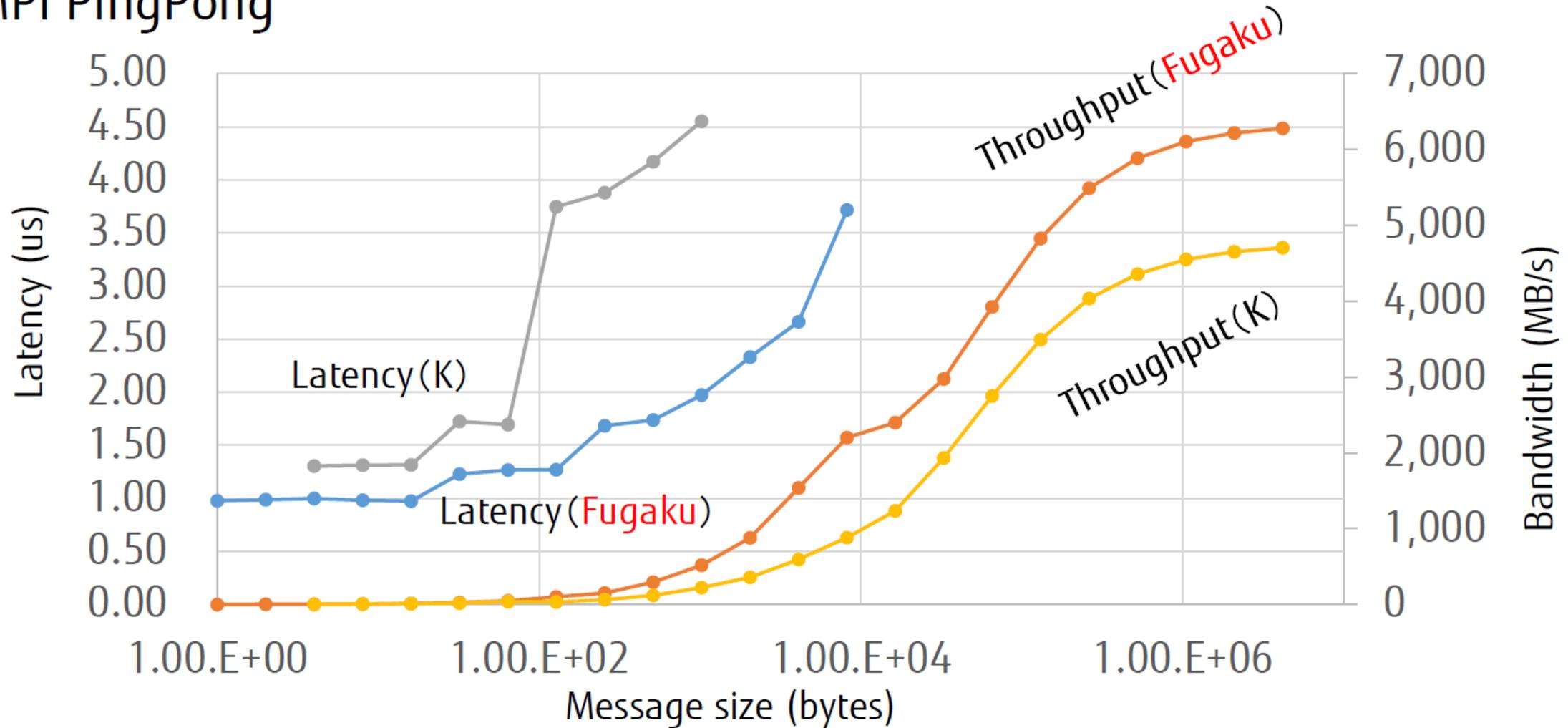
Co-design of network

- We have selected “Tofu” network for performance compatibility in large-scale applications.
- Select Link-speed 28Gps x 4 due to technology availability around 2019
- Communication patterns were extracted, and the communication performance was estimated by “analytical model”.
 - Many target applications have neighbor communication pattern, or communication to near nodes. ⇒ “Tofu” and 28Gbps link were sufficient.
 - Some apps have all-to-all communication.
 - We studied the benefits or feasibility of additional “dedicated” all-to-all network, but it was not selected due to cost
 - Support 3 DP reduction by Tofu TBI for QCD apps
- “Common” programming model will be to run each MPI process on a NUMA node (CMG) with OpenMP-MPI hybrid programming.
 - 48 threads OpenMP is also supported.

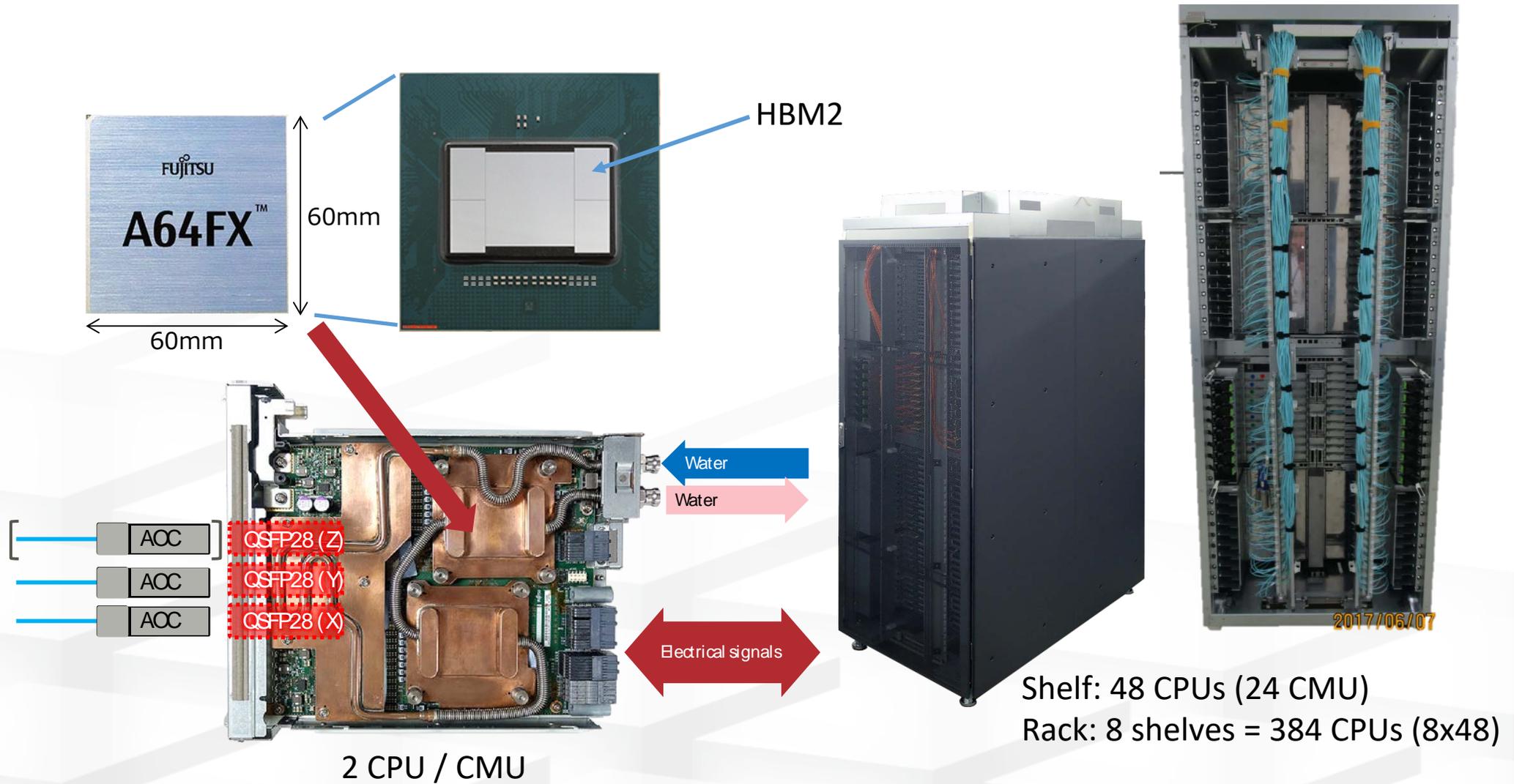


TofuD: MPI_Send/Receive Latency and BW

■ MPI PingPong



Fugaku prototype board and rack



Fugaku System Configuration

- **150k+ node**

Boost mode: 3.3792TF x 150k+ = 500+ PF

- **Two types of nodes**

- Compute Node and Compute & I/O Node connected by Fujitsu TofuD, 6D mesh/torus Interconnect

- **3-level hierarchical storage system**

- 1st Layer

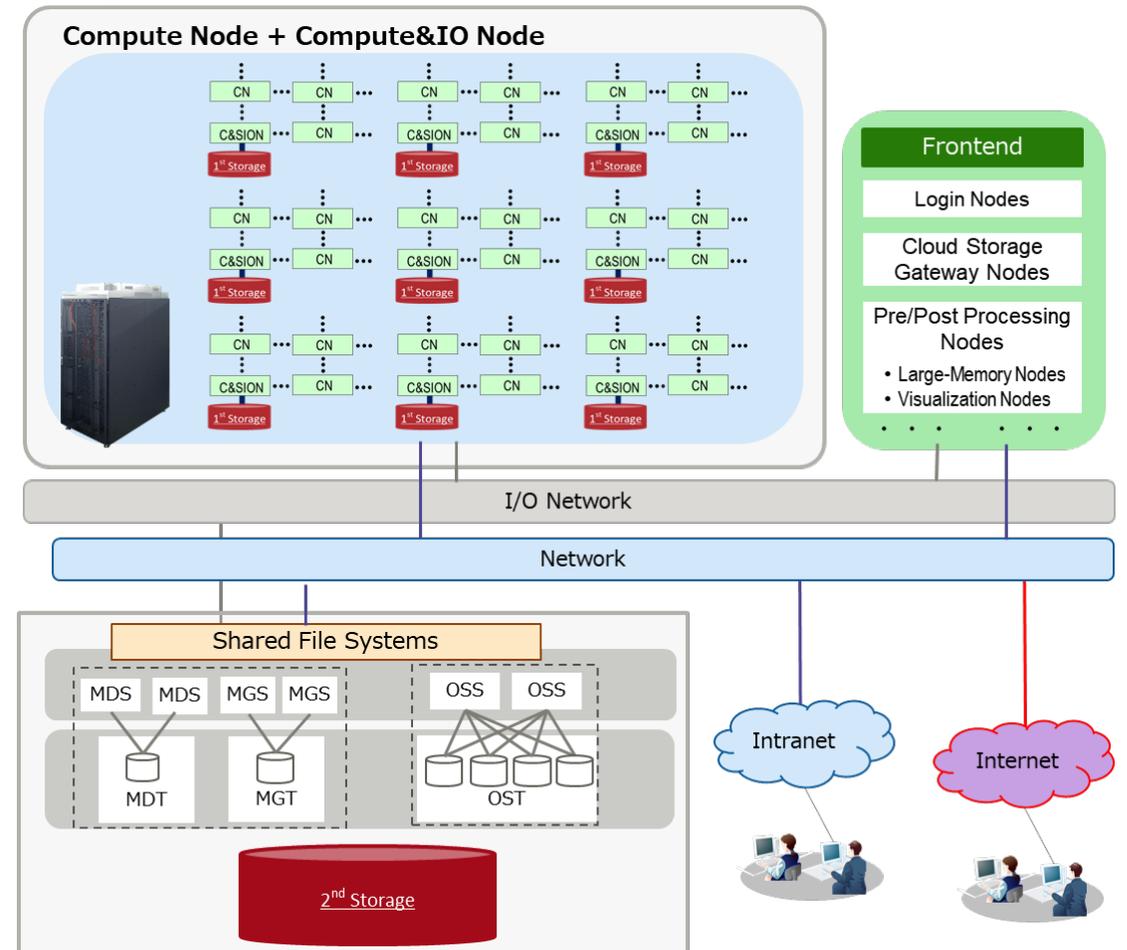
- One of 16 compute nodes, called Compute & Storage I/O Node, has SSD about 1.6 TB
- Services
 - Cache for global file system
 - Temporary file systems
 - Local file system for compute node
 - Shared file system for a job

- 2nd Layer

- Fujitsu FEFS: Lustre-based global file system

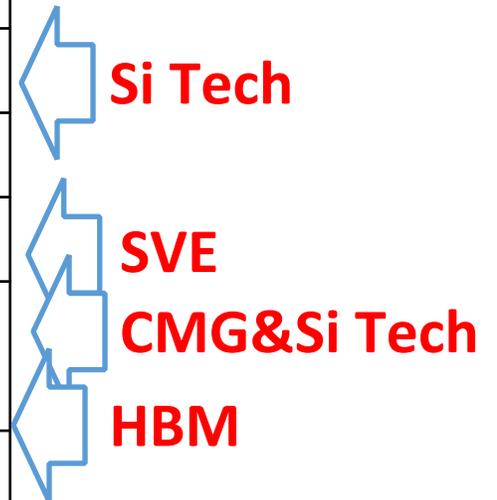
- 3rd Layer

- Cloud storage services



Advances from the K computer

	K computer	Fugaku	ratio
# core	8	48	
Si tech. (nm)	45	7	
Core perf. (GFLOPS)	16	64(70)	4(4.4)
Chip(node) perf. (TFLOPS)	0.128	3.072 (3.379)	24 (26.4)
Memory BW (GB/s)	64	1024	
B/F (Bytes/FLOP)	0.5	0.4	
#node / rack	96	384	4
#node/system	82,944	158,976	
System perf.(DP PFLOPS)	10.6	488 (537) 977(1070)	42.3(52.2) 84.6(104.4)



More than **7.6 M**
General-purpose
cores!

- SVE increases core performance
- Silicon tech. and scalable architecture (CMG) to increase node performance
- HBM enables high bandwidth

Value in brackets
Indicate the number
At boost mode (2.2GHz)

- **CloverLeaf (UK Mini-App Consortium), Fortran/C**
 - A hydrodynamics mini-app to solve the compressible Euler equations in 2D, using an explicit, second-order method
 - Stencil calculation
- **TeaLeaf (UK Mini-App Consortium), Fortran**
 - A mini-application to enable design-space explorations for iterative sparse linear solvers
 - https://github.com/UK-MAC/TeaLeaf_ref.git
 - Problem size: Benchmarks/tea_bm_5.in, end_step=10 -> 3
- **LULESH (LLNL), C**
 - Mini-app representative of simplified 3D Lagrangian hydrodynamics on an unstructured mesh, indirect memory access

Disclaimer:

The software used for the evaluation, such as the compiler, is still under development and its performance may be different when **the supercomputer Fugaku** starts its operation.

● Platform

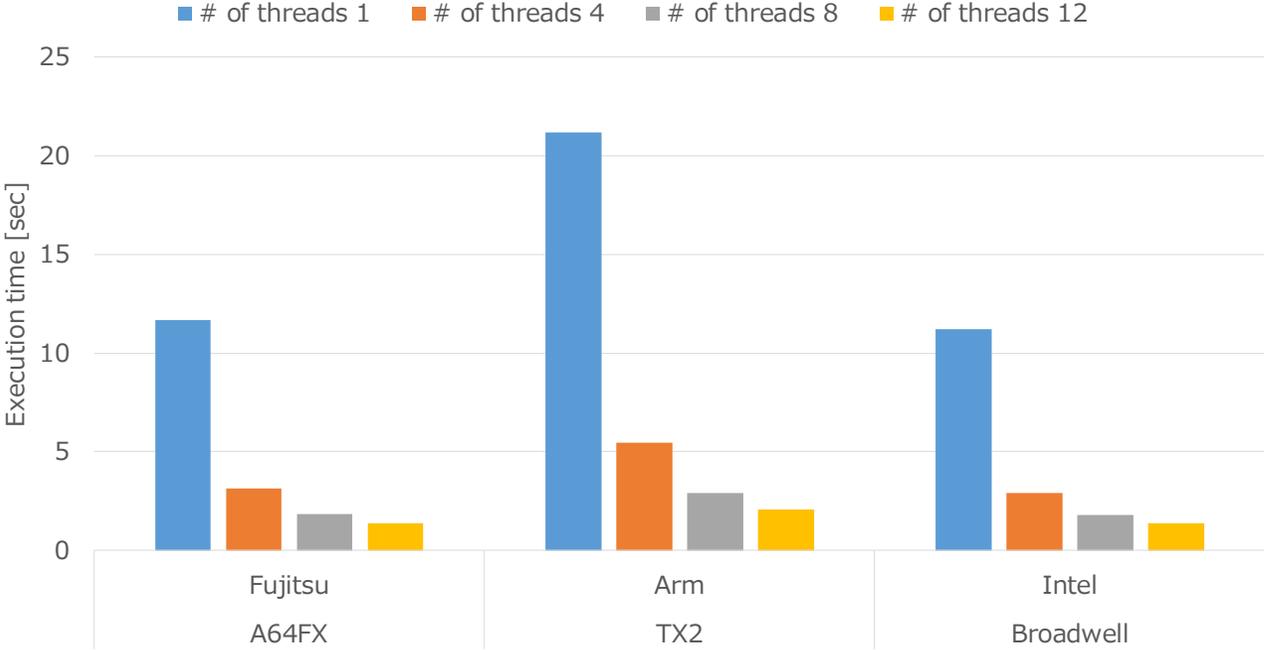
- A64FX test chip (2.0 GHz)
 - ThunderX2 @ Apollo70
 - 28C/2S @ 2.0GHz
 - Arm HPC compiler 19.1
 - Broadwell (Xeon E5-2680 v4)
 - 14C/2S @ 2.4GHz
 - Intel compiler 2019.0.045
 - Skylake (Xeon Gold 6126) @ Cygnus, Univ. of Tsukuba
 - 12C/2S @ 2.6GHz
 - Intel compiler 19.0.3.199

● Compiler Options

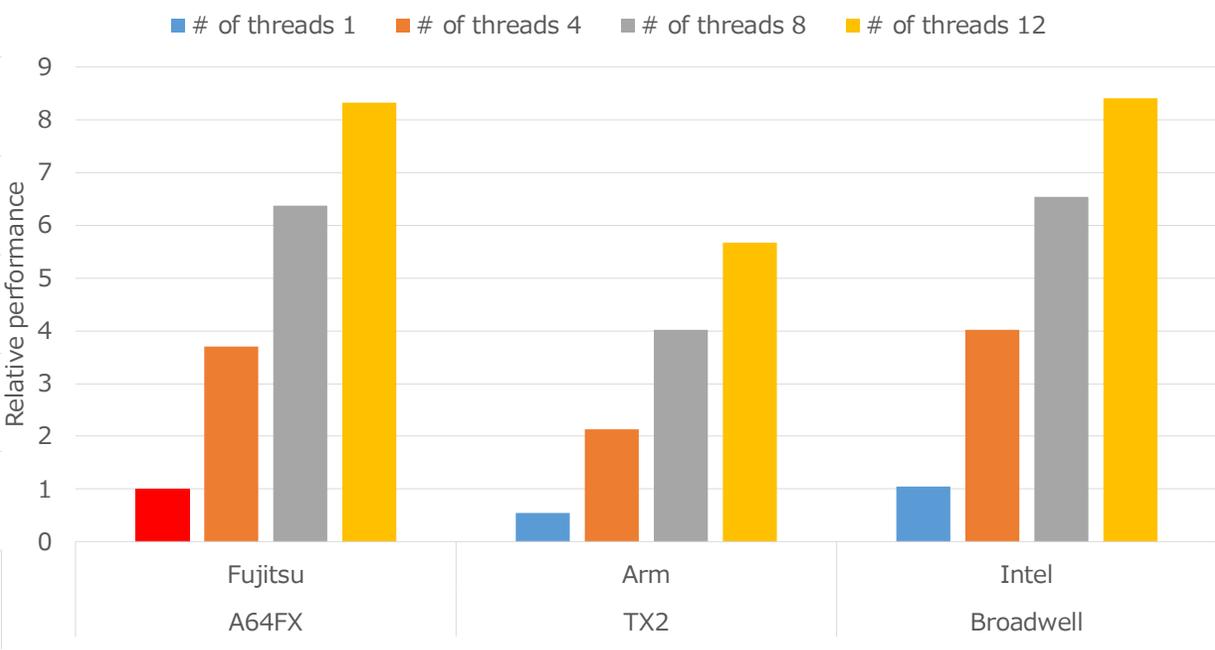
- Fujitsu compiler
 - -Kfast,openmp
- Arm HPC compiler
 - -Ofast -march=armv8-a(+sve)
- Intel compiler
 - -O3 -qopenmp -march=native

CloverLeaf

Execution time



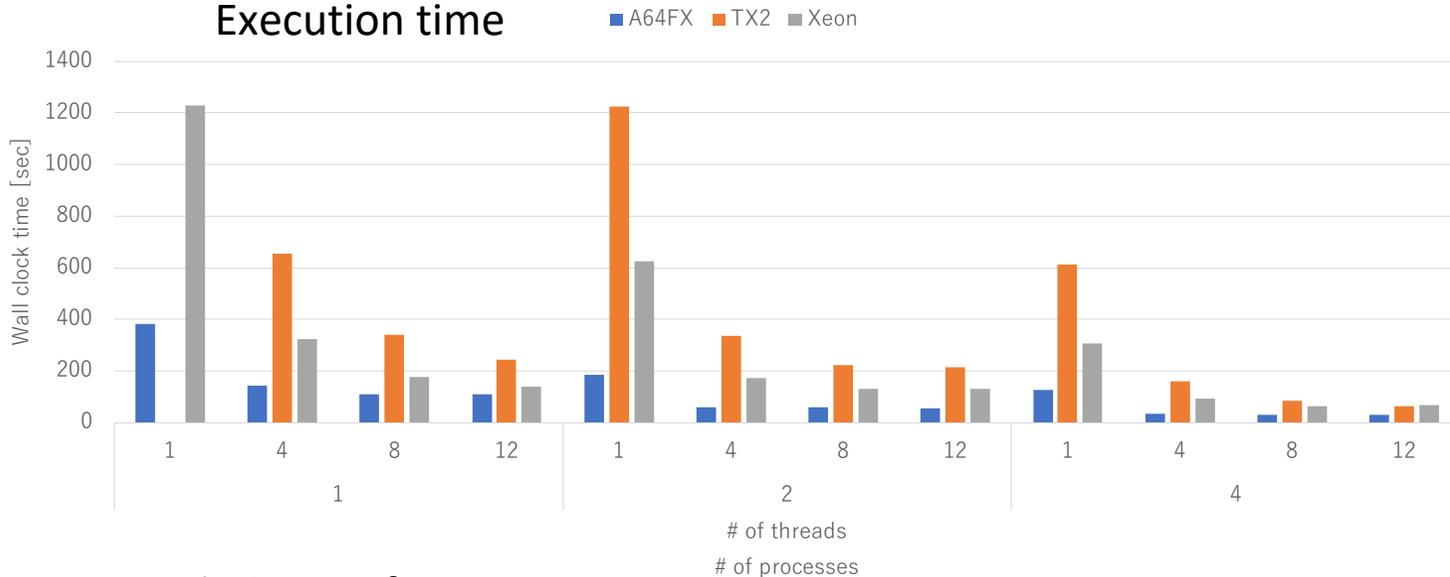
Relative performance (to 1T/A64FX)



- Evaluation using one CMG(NUMA node) without MPI
- Good scalability by increasing the number of threads within CMG.
- One GMG performance is comparable to Intel one. (Chip contains 4 CMG!)

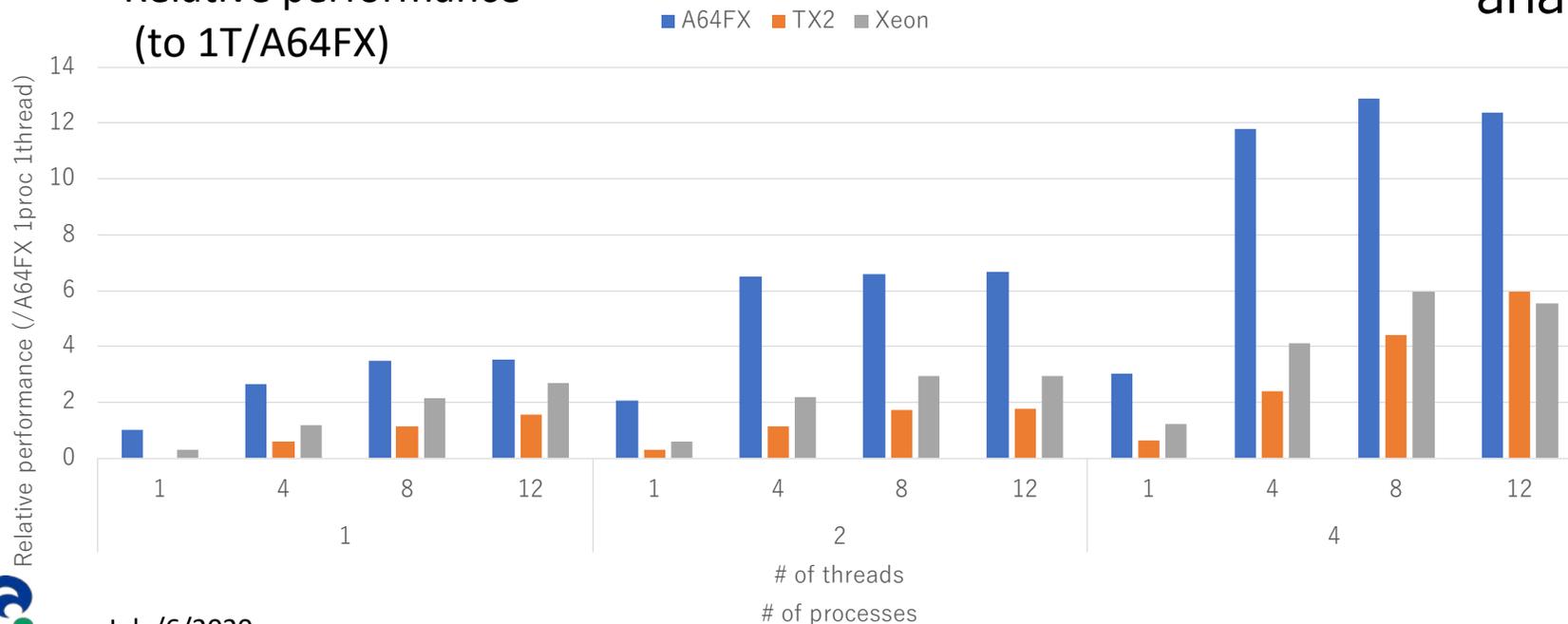
TeaLeaf

Execution time



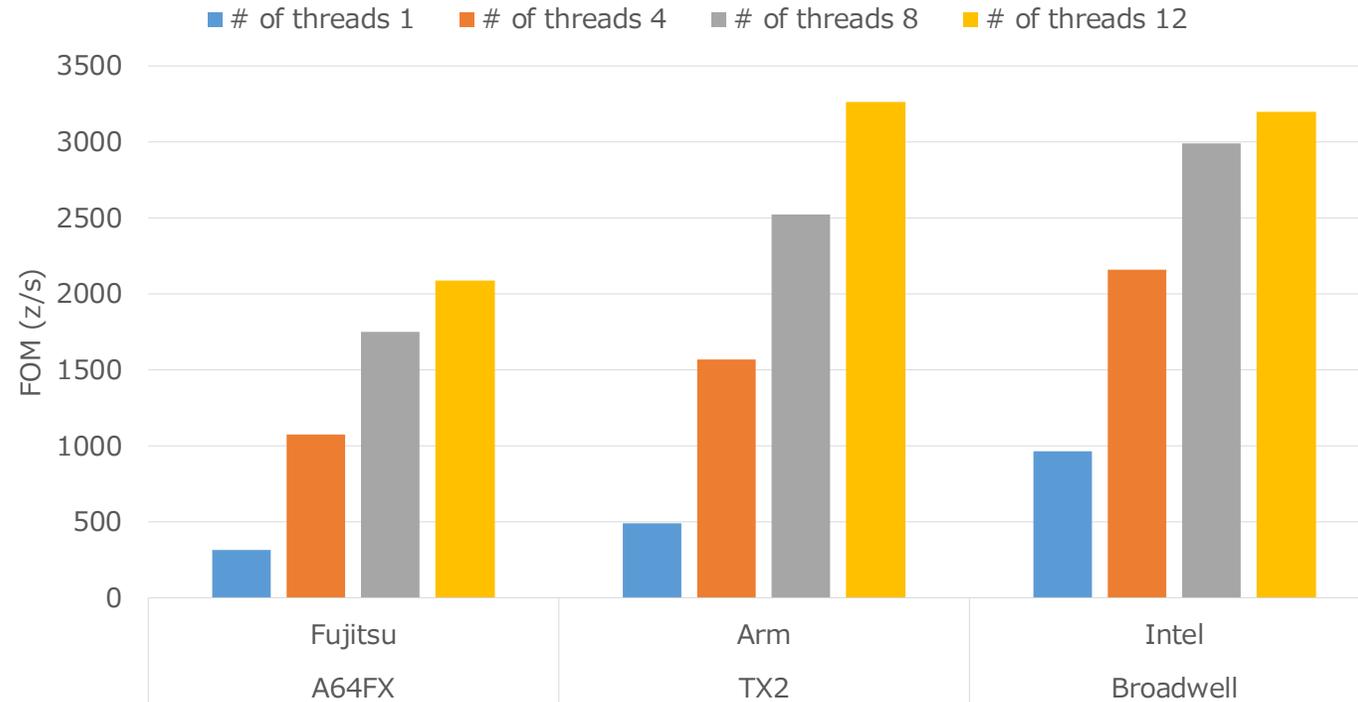
- Evaluation of MPI program within one chip (upto 4 MPI process)
- Changing #threads within CMG
- The speedup is limited for more than 4 threads due to the memory bandwidth (?)
- We need more performance analysis.

Relative performance (to 1T/A64FX)



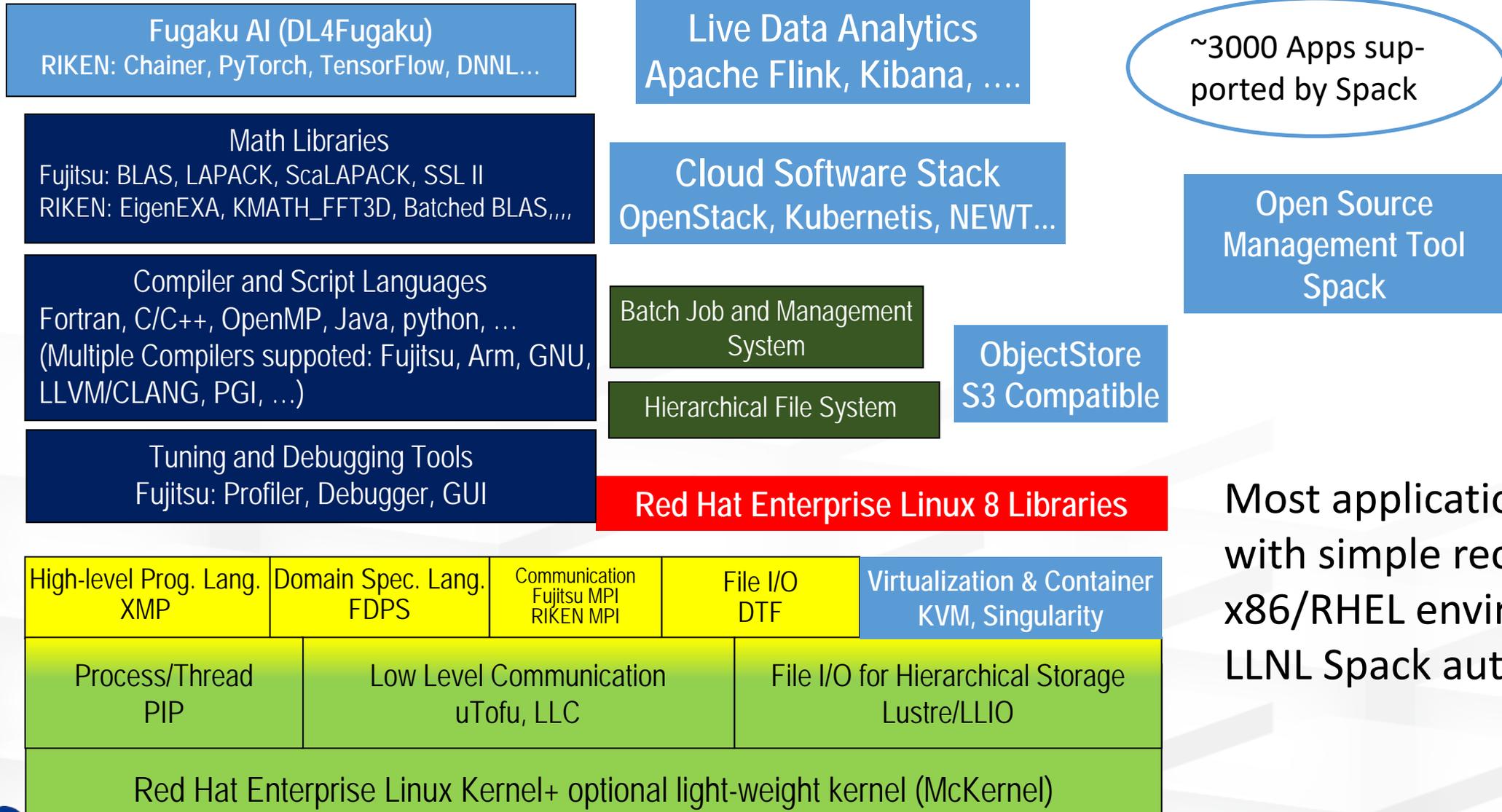
Xeon @ Cygnus, Univ. of Tsukuba
 Intel Xeon Gold 6126
 2.6GHz; 12 core x 2 socket

LULESH



- Evaluation using one CMG(NUMA node) without MPI
- One CMG performance is less than Thx2 and Intel one
- We found low vectorization (SIMD (SVE) instructions ratio is a few %)
- We need more code tuning for more vectorization using SIMD

Fugaku / Fujitsu FX1000 System Software Stack



Most applications will work with simple recompile from x86/RHEL environment. LLNL Spack automates this.

OSS Application Porting @ Arm HPC Users Group

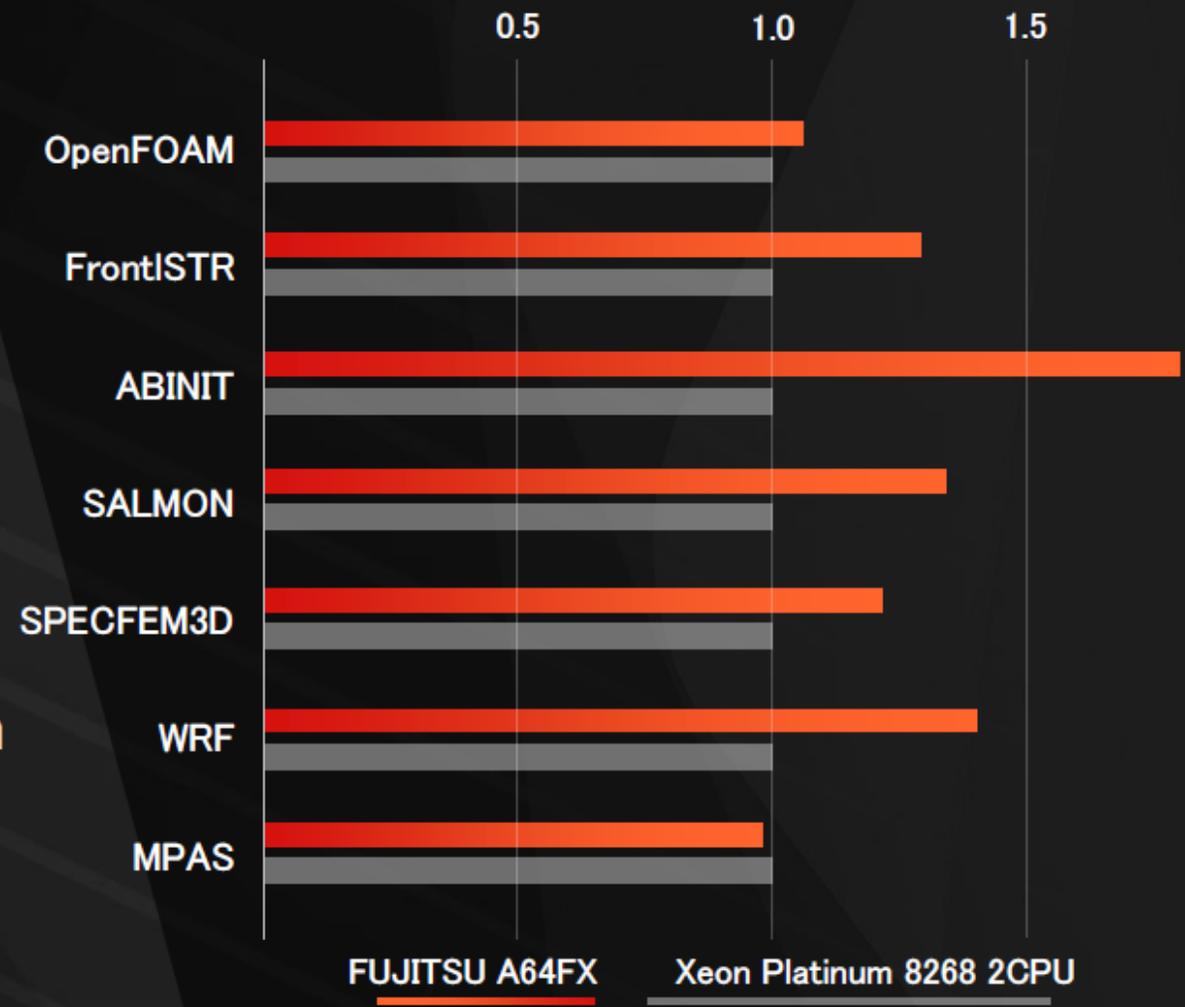
(<http://arm-hpc.gitlab.io/>)

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

High Performance on Real Applications

The performance on 1 node is evaluated for seven OSS applications

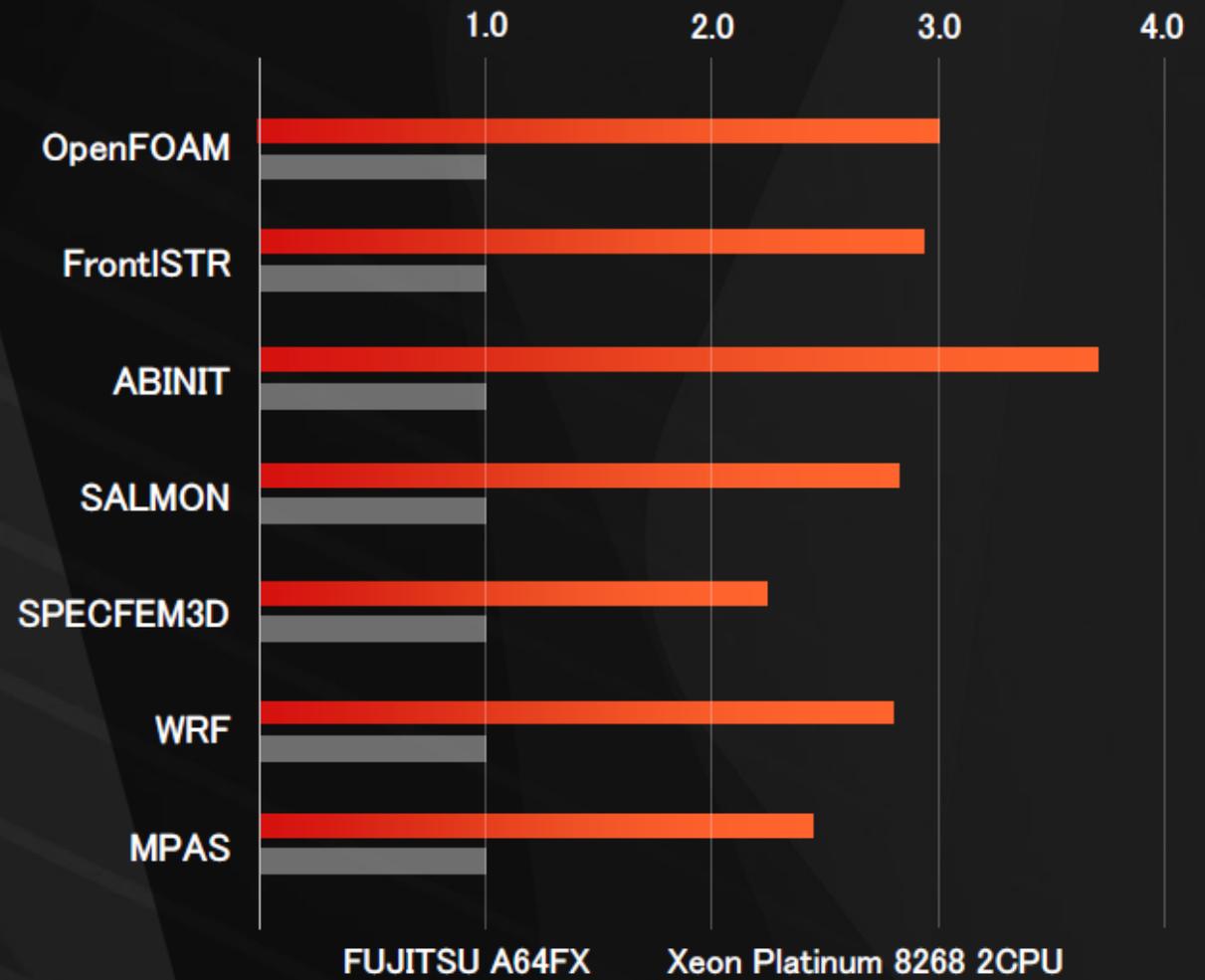
- Measured on PRIMEHPC FX1000, A64FX 2.2GHz
- Up to 1.8x faster over Xeon Platinum 8268 x2
- High memory B/W and long SIMD length Applications work effectively



High Performance in Power Efficiency

The power efficiency on 1 node is evaluated for seven OSS applications

- Measured on PRIMEHPC FX1000, A64FX 2.2GHz
- High efficiency is achieved by energy-conscious design and implementation



Concluding remarks

- **We are now sure to achieve 3 KPIs**
 - Power-efficiency
 - Effective Performance of applications.
 - Ease-of-use
- **Well-balanced system for several apps**
- **2020 early users, incl. COVID-19 apps running already**
- **Open to international users through HPCI, general allocation April 2021 (application starting Sept. 2020)**
- **But, ... “Dark-side” of (our) co-design of HPC**
 - No so “disruptive” architecture (but, ... ease-of-use)
 - Will need application-specific accelerators for more power-efficiency in near future?
 - Or is there any room to improve on the existing processor architecture?