Towards FugakuNEXT: Experiences of Fugaku and path moving forward --- debunking the ‘myths’ of exascale

Satoshi Matsuoka, Director Riken R-CCS
Modsim Workshop, Seattle, WA, USA
2022/08/10-12

Fugaku: Largest Supercomputer Ever, 160K nodes, 8 mil cores

'Applications First' R&D Challenge--- High Risk "Moonshot" R&D

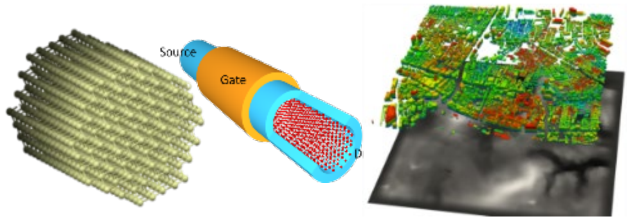
- A new high performance & low power Arm A64FX CPU co-developed by Riken R-CCS & Fujitsu along with nationwide HPC researchers as a National Flagship 2020 project

- 3x perf c.f. top CPU in HPC apps
- 3x power efficiency c.f. top CPU
- General purpose Arm CPU, runs sa me program as Smartphones
- Acceleration features for AI

"Moonshot" R&D Target

- Fugaku x 2~3 = Entire annual IT in Japan

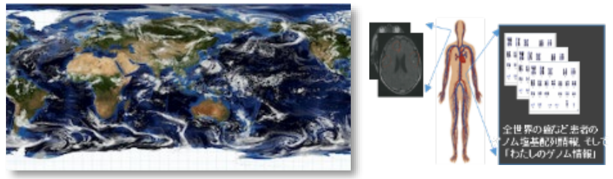
	Smartphones		Servers (incl. IDC)		Fugaku		K Computer
Untis	20 million ~annual shipment in Japan	=	300,000 (~annual shipment in Japan	=	1 (160K nodes)		Max 120
Power (W)	10W×2,000万台＝ 200MW	=	600-700W×30万台＝ 200MW (incl cooling)	>>	18MW (very low)		14MW (less than 1/10 efficiency c.f. Fugaku)



- Developed via extensive co-design

"Science of Computing"

By Riken & Fujitsu & HPCI Centers, etc., Arm Ecosystem, Reflecting numerous research results

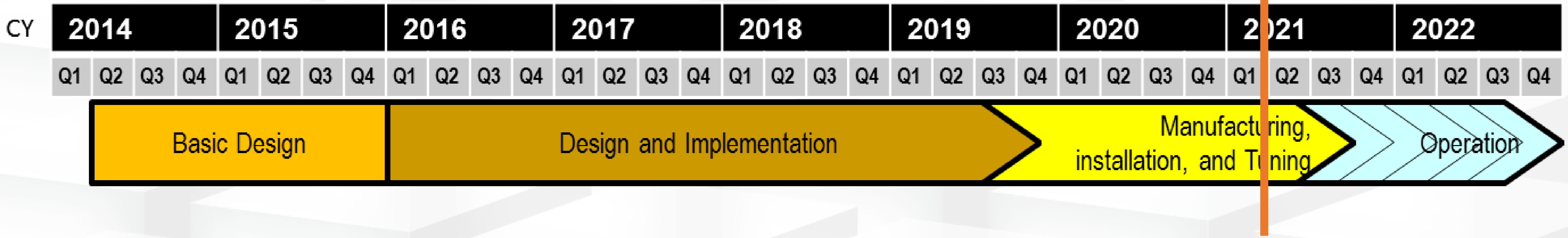


"Science by Computing"

"9 Priority Areas" SDGs goals

‘Fugaku’-FLAGSHIP2020 Project: Mission and Timeline

- **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 our country and all over the world
- **Organization**
  - The RIKEN Center for Computational Science in charge of the research and development of the Post-K - Fugaku
  - Fujitsu is a vendor partner
- **Started from 2014, ended in March, 2021**
- **The service to public users started from March 2021**



Technologies and Architectural Parameters to be determined by Codesign

- **Basic Architecture Design (by Feasibility Studies)**
  - Manycore approach, O3 cores, some parameters on chip configuration and SIMD
- **Instruction Set Architecture and SIMD Instructions**
  - Fujitsu collaborated with Arm, contributing to the design of the SVE as a lead partner
- **Chip configuration**
- **Memory technology**
  - DDR, HBM, HMC
- **Cache structure**
- **Out of order (O3**
- **Enhancement fo**
- **Interconnect between Nodes**
  - SerDes, topologies "Tofu" or other network?

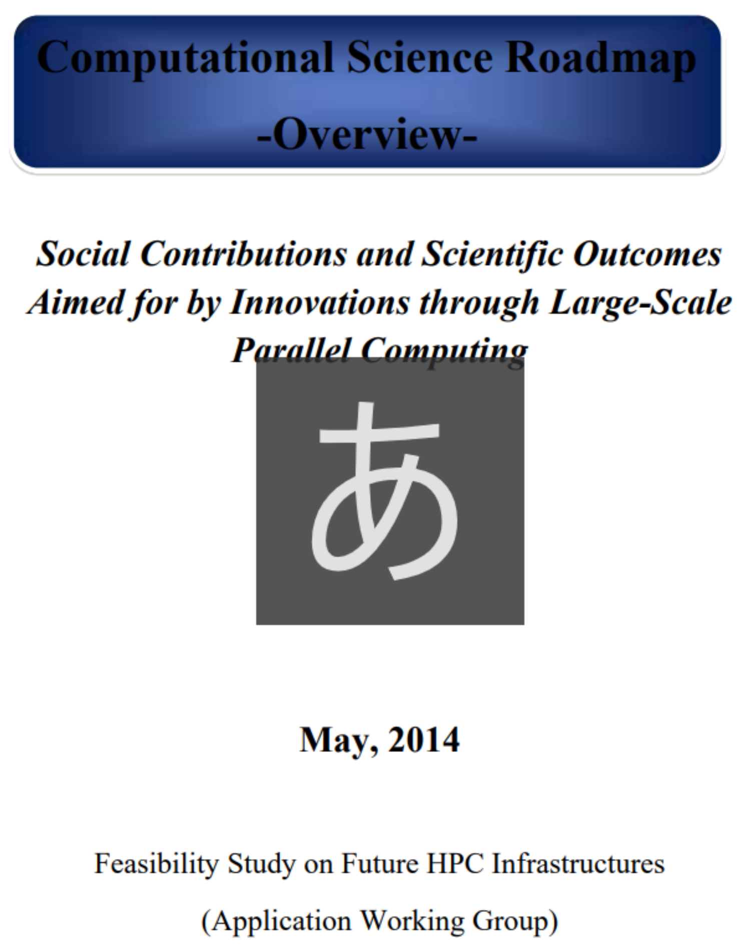
✓ The number of cores in a CMG
✓ The number of CMGs in a chip
s to shared L2 in a CMG
, the size, and throughp
vork-on-chip to connect
✓ The die size of the chip
✓ The number of chips in a node

SC20 technical paper. "Co-Design for A64FX Manycore Processor and "Fugaku""

M. Sato, Y. Ishikawa, H. Tomita, Y. Kodama, T. Odajima, M. Tsuji, H. Yashiro, M. Aoki, N. Shida, I. Miyoshi,K. Hirai, A. Furuya, A. Asato, K. Morita, T. Shimizu

Post-K Application Feasibility Study 2012-2013

https://hpci-aplfs.r-ccs.riken.jp/document/roadmap/roadmap_e_1405.pdf



Computational Science Roadmap

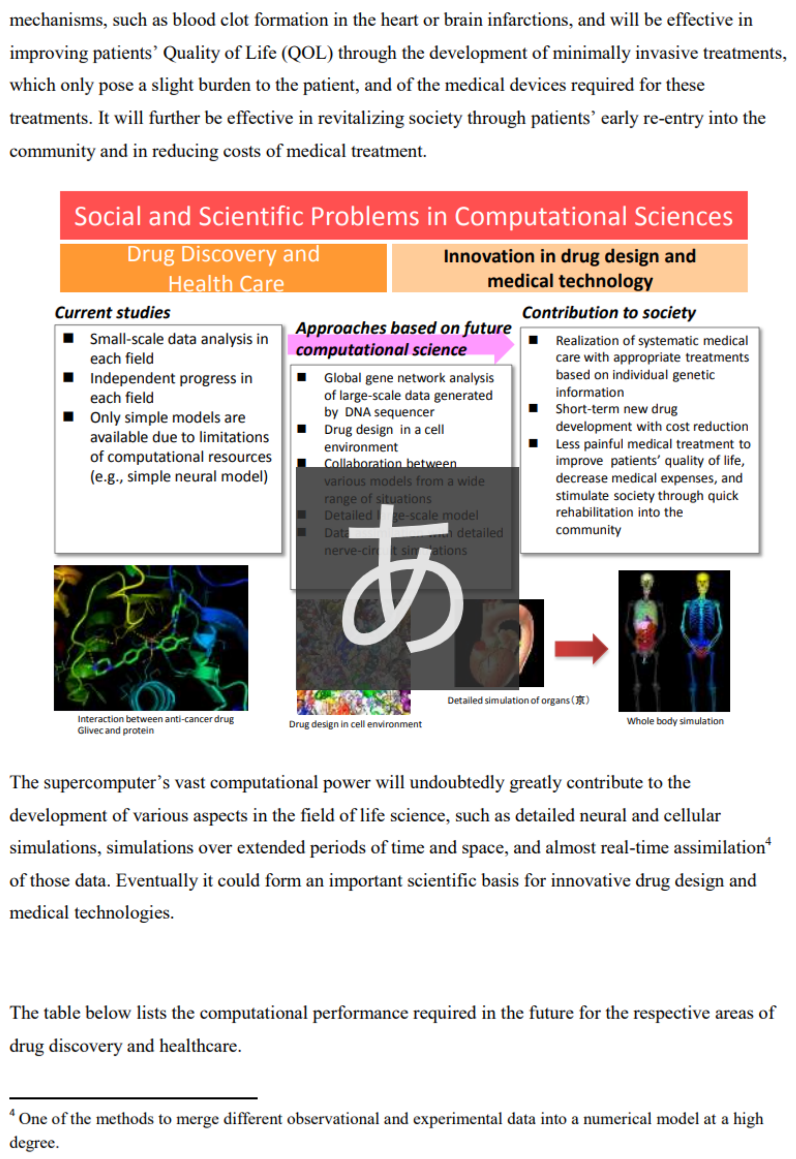
-Overview-

Social Contributions and Scientific Outcomes Aimed for by Innovations through Large-Scale Parallel Computing

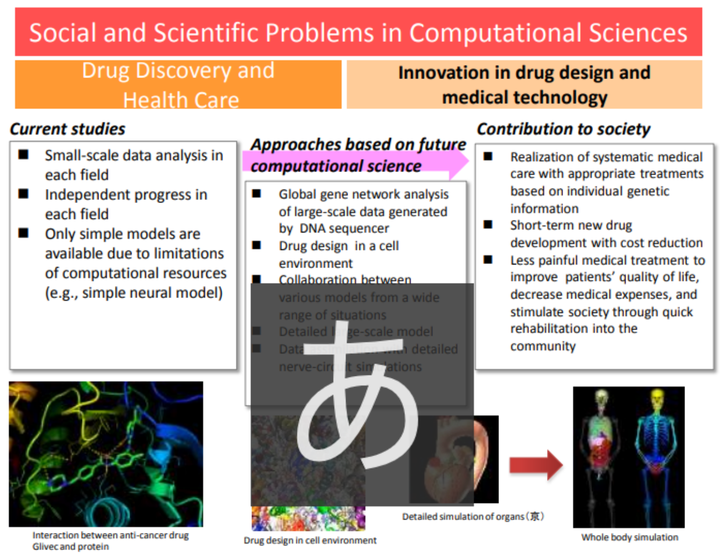
May, 2014

Feasibility Study on Future HPC Infrastructures

(Application Working Group)



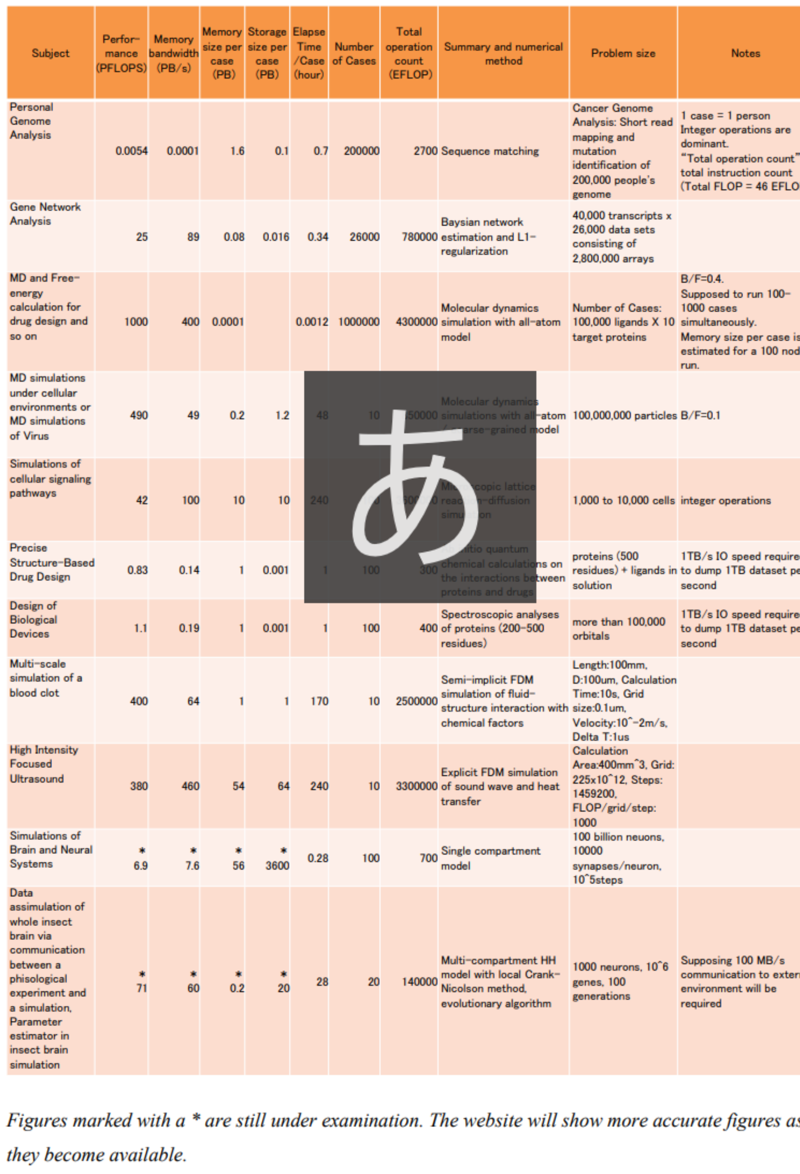
mechanisms, such as blood clot formation in the heart or brain infarctions, and will be effective in improving patients' Quality of Life (QOL) through the development of minimally invasive treatments, which only pose a slight burden to the patient, and of the medical devices required for these treatments. It will further be effective in revitalizing society through patients' early re-entry into the community and in reducing costs of medical treatment.


Social and Scientific Problems in Computational Sciences
Drug Discovery and Health Care
Innovation in drug design and medical technology
Current studies
Small-scale data analysis in each field
Independent progress in each field
Only simple models are available due to limitations of computational resources (e.g., simple neural model)
Approaches based on future computational science
Global gene network analysis of large-scale data generated by DNA sequencer
Drug design in a cell environment
Collaboration between various models from a wide range of situations
Detailed large-scale model
Data assimilation with detailed nerve-circuit simulations
Contribution to society
Realization of systematic medical care with appropriate treatments based on individual genetic information
Short-term new drug development with cost reduction
Less painful medical treatment to improve patients' quality of life, decrease medical expenses, and stimulate society through quick rehabilitation into the community
Interaction between anti-cancer drug Glivec and protein
Drug design in cell environment
Detailed simulation of organs (京)
Whole body simulation

The supercomputer's vast computational power will undoubtedly greatly contribute to the development of various aspects in the field of life science, such as detailed neural and cellular simulations, simulations over extended periods of time and space, and almost real-time assimilation[4] of those data. Eventually it could form an important scientific basis for innovative drug design and medical technologies.

The table below lists the computational performance required in the future for the respective areas of drug discovery and healthcare.

[4] One of the methods to merge different observational and experimental data into a numerical model at a high degree.

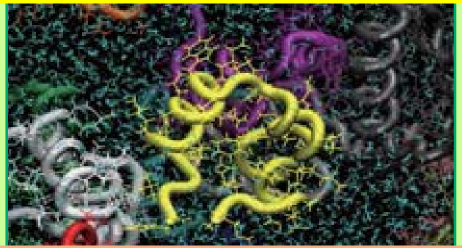


Subject	Perfor-mance (PFLOPS)	Memory bandwidth (PB/s)	Memory size per case (PB)	Storage size per case (PB)	Elapse Time /Case (hour)	Number of Cases	Total operation count (EFLOP)	Summary and numerical method	Problem size	Notes
Personal Genome Analysis	0.0054	0.0001	1.6	0.1	0.7	200000	2700	Sequence matching	Cancer Genome Analysis: Short read mapping and mutation identification of 200,000 people's genome	1 case = 1 person Integer operations are dominant. "Total operation count" total instruction count (Total FLOP = 46 EFLO
Gene Network Analysis	25	89	0.08	0.016	0.34	26000	780000	Baysian network estimation and L1-regularization	40,000 transcripts x 26,000 data sets consisting of 2,800,000 arrays	
MD and Free-energy calculation for drug design and so on	1000	400	0.0001		0.0012	1000000	4300000	Molecular dynamics simulation with all-atom model	Number of Cases: 100,000 ligands X 10 target proteins	B/F=0.4. Supposed to run 100-1000 cases simultaneously. Memory size per case is estimated for a 100 nod run.
MD simulations under cellular environments or MD simulations of Virus	490	49	0.2	1.2	48	10	[illegible]	Molecular dynamics simulations with all-atom / coarse-grained model	100,000,000 particles	B/F=0.1
Simulations of cellular signaling pathways	42	100	10	10	240	[illegible]	[illegible]	Microscopic lattice reaction-diffusion simulation	1,000 to 10,000 cells	integer operations
Precise Structure-Based Drug Design	0.83	0.14	1	0.001	1	100	300	Ab initio quantum chemical calculations on the interactions between proteins and drugs	proteins (500 residues) + ligands in solution	1TB/s IO speed require to dump 1TB dataset pe second
Design of Biological Devices	1.1	0.19	1	0.001	1	100	400	Spectroscopic analyses of proteins (200-500 residues)	more than 100,000 orbitals	1TB/s IO speed require to dump 1TB dataset pe second
Multi-scale simulation of a blood clot	400	64	1	1	170	10	2500000	Semi-implicit FDM simulation of fluid-structure interaction with chemical factors	Length:100mm, D:100um, Calculation Time:10s, Grid size:0.1um, Velocity:10^-2m/s, Delta T:1us	
High Intensity Focused Ultrasound	380	460	54	64	240	10	3300000	Explicit FDM simulation of sound wave and heat transfer	Calculation Area:400mm^3, Grid: 225x10^12, Steps: 1459200, FLOP/grid/step: 1000	
Simulations of Brain and Neural Systems	* 6.9	* 7.6	* 56	* 3600	0.28	100	700	Single compartment model	100 billion neuons, 10000 synapses/neuron, 10^5steps	
Data assimilation of whole insect brain via communication between a phisological experiment and a simulation, Parameter estimator in insect brain simulation	* 71	* 60	* 0.2	* 20	28	20	140000	Multi-compartment HH model with local Crank-Nicolson method, evolutionary algorithm	1000 neurons, 10^6 genes, 100 generations	Supposing 100 MB/s communication to exter environment will be required

*Figures marked with a * are still under examination. The website will show more accurate figures as they become available.*

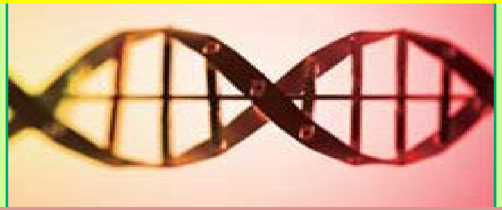
Target science: 9 Priority Areas (Mostly SDGs)

①Innovative Drug Discovery



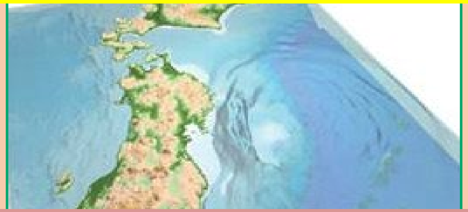
RIKEN Quant. Biology Center

②Personalized and Preventive Medicine



Inst. Medical Science, U. Tokyo

③Hazard and Disaster induced by Earthquake and Tsunami



Earthquake Res. Inst., U. Tokyo

⑧ Innovative Design and Production Processes for the Manufacturing Industry in the Near Future

Cent. for Earth Info., JAMSTEC

⑨Fundamental Laws and Evolution of the Universe

Cent. for Comp. Science, U. Tsukuba

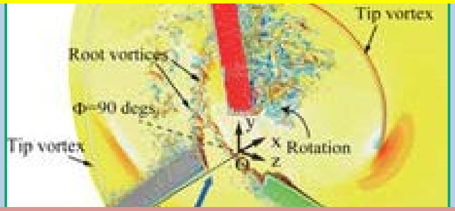
④Environmental Predictions with Observational Big Data

Center for Earth Info., JAMSTEC

⑦New Functional Devices and High-Performance

Inst. For Solid State Phys., U. Tokyo

⑥Innovative Clean Energy Systems



Grad. Sch. Engineering, U. Tokyo

⑤High-Efficiency Energy Creation, Conversion/Storage and Use

Inst. Molecular Science, NINS

One representative 'target app' was picked from each area for co-design, total of 9

Achieve nearly two orders of magnitude speedup, some > 100x

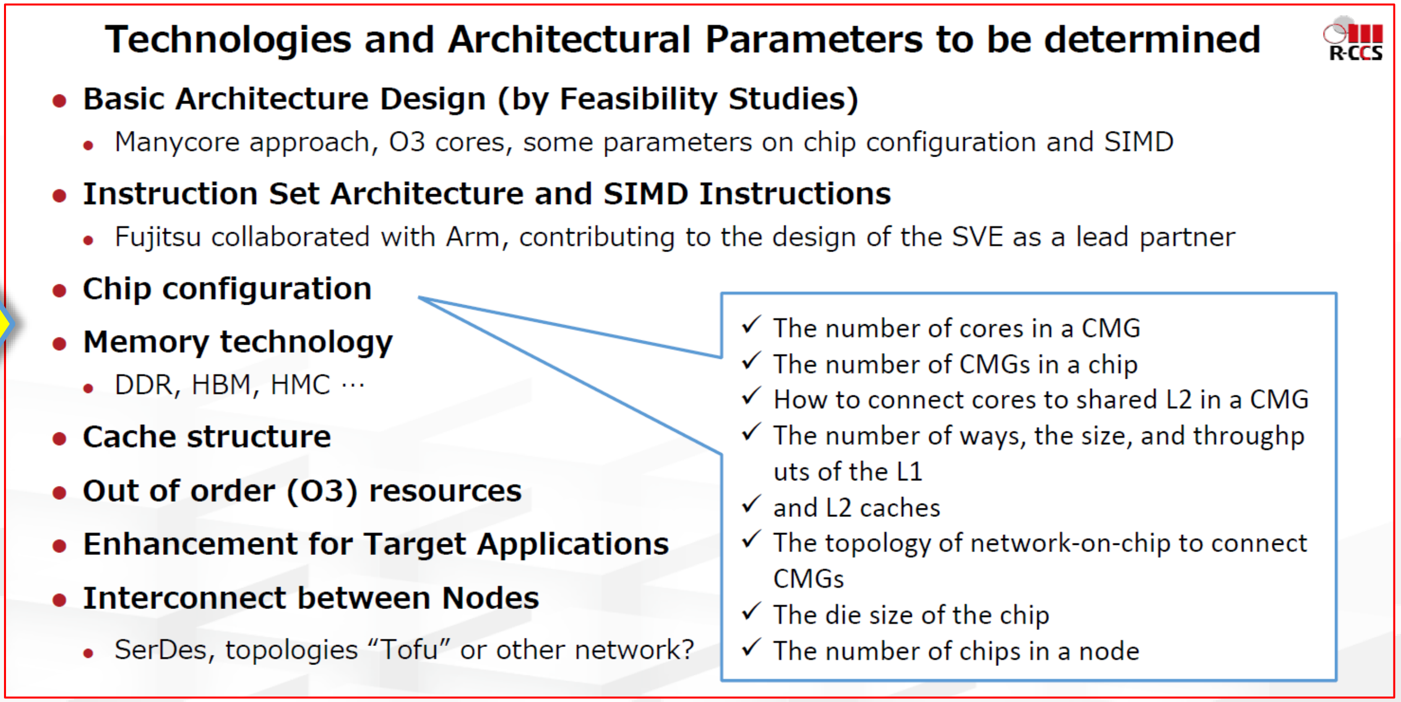
Codesign of "Fugaku"

3 Design Targets:

- **1. Extreme Power-Efficient System**
  - Maximum performance under Power consumption of 30 - 40MW (for system)
- **2. Effective performance of target applications**
  - It is expected to exceed 100 times higher than the K computer's performance in some applications
- **3. Ease-of-use system for wide-range of users**

Cool (Low-power) technology is important!!

Codesign



Technologies and Architectural Parameters to be determined

- **Basic Architecture Design (by Feasibility Studies)**
  - Manycore approach, O3 cores, some parameters on chip configuration and SIMD
- **Instruction Set Architecture and SIMD Instructions**
  - Fujitsu collaborated with Arm, contributing to the design of the SVE as a lead partner
- **Chip configuration**
- **Memory technology**
  - DDR, HBM, HMC …
- **Cache structure**
- **Out of order (O3) resources**
- **Enhancement for Target Applications**
- **Interconnect between Nodes**
  - SerDes, topologies "Tofu" or other network?

✓ The number of cores in a CMG
✓ The number of CMGs in a chip
✓ How to connect cores to shared L2 in a CMG
✓ The number of ways, the size, and throughputs of the L1
✓ and L2 caches
✓ The topology of network-on-chip to connect CMGs
✓ The die size of the chip
✓ The number of chips in a node

Sato et. Al. "Co-Design for A64FX Manycore Processor and 'Fugaku'", ACM/IEEE Supercomputing 2020

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)**

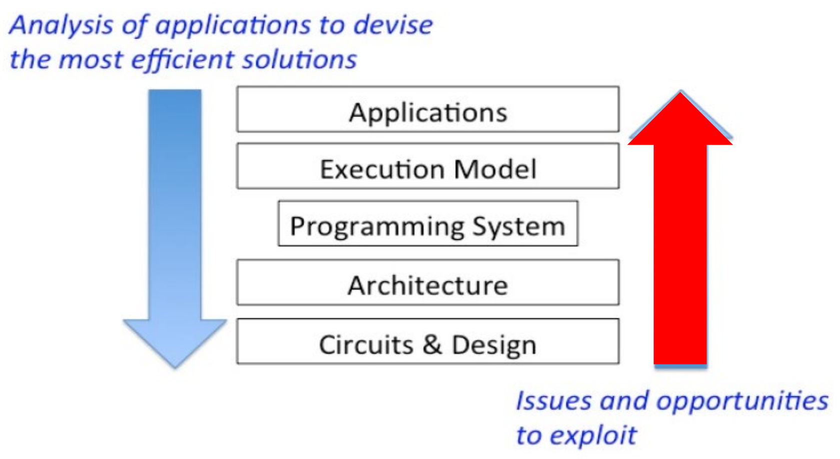
1. **Setting set of system parameters**
2. **Tuning target applications under the system parameters**
3. **Evaluating execution time using prediction tools**
4. **Identifying hardware bottlenecks and changing the set of system parameters**

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

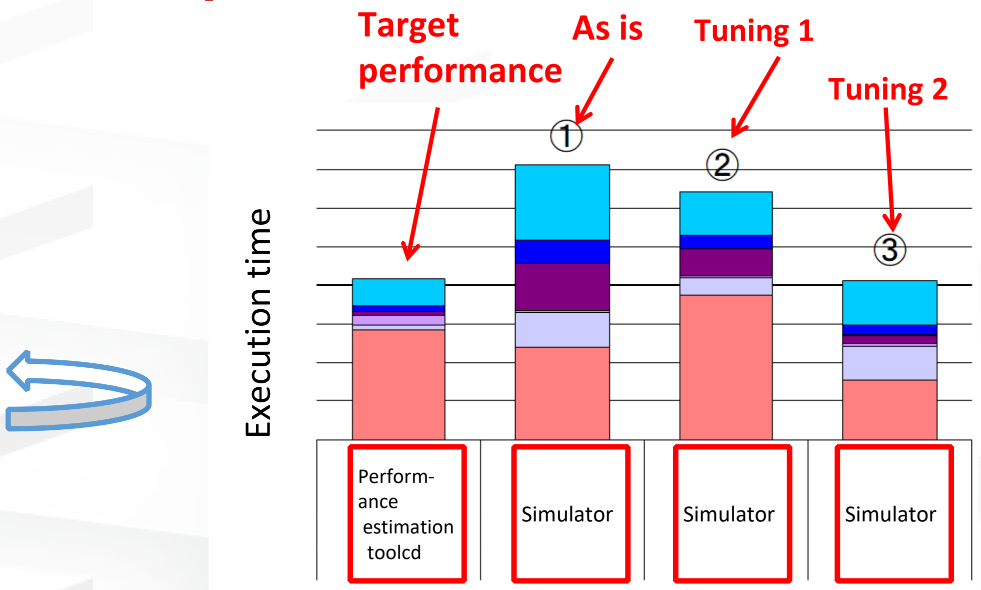
	Target Application	
	Program	**Brief description**
①	GENESIS	MD for proteins
②	Genomon	Genome processing (Genome alignment)
③	GAMERA	Earthquake simulator (FEM in unstructured & structured grid)
④	NICAM+LETK	Weather prediction system using Big data (structured grid stencil & ensemble Kalman filter)
⑤	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)

Co-design of Apps for Architecture

- **Tools for performance tuning**
  - Performance estimation tool – Proxy Arch.
    - Performance projection using Fujitsu FX100 execution profile
    - Gives "target" performance
  - **GEM5 based A64FX processor simulator**
    - **Based on gem5, O3, cycle-level simulation**
    - **Very slow, so limited to kernel-level evaluation (Note: Fujitsu had its private cycle-accurate sim)**


Analysis of applications to devise the most efficient solutions
Applications
Execution Model
Programming System
Architecture
Circuits & Design
Issues and opportunities to exploit

- **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

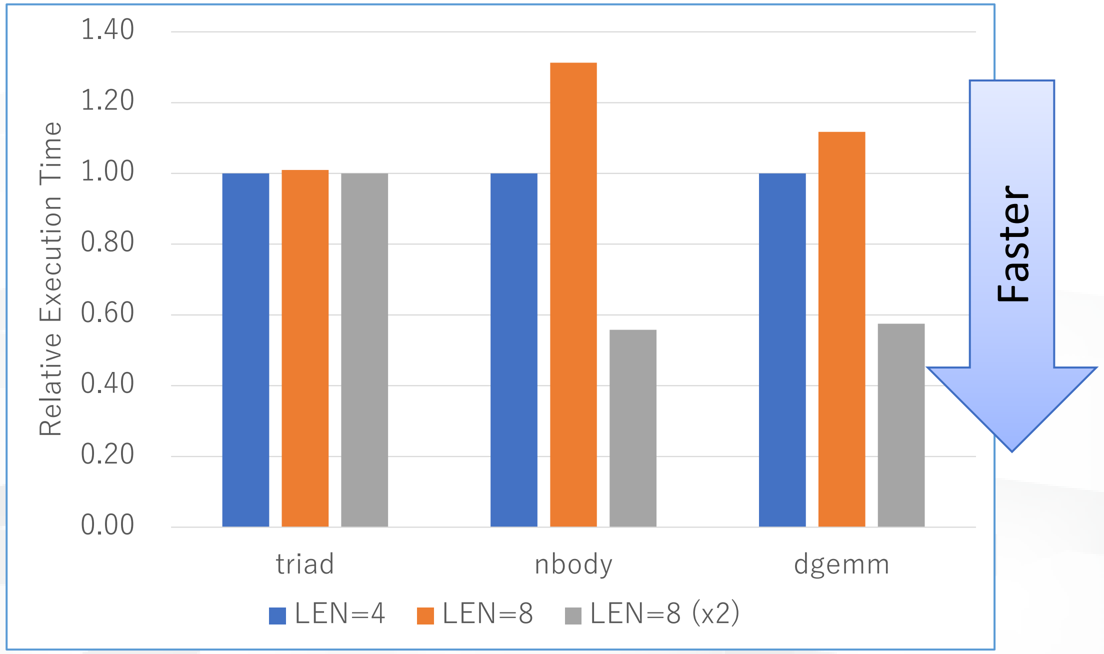


Example: ARM for HPC - Co-design using Riken Gem5 for ArmSVE

- **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 only 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

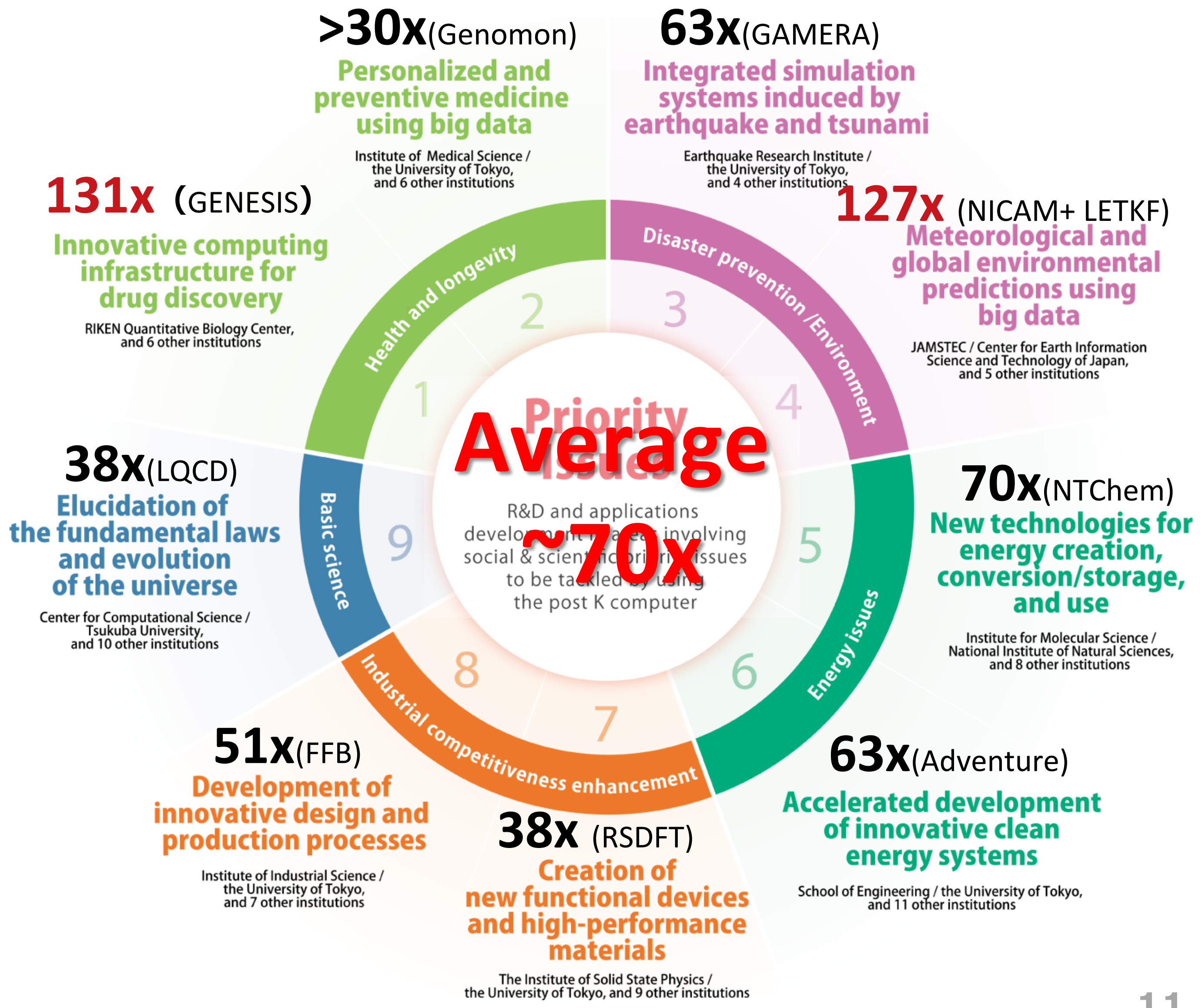


From K computer to Fugaku

	K computer	Fugaku	
Official operation start	Sep. 2012	Mar. 2021	
CPU Architecture	SPARC64VIIIfx 8 core	A64FX(Armv8.2-A SVE) 48 core	
Peak performance DP/SP	11.28 PF/-	488PF/977PF	50x
# of node/rack	82,944/864	158,976/432	2x/0.5x
Voltage	3-phase AC 200V	->	
Peak/average Power	15MW/12MW	35MW/18MW	
Cooling ratio (water vs air)	65:35	90:10	

“Applications First” Exascale R&D
Fugaku Target Applications – Priority Research Areas

- **Advanced Applications Co-Design Program to Parallel Fugaku R&D**
- **Select one representative app from 9 priority areas**
  - Health & Medicine
  - Environment & Disaster
  - Energy
  - Materials & Manufacturing
  - Basic Sciences
- **Up to 100x speedup c.f. K-Computer => achieved!**

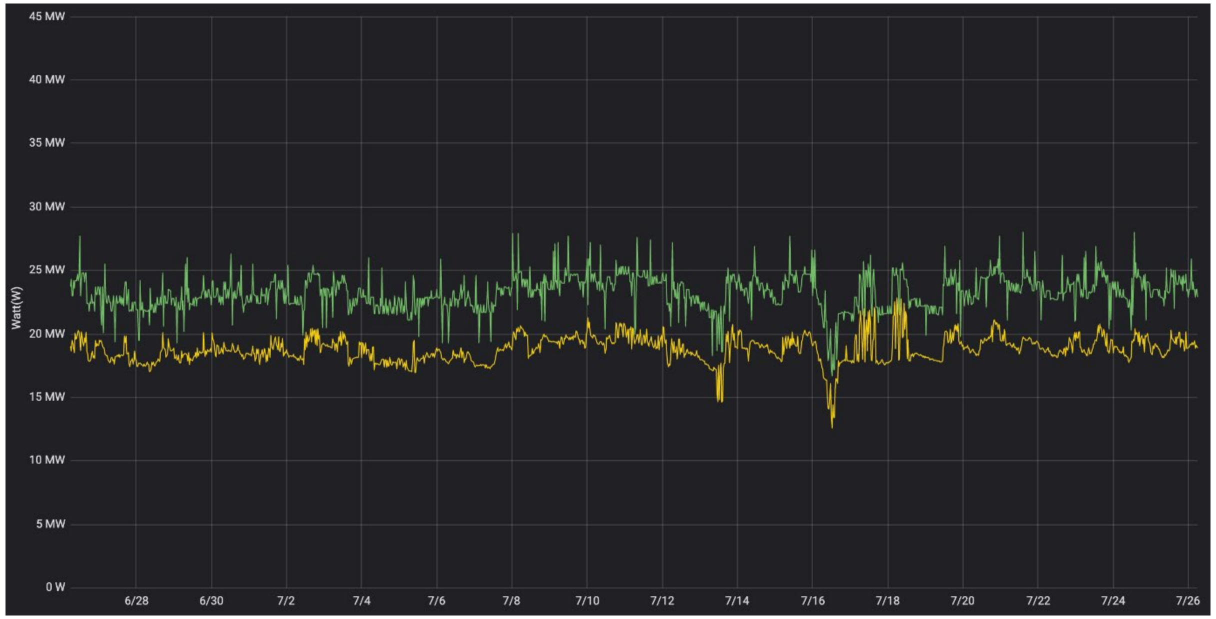


>30x(Genomon)
Personalized and preventive medicine using big data
Institute of Medical Science / the University of Tokyo, and 6 other institutions
63x(GAMERA)
Integrated simulation systems induced by earthquake and tsunami
Earthquake Research Institute / the University of Tokyo, and 4 other institutions
131x (GENESIS)
Innovative computing infrastructure for drug discovery
RIKEN Quantitative Biology Center, and 6 other institutions
127x (NICAM+ LETKF)
Meteorological and global environmental predictions using big data
JAMSTEC / Center for Earth Information Science and Technology of Japan, and 5 other institutions
38x(LQCD)
Elucidation of the fundamental laws and evolution of the universe
Center for Computational Science / Tsukuba University, and 10 other institutions
70x(NTChem)
New technologies for energy creation, conversion/storage, and use
Institute for Molecular Science / National Institute of Natural Sciences, and 8 other institutions
51x(FFB)
Development of innovative design and production processes
Institute of Industrial Science / the University of Tokyo, and 7 other institutions
38x (RSDFT)
Creation of new functional devices and high-performance materials
The Institute of Solid State Physics / the University of Tokyo, and 9 other institutions
63x(Adventure)
Accelerated development of innovative clean energy systems
School of Engineering / the University of Tokyo, and 11 other institutions
Health and longevity
Disaster prevention /Environment
Energy issues
Industrial competitiveness enhancement
Basic science
1
2
3
4
5
6
7
8
9
Priority Issues
R&D and applications development targeting priority issues involving social & scientific priority issues to be tackled by using the post K computer
Average
~70x

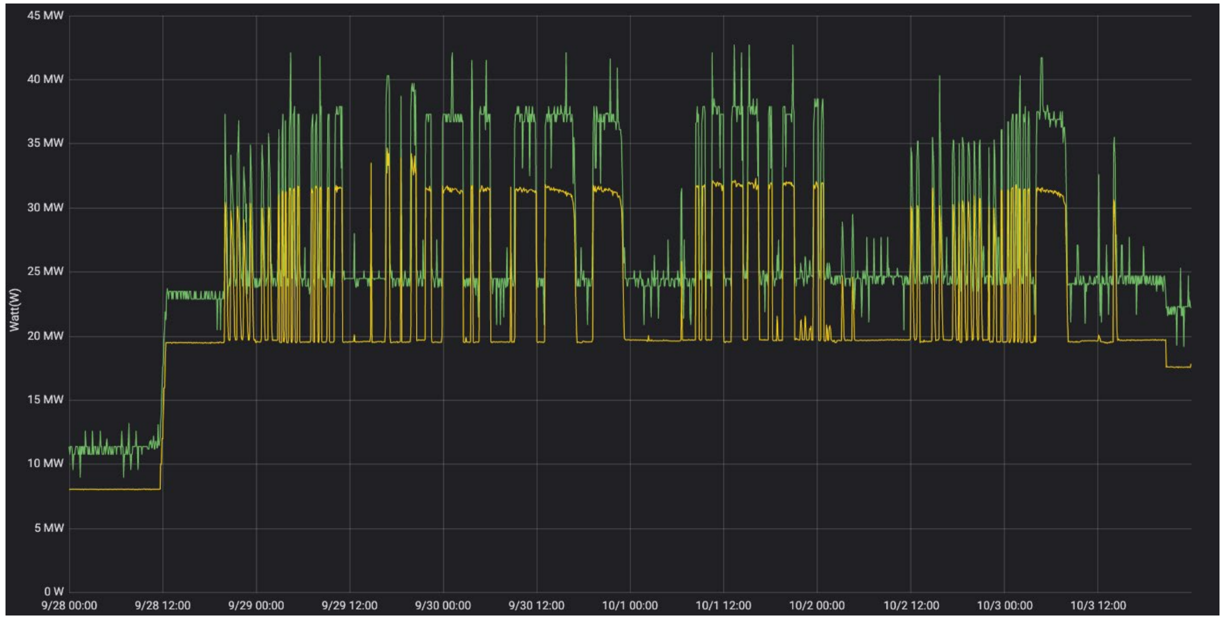
We missed the power target… positively

🟩 site total
🟨 Fugaku

30 days power consumption history



Full node HPCG/HPL measurement



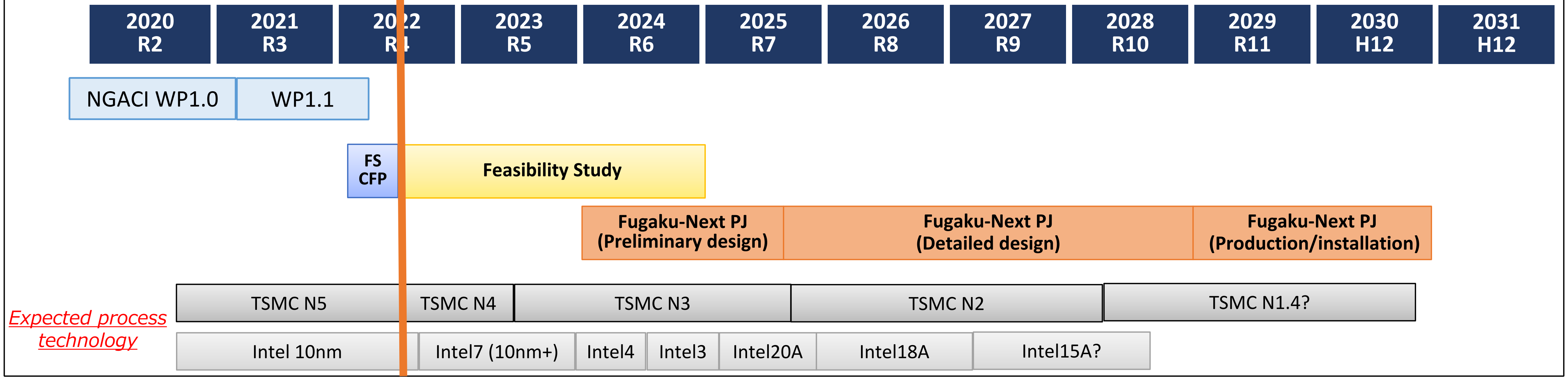
Initial design goal: x2~x3 c.f. K
=> average power consumption
~22-23MW(site total)
~18-19MW(Fugaku) (1.3~1.4x K)
"DoE Goal: Exascale at 20 MW"

max power consumption (HPCG)
42.70MW(site total)
34.66MW(Fugaku)
power swing ~15MW

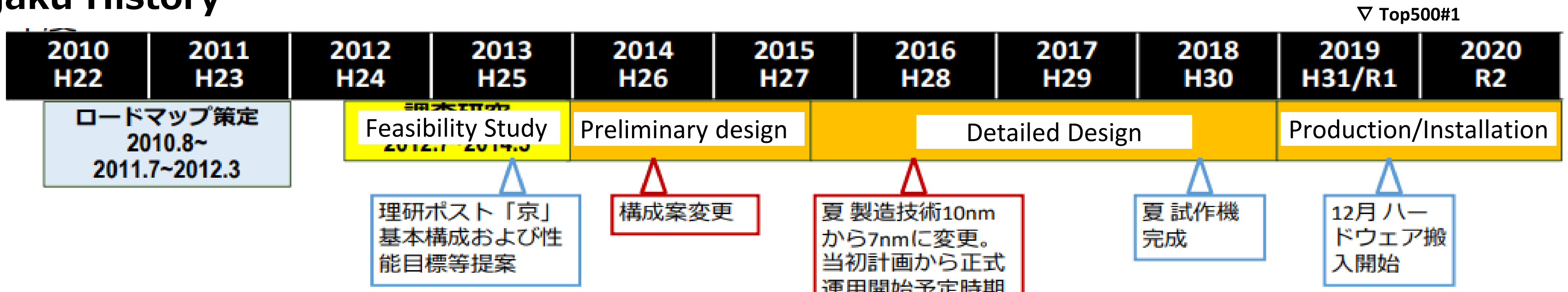
120~130W/node (CPU, HBM, TOFU-HCA&AOC, PSU,…) => Goal now to achieve ~100W/node due to energy crisis

Expected Schedule towards Fugaku-Next Involving JP & US vendors

Fugaku-Next Expected Schedule



Fugaku History



https://www.ssken.gr.jp/MAINSITE/event/2020/20210121-sci/lecture-01/20210121_sci_ishikawa.pdf

Exascale and beyond ‘myths’ to be debunked

- “Co-design with proxy apps is the ***best*** method for designing an effective exascale machine”
- “Compute centric AI friendly chips (with dense concentration of ALUs) will dominate supercomputing”
- “Supercomputers will become a plethora of domain specific heterogeneous accelerators beyond exascale”
- “Zettascale is the next goal beyond exascale (in 2027)”
- “Quantum computers will completely supersede ALL ‘classical’ supercomputers” (another talk another day)

Co-design outcome: A64FX processor and #Fugaku

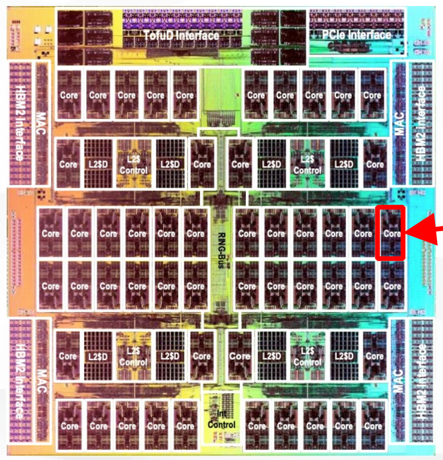
- **HPC-oriented design**
  - Small core ⇒ Less O3 resources
  - (Relatively) Long pipeline
    - 9 cycles for floating point operations
    - Core has only L1 cache
  - High-throughput, but long-latency
  - Pipeline often stalls
    for loops having complex body.

	A64FX	Skylake
ReOrder Buffer	128 entries	224 entries
Reservation Station	60 (=10x2+20x2) entries	97 entries
Physical Vector Register	128 (=32 + 96) entries	168 entries
Load Buffer	40 entries	72 entries
Store Buffer	24 entries	56 entries

A64FX : https://github.com/fujitsu/A64FX
Skylake : https://en.wikichip.org/wiki/intel/microarchitectures/skylake_(server)

- A64FX: 52 cores (48 cores), 400+ mm² die size (8.3 mm²/core), 7nm FinFET process (TSMC)
- Xeon Skylake: 20 tiles (5x4), 18 cores, ~485 mm² die size (estimated) (26.9 mm²/core), 14 nm process (Intel)
- A64FX core is more than 3 times smaller per core.

A64FX:
400 mm²
(20 x 20)



Xeon Skylake, High Core Count:
4 x 5 tiles, 18 cores, 2 tiles used for memory interface
485 mm² (22 x 22)

https://www.fujitsu.com/jp/solutions/business-technology/tc/catalog/ff2019-post-k-computer-development.pdf

https://en.wikichip.org/wiki/intel/microarchitectures/skylake_(server)

SPEC HPC performance – grossly divergent performance

- Fugaku (12 thread x 12 ranks) vs. Ice Lake (2-socket x 36-core x hyperthreading)
- Most of the speedup comes from bandwidth bound Fortran code

Benchmark (12x12)	ratio	exec time (s)	GFLOPS /core	Mem GB/s /core	SIMD inst rate	SVE op rate	IPC	Xeon 8360Y	A64FX /Xeon
505.lbm_t	2.81	789	2.56	0.44	20.6%	60.3%	0.78	5.14	54.7%
513.soma_t	3.32	1111	0.92	0.38	9.2%	49.3%	0.90	9.04	36.7%
518.tealeaf_t	4.01	411	0.66	3.22	1.0%	8.7%	1.11	2.63	152.5%
519.clvleaf_t	11.70	131	4.49	9.60	33.4%	91.3%	0.93	3.03	386.1%
521.miniswp_t	2.69	590	1.08	0.39	0.6%	0.2%	1.47	7.10	37.9%
528.pot3d_t	17.50	120	1.44	15.60	41.2%	99.9%	0.43	2.58	678.3%
532.sph_exa_t	1.27	1525	0.73	0.19	4.7%	0.2%	0.73	6.90	18.4%
534.hpgmgfv_t	2.53	465	0.82	2.39	0.4%	0.8%	1.51	2.97	85.2%
535.weather_t	21.90	146	3.84	7.91	49.6%	100.0%	0.69	5.80	377.6%
	4.84	5287						4.53	106.7%

Significant ongoing SW work to make A64FX robust to general apps, but fundamentally difficult

“Dark” side of codesign with small set of proxy apps

- The architecture ‘overfits’ to a small set of target apps
  - Difficult to cover all applications and workloads (as Intel/AMD processors) **– similar to overfitting in DL**
  - We need methodologies to make co-designed architecture robust – **similar to generalization in DL**
- Straight-line harmonious progression from existing hardware proxies and proxy apps of the time only results in evolutionary architectures
  - E.g. AI/ML workloads were not initially considered, inclusion of half precision HW SVE + OneDNN for SVE was disruptively incorporated at the very last stage of the project
  - Need inject disruptive architectural ideas, continuous compete & mingle with immediate evaluation to select – **similar to genetic algorithms** (aka Darwinian evolution)

Benchmarking and Performance modeling efforts on Fugaku at R-CCS

- **Broad Application selections (as in broad data sets for DL)**
  - R-CCS production apps
  - Major benchmark apps（ECP, PolyBench, SPEC OMP, Rodina, etc.）from US, EU, Asia, industry, …
- **Broad Benchmarking platform across leadership SC centers (as in multi-network training in DL)**
  - Intel Xeon IceLake/CascadeLake (at U-Tokyo)
  - GPU: A100 (at AIST, …), MI250 (at CSC)
  - AMD Milan-X (at CSC)
  - Intel Sapphire Rapids and others ()
- **Continuous benchmarking platform (as in genetic algorithms)**
  - Performance improvement/sanity check on various versions of system software (continuous BM)
  - Large scale performance study for applications' characteristics exploration
  - Basic performance data acquisition for Fugaku-Next study
  - Continuous assessment for accommodating and evaluating "what if" ideas rapidly
- **'Octopodes' or parameterizable Berkeley Dwarf-like kernels (as in augmentation in DL)**
  - Extract application kernels and make them parametrizable
  - Apps performance model as composition of parameterized octopodes
  - For details, S. Matsuoka *et al*., "Preparing for the Future—Rethinking Proxy Applications," in *Computing in Science & Engineering*, vol. 24, no. 2, pp. 85-90, 1 March-April 2022, doi: 10.1109/MCSE.2022.3153105. also available in ArXiv.

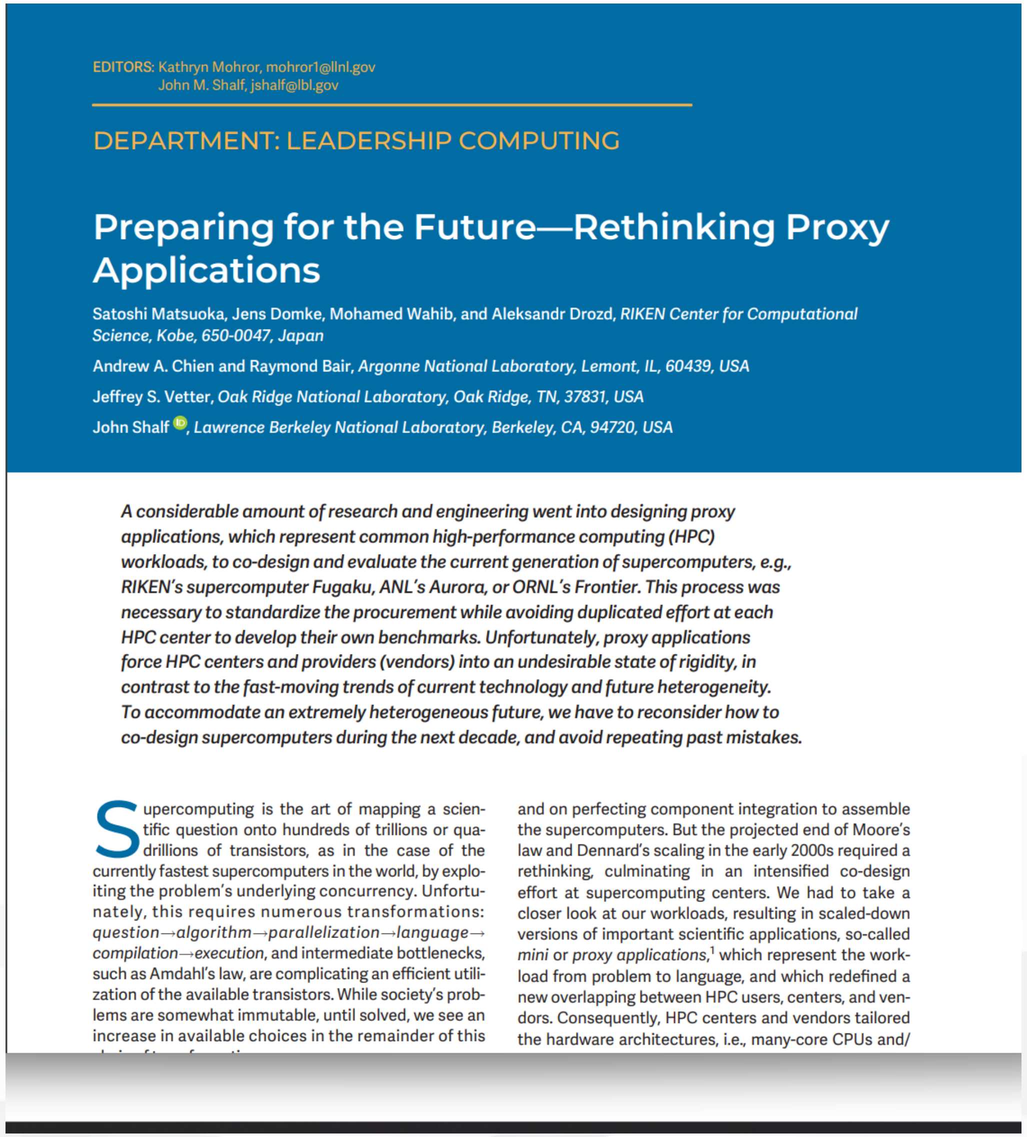
Benchmark list and result available? (as of 3/31/2022)

Category	Team or Benchmark Suite	App name	Result (03312022)	source code?	scalability test	Remarks
R-CCS Apps	Computational Climate Science	SCALE	○	○		Climate Simulation
	Field Theory Research	Bridge++	○	○	○	QCD
		QWS	○	○	○	QCD
	Computational Biophysics	GENESIS	○	○	○	MD
		Gromacs, NAMD, LAMMPS	○	○	○	MD
	Computational Molecular Science	NTChem	○	—	○	Quantum Chemistry
		CP2K	○	○		Quantum Chemistry
		BigDFT				
		NWChem				Quantum Chemistry
	Computational Structural Biology	RELION	○	○		Biopolymer analysis
	Complex Phenomena Unified Simulation	CUBE	○	—		CUBE(Complex Unified Buil
		FrontFlow/red-HPC	○	—	○	Thermal fluid dynamics
	Data Assimilation	NICAM-LETKF				Global Numerical Weather
		resnet_channels.py				Neural network based multi
	High Performance Artificial Intelligence	NEST				Brain simulation
		MONET				Brain simulation
		DeepBench	○	—		AI
		Alex's Benchmarker				AI
		MLPerf, MLPerf HPC	○	—		AI
		CosmoFlow	○	—	●	MLPerfHPC
	Computational Materials Science	qNET				DMRG
		Turbo-RVB				QMC
	High Performance Big Data	Intel HiBench	○	—		
	Large-scale Parallel Numerical Computing	EigenExa, ScaLAPACK, ELPA, SLATE, PETSc, SLEPc, kokkos, FFTE-C				Numerical ribrary

Category	Team or Benchmark Suite	App name	Result (03312022)	source code?	scalability test	Remarks
top500	top500 Benchmarking	HPL	○	○		Linpack
		HPCG	○	○		CG
		HPL-AI	○	○	○	Linpack (single precision)
		Graph500	○	○	○	Graph
US	DoE/ECP Proxy Apps	AMG	△	●		Algebraic Multi-Grid linear sys
		CANDLE	△	●		These codes implement deep
		Laghos	△	●		Laghos computes compressi
		MACSio	△	●		MACSio is being developed to
		miniAMR	△	●		miniAMR applies a stencil cal
		miniFE	△	●		MiniFE is an proxy application
		miniTri	△	●		This directory contains differe
		Nekbone	△	●		Nekbone solves a standard P
		SW4lite	△	●		SW4lite is lite version of SW4
		SWFFT	△	●		The Hardware Accelerated C
		XSBench	△	●		XSBench is a mini-app repres
		Lulesh	△	●		Shock hydrodynamics for uns
Standard BM	SPEC	SPEC OMP	○	●		
		SPEC MPI		●		
		SPEC HPC	○	●		
		SPEC CPU	○	●		
Quantum	Quantum Comp. Simulation	qulacs				
		blaket				
Commercia	from RIST	OpenForm	○	○	○	
		lammps	○	○	○	
	Others?					

“Octopodes”

- Essentially, extension of Berkely Dwarf
- Extract compute kernels and their essential parameters, turn them into ‘octopodes’
- Proxy app performance model made of compositions of parameterized performance models
- By varying the individual parameters, we should obtain parameterizable performance model for the whole app, allowing performance models to be constructed easily
- By artificially varying the parameters for performance model ‘augmentation’, we could avoid the ‘overfitting’ problem in co-design



EDITORS: Kathryn Mohror, mohror1@llnl.gov
John M. Shalf, jshalf@lbl.gov

DEPARTMENT: LEADERSHIP COMPUTING

Preparing for the Future—Rethinking Proxy Applications

Satoshi Matsuoka, Jens Domke, Mohamed Wahib, and Aleksandr Drozd, *RIKEN Center for Computational Science, Kobe, 650-0047, Japan*

Andrew A. Chien and Raymond Bair, *Argonne National Laboratory, Lemont, IL, 60439, USA*

Jeffrey S. Vetter, *Oak Ridge National Laboratory, Oak Ridge, TN, 37831, USA*

John Shalf, *Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, USA*

A considerable amount of research and engineering went into designing proxy applications, which represent common high-performance computing (HPC) workloads, to co-design and evaluate the current generation of supercomputers, e.g., RIKEN’s supercomputer Fugaku, ANL’s Aurora, or ORNL’s Frontier. This process was necessary to standardize the procurement while avoiding duplicated effort at each HPC center to develop their own benchmarks. Unfortunately, proxy applications force HPC centers and providers (vendors) into an undesirable state of rigidity, in contrast to the fast-moving trends of current technology and future heterogeneity. To accommodate an extremely heterogeneous future, we have to reconsider how to co-design supercomputers during the next decade, and avoid repeating past mistakes.

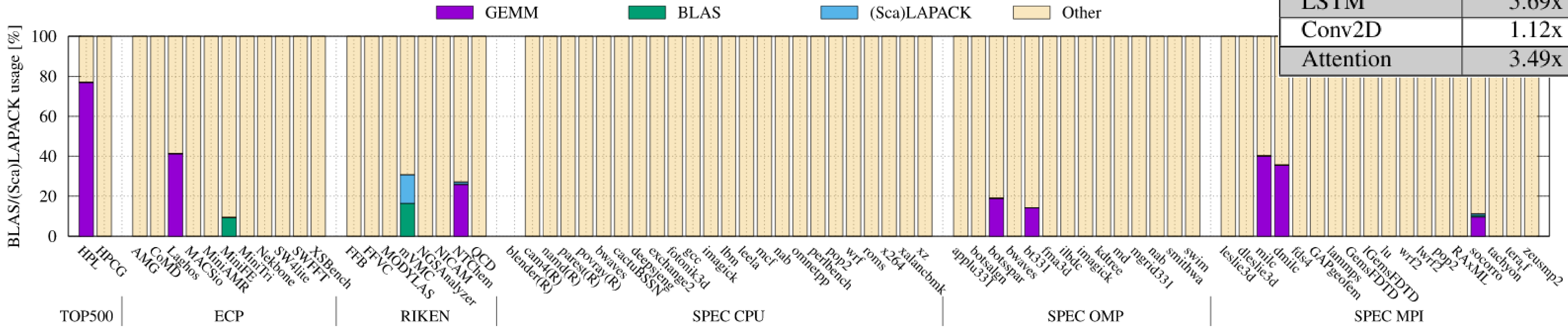
Supercomputing is the art of mapping a scientific question onto hundreds of trillions or quadrillions of transistors, as in the case of the currently fastest supercomputers in the world, by exploiting the problem’s underlying concurrency. Unfortunately, this requires numerous transformations: *question→algorithm→parallelization→language→compilation→execution*, and intermediate bottlenecks, such as Amdahl’s law, are complicating an efficient utilization of the available transistors. While society’s problems are somewhat immutable, until solved, we see an increase in available choices in the remainder of this

and on perfecting component integration to assemble the supercomputers. But the projected end of Moore’s law and Dennard’s scaling in the early 2000s required a rethinking, culminating in an intensified co-design effort at supercomputing centers. We had to take a closer look at our workloads, resulting in scaled-down versions of important scientific applications, so-called *mini* or *proxy applications*,[1] which represent the workload from problem to language, and which redefined a new overlapping between HPC users, centers, and vendors. Consequently, HPC centers and vendors tailored the hardware architectures, i.e., many-core CPUs and/

Linpack considered harmful --- BLAS / GEMM utilization in HPC Applications [Domke et. al.@R-CCS, IPDPS2020]

- **Analyzed various data sources:**
  - **Historical data from K computer**: only 53,4% of node-hours (in FY18) were consumed by applications which had GEMM functions in the symbol table
  - **Library dependencies**: only 9% of Spack packages have *direct* BLAS lib dependency (51.5% have indirect dependency)
  - **TensorCore benefit for DL**: up to 7.6x speedup for MLperf kernels
  - **GEMM utilization in HPC**: sampled across 77 HPC benchmarks (ECP proxy, RIKEN fiber, TOP500, SPEC CPU/OMP/MPI) and measured/profiled via Score-P and Vtune

Benchmark	Speedup
BERT	3.39x
Cosmoflow	1.16x
VGG16	1.71x
Resnet50	1.97x
DeepLabV3	1.75x
SSD300	1.78x
NCF	0.97x
GEMM	7.59x
GRU	3.67x
LSTM	5.69x
Conv2D	1.12x
Attention	3.49x

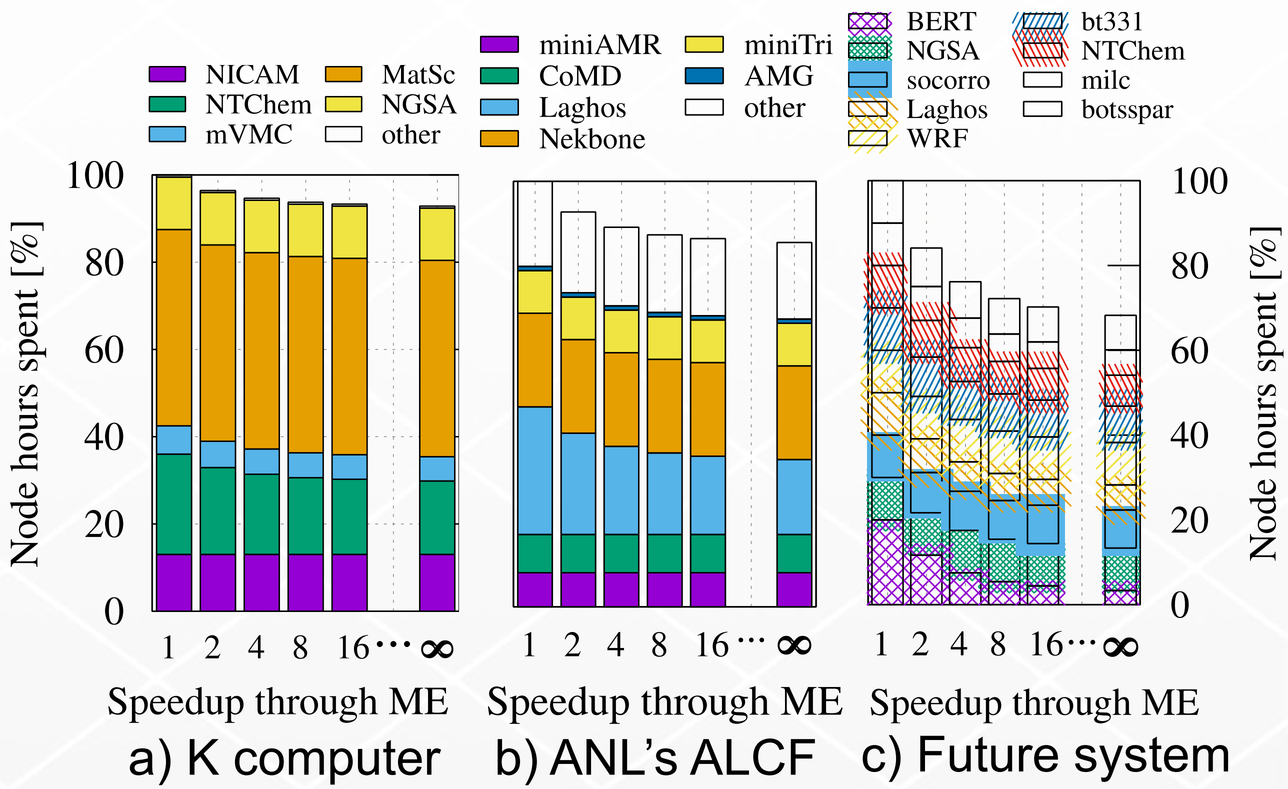


Jens Domke, Emil Vatai, Aleksandr Drozd, Peng Chen, Yosuke Oyama, Lingqi Zhang, Shweta Salaria, Daichi Mukunoki, Artur Podobas, Mohamed Wahib, Satoshi Matsuoka. “Matrix Engines for High Performance Computing:A Paragon of Performance or Grasping at Straws?”, IEEE IPDPS 2020

Q: "How much performance gain can we expect with 'infinite' matrix engine speedup?"

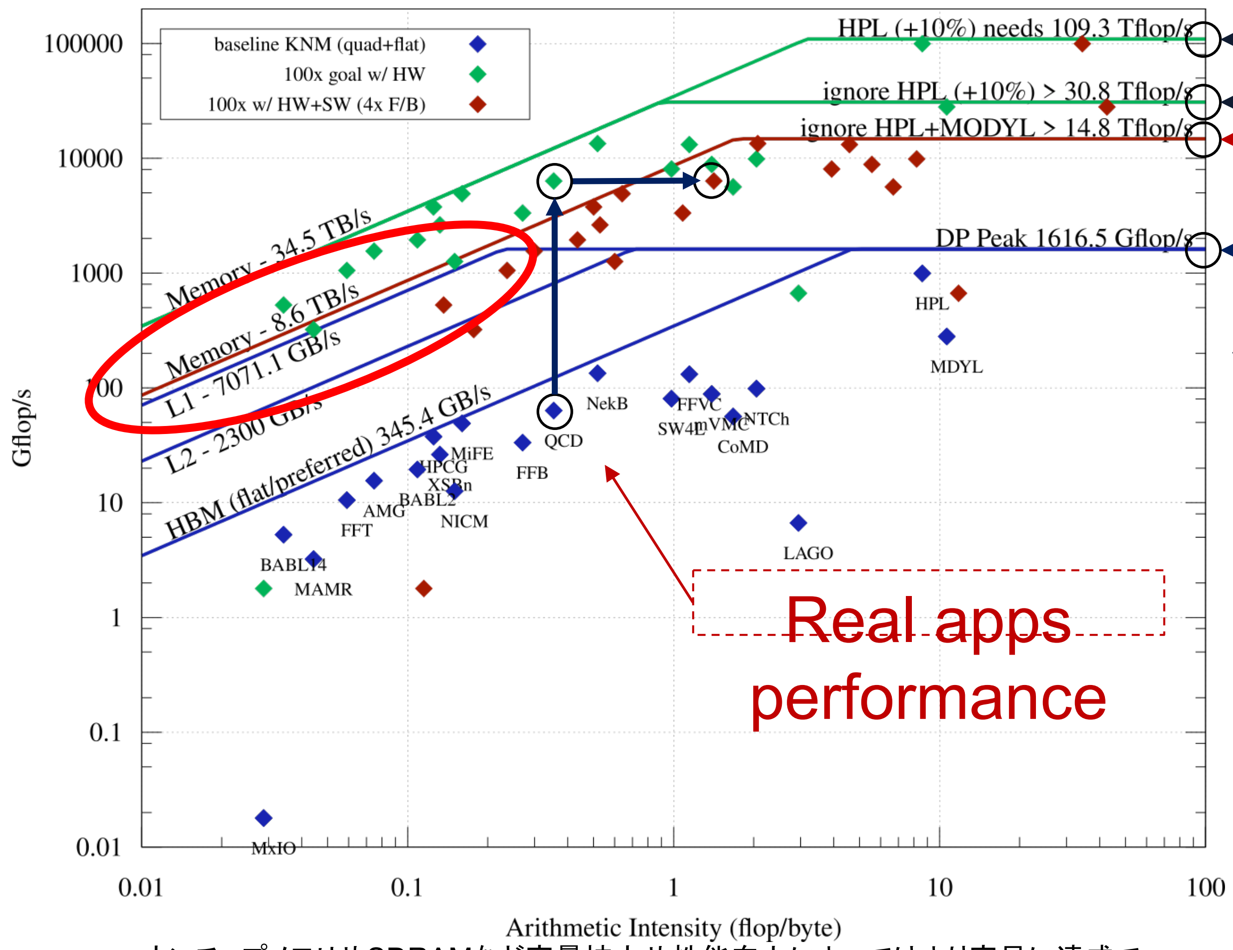
Computer simulations create the future

- We extrapolate node hours spent while assuming that applications were accelerated by a ME for all GEMM portions
- We select for each domain a represntitive benchmark(s)
- Different levels of speedups (up to infinitly fast MEs)
- Results: 7.1% for K; 10.8% for ANL; 32.8% for future sysem (∞ ME) ➔ 'marginal' at best…



Node hours reduced by utilizing hypothetical MEs. Breakdown of node hours per science domain based on historical data [a) and b)]. Hypothetical system c) assumed to execute 20% AI/ML tasks

How to achieve our performance target for dominant memory-bound HPC applications?



現行のマシンでのアプリの性能

倍精度演算性能：1.6 Tflop/s
メモリ性能：0.35 TB/s

ハードウェアのみで
性能100倍を達成する場合

シナリオ1
全てのベンチマークで100倍達成
（要求）倍精度演算性能：110 Tflop/s
（要求）メモリ性能：34.5 TB/s

シナリオ2
（HPLを除く）全てのベンチマークで100倍達成
（要求）倍精度演算性能：30.8 Tflop/s
（要求）メモリ性能：34.5 TB/s

ハードウェア/アプリチューニング
性能100倍を達成する場合

シナリオ3
（HPL, MODYLASを除く）低精度演算によりF/Bが4倍向上した場合
（要求）倍精度演算性能：15 Tflop/s
（要求）メモリ性能：8.6 TB/s

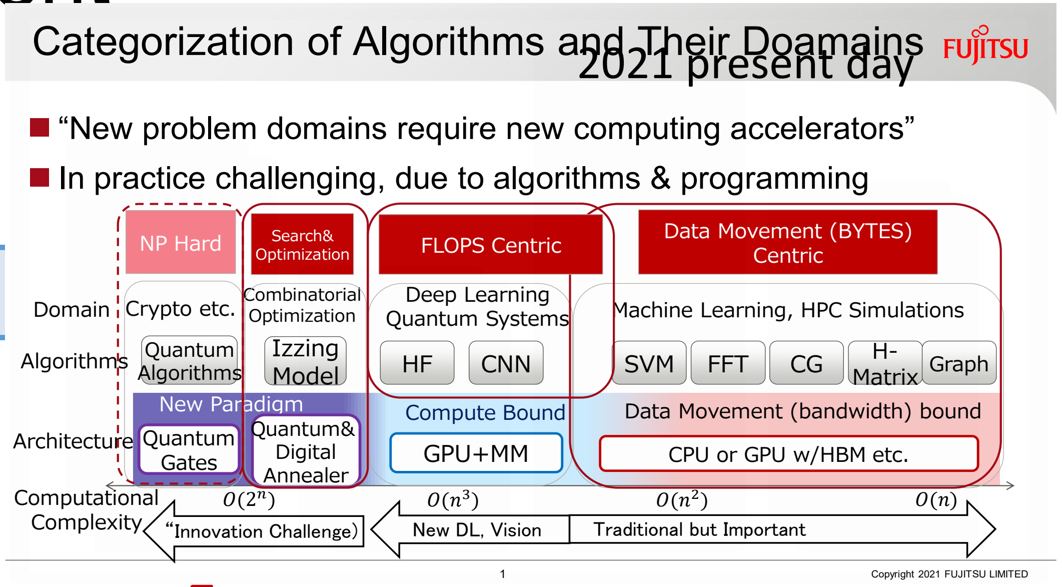
- オンチップメモリやSDRAMなど容量拡大や性能向上によってはより容易に達成できる可能性もある
- データフロー、Massive Cores in Memory side、アルゴリズム向上による演算密度の向上

→ それらを考慮したより広範な探索も今後行う

* GPU V100: 7.45 Tflops (0.9 TB/s)
* A64FX: 2.7 Tflops (1TB/s)

Non-Quantum and Quantum Future Algorithmic Development

- **Towards 2030 Post-Moore era**
  - End of ALU compute (FLOPS) advance
  - Disrupitve reduction in data movement cost with new devices, packaging
  - Algorithm advances to reduce the computational order (+ more reliance on data movement)
  - Unification of BD/AI/Simulation towards data-centric view

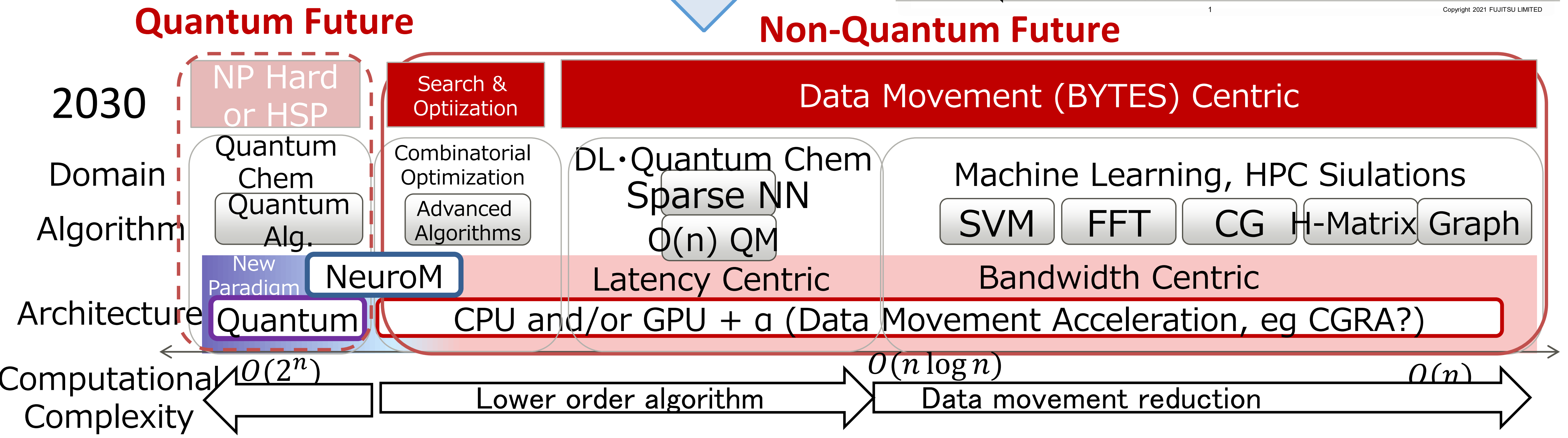


Categorization of Algorithms and Their Doamains

2021 present day

- "New problem domains require new computing accelerators"
- In practice challenging, due to algorithms & programming

FUJITSU
NP Hard
Search& Optimization
FLOPS Centric
Data Movement (BYTES) Centric
Domain
Crypto etc.
Combinatorial Optimization
Deep Learning Quantum Systems
Machine Learning, HPC Simulations
Algorithms
Quantum Algorithms
Izzing Model
HF
CNN
SVM
FFT
CG
H-Matrix
Graph
New Paradigm
Compute Bound
Data Movement (bandwidth) bound
Architecture
Quantum Gates
Quantum& Digital Annealer
GPU+MM
CPU or GPU w/HBM etc.
Computational Complexity
$O(2^n)$
$O(n^3)$
$O(n^2)$
$O(n)$
"Innovation Challenge)
New DL, Vision
Traditional but Important
1
Copyright 2021 FUJITSU LIMITED

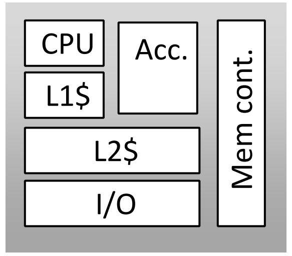


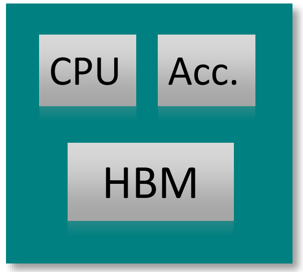
Are Domain-Specific Accelerators Useful for HPC?

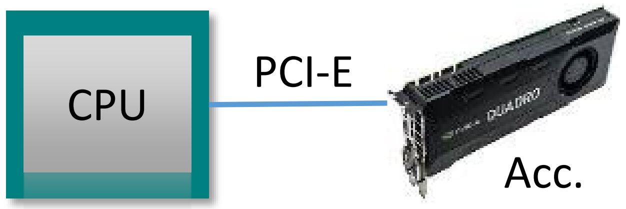
- On chip integration（SoC）
  - Accelerator on the same die with CPU or even embedded within a CPU (e.g. vector/matrix engines within CPU cores)n
  - Shared various resources with CPUs e.g. on-chip cache
  - low energy of data movement, homogeneous across nodes.
- Multi-chip packaging
  - Interconnect accelerator chiplets with CPU chiplets using interposers etc.
  - Shared main memory, medium energy data movement
- On-Node accelerators + CPUs
  - Accelerator – CPU connection via standard chip-chip interconnect e.g. PCI-E, CXL, CAPI
  - Low bandwidth, higher energy of data movement
  - Scalable if homogeneous and workload exclusive to ACC or CPU
- Specific accelerated nodes/machines, via LAN or even WAN
  - Expensive data movement, workload largely confined to each
  - Limited utility, high cost of heterogeneous management, not scalable
  - Only makes sense if workload is well known and largely fixed

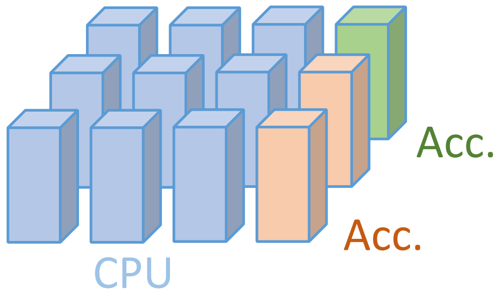
➔ **Accelerators are *means to and end*, not a purpose by itself**

➔ **Need detailed analysis of the workloads & their evolutions from which accelerators are defined, not the other way around**









Application Kernel Categorization & SC Architecture

Compute Bound (aka Top500)

Bandwidth Bound (aka HPCG)

Latency Bound (aka Graph500)

Classic Vector (e.g. Earth Simulator) ~90s

COTS-CPU based clusters late 90s~late 2000s (ASCI XXX, Tsubame1/T2K, Jaguar, K)
Standard Memory Technologies (DDR DRAM), Massively Parallel

GPU

CPU

GPU-Based 'Heterogeneous' Machines: high (compute & BW & latency) for GPU
Tsubame2/3, ABCI, Summit, Piz-Daint, Fronter, Aurora, …

Fugaku/A64FX, Sapphire Rapids: incorporating high bandwidth vectors & Good SW Ecosystem

GPU/Matrix

CPU

Unexplored but good? (programmability, performance, industry adoption, …)

Strong Scaling CGRA/Matrix

CPU/PIM

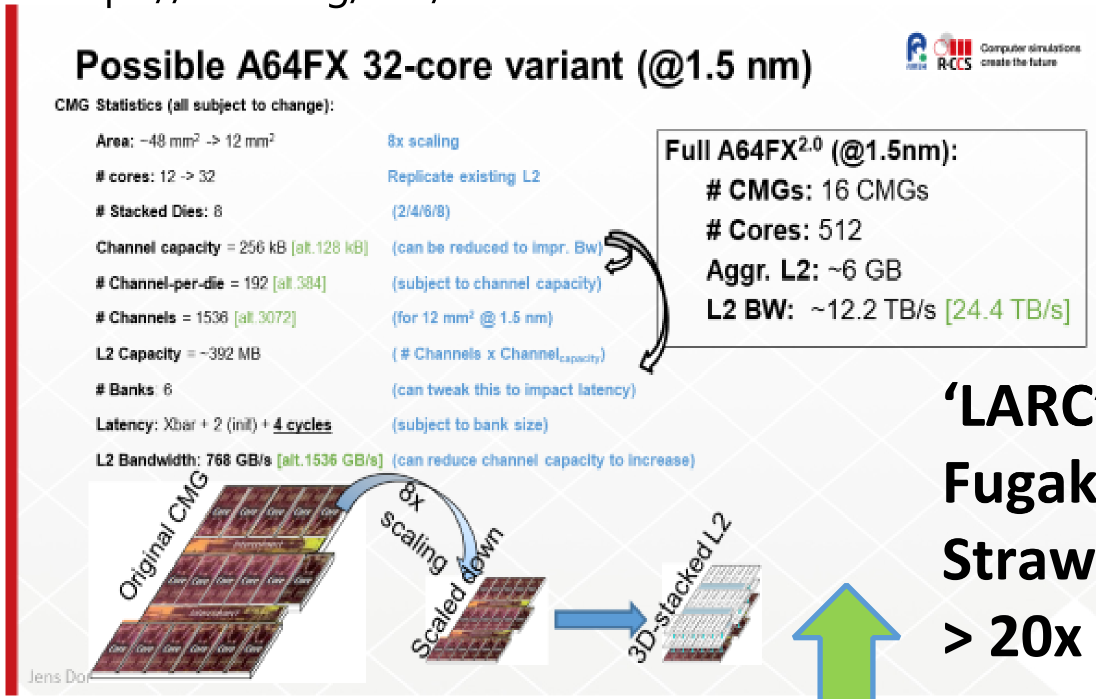
Strong Scaling CGRA

NEDO Project, CPU/PIM for BW bound, Strong Scaling CGRA for compute&latency bound

All is not Rosy: Modernizing & Downselecting Application & Algorithm Types

- **Compute bound via matrix/tensor HW**
  - Fairly low utilization
  - Low memory capacity (O(n^k))
  - Easy to encapsulate in library etc.
- **Latency bound via standard localization & hiding techniques**
  - Good single thread / low latency communication HW
  - Multithreading/latency hiding
  - Latency-avoiding / localization algorithms

Domke et. al. "At the Locus of Performance: A Case Study in Enhancing CPUs with Copious 3D-Stacked Cache" https://arxiv.org/abs/2204.02235



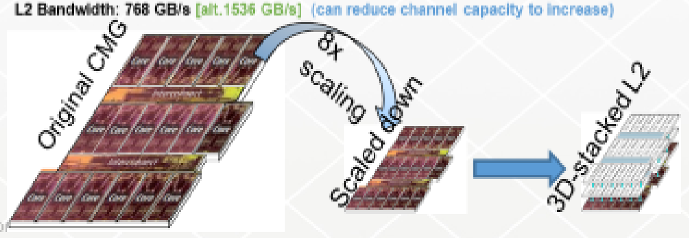
Possible A64FX 32-core variant (@1.5 nm)

Computer simulations create the future

CMG Statistics (all subject to change):

Area: ~48 mm² -> 12 mm²	8x scaling
# cores: 12 -> 32	Replicate existing L2
# Stacked Dies: 8	(2/4/6/8)
Channel capacity = 256 kB [alt.128 kB]	(can be reduced to impr. Bw)
# Channel-per-die = 192 [alt.384]	(subject to channel capacity)
# Channels = 1536 [alt.3072]	(for 12 mm² @ 1.5 nm)
L2 Capacity = ~392 MB	(# Channels x Channel$_{capacity}$)
# Banks: 6	(can tweak this to impact latency)
Latency: Xbar + 2 (init) + 4 cycles	(subject to bank size)
L2 Bandwidth: 768 GB/s [alt.1536 GB/s]	(can reduce channel capacity to increase)

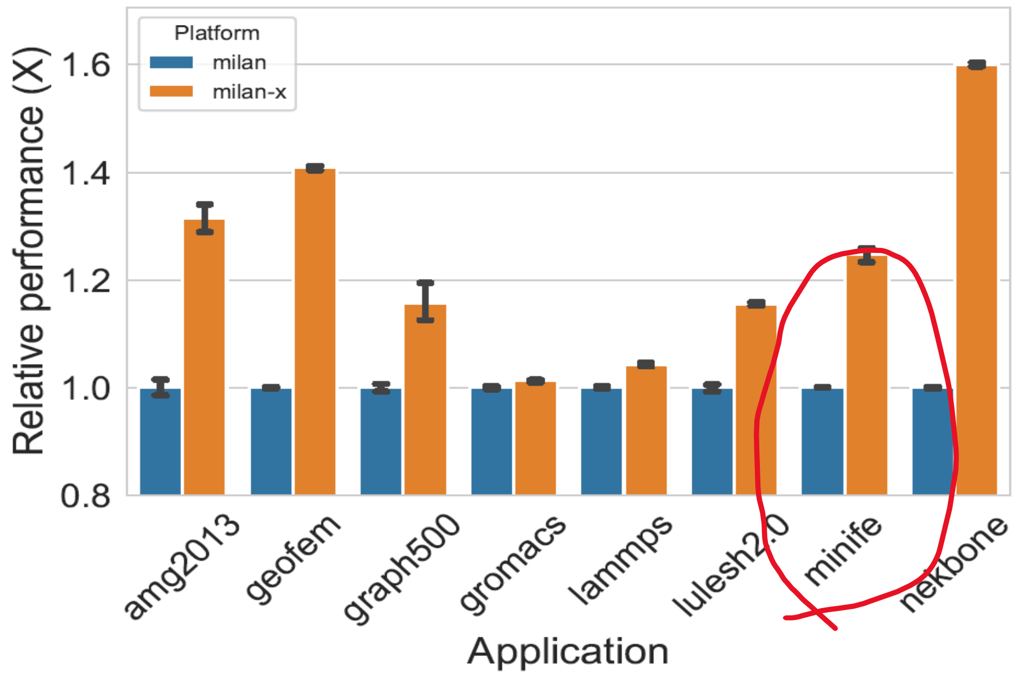
Full A64FX$^{2.0}$ (@1.5nm):
- # CMGs: 16 CMGs
- # Cores: 512
- Aggr. L2: ~6 GB
- L2 BW: ~12.2 TB/s [24.4 TB/s]

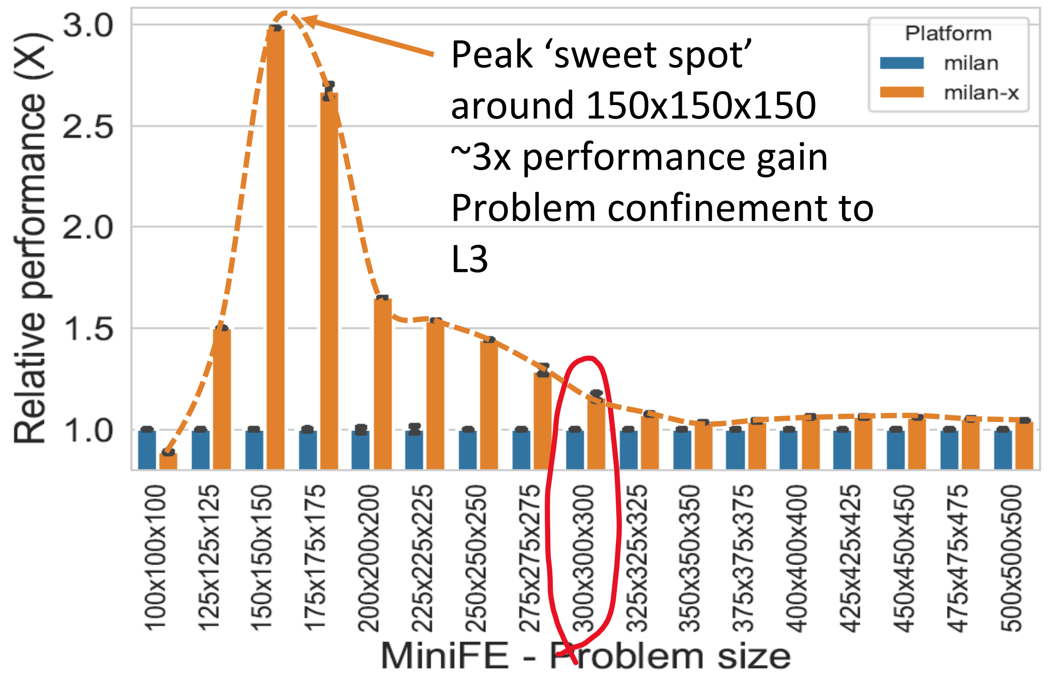


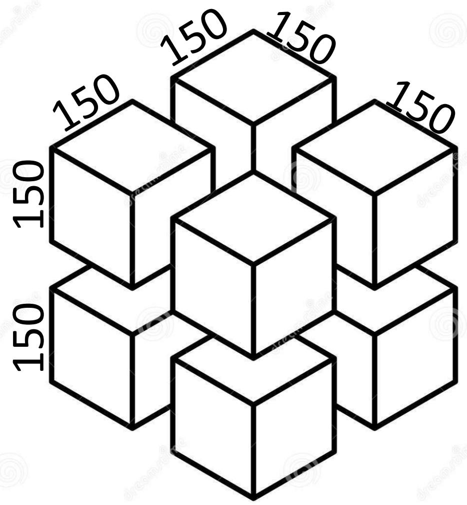
'LARC' CPU FugakuNEXT Strawman > 20x BW

- **BW bound via 3D stacked near memory & photonics**
  - Tiered memory, extreme high BW memory is capacity limited c.f. FLOPS (see figure)
  - Require algorithmic changes and innovations, generic (eg temporal blocking), customized, …
  - *Some apps/algorithms may not survive the change (eg traditional unstructured mesh…)*

LARC: Milan-X (large 768MB on-chip L3) experiment: early proxy for FugakuNEXT main CPU



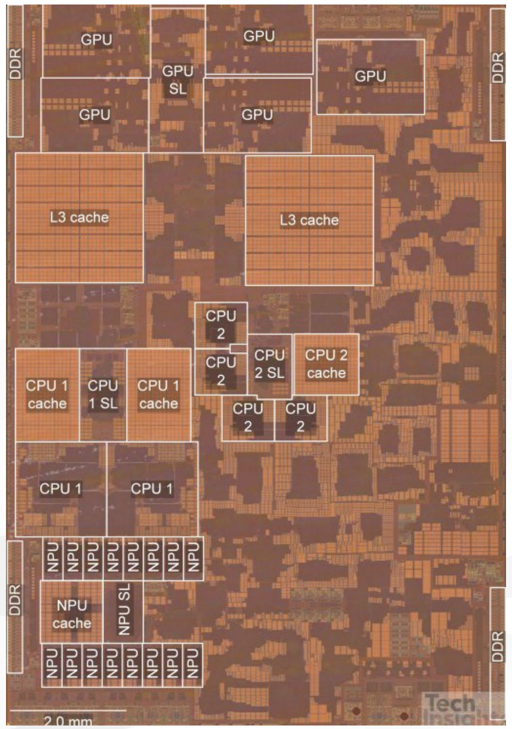




- **Performance gain over 300x300x300**
  - 3x by confining to enlarged L3
  - 8x by core parallelism with scaling => **total 24x speedup**
  - *Caveat: assuming algorithmic strong scaling and process/packaging scaling*

Smartphones *NOT* extrapolatable to HPC

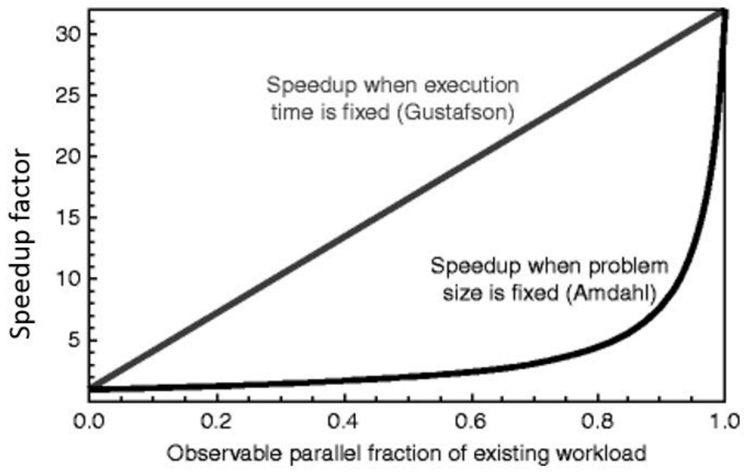
- **SmartPhone SOC subject to Amdahl Speedup (Law)**
- **Supercomputers subject to Amdahl & Gustafson Speedup**



Apple A15 SoC
(source https://semianalysis.com/apple-a15-die-shot-and-annotation-ip-block-area-analysis/)

Gustafson's Law

Instead of running the same size problem for all N, we can also consider running larger problems with better code or greater resources, which leads to Gustafson's law



1/13/2017 ECE 695, Prof. Bermel 10

ECE 695NS Lecture 3: Practical Assessment of Code Performance
by: Peter Bermel, Harvard University
https://nanohub.org/resources/20560/watch?resid=25763

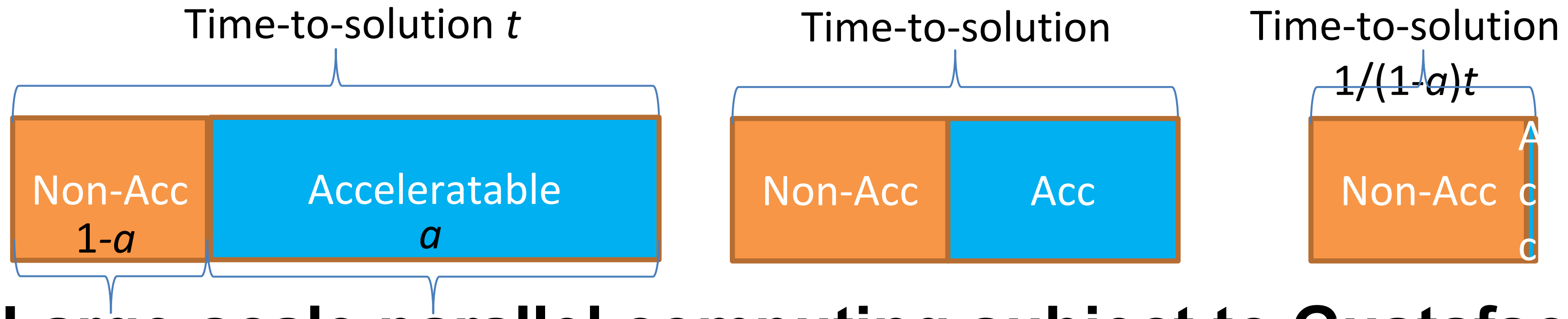
Gustafson J.L. (2011) Gustafson's Law. In: Padua D. (eds) Encyclopedia of Parallel Computing. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-09766-4_78

Reality of Accelerated Computing

- **From the user's point of view, computing system should be uniform, with heterogeneity, distribution etc. hidden under the hood**
  - Success of clouds achieved with this principle
- **Modern IT involves massive software ecosystem, heterogeneity hinders their use => integration with CPU(orGPU) most sensible**
  - Fugaku / A64FX was designed exactly with this principle
- **Performance always governed by Amdahl's law (strong scaling) and Gustafson's law (weak scaling)**
  - Employing multiple heterogeneous accelerators in an app => bad idea
  - "Homogeneous" parallelization of workloads exclusively confined to a SINGLE accelerator type (or CPU) per each node with good load balancing is the ONLY way to overcome the Amdahl's law
  - Successful applications on large GPU machines follow this principle
    - "Balanced" use of GPU and CPU a myth => EITHER GPU or CPU

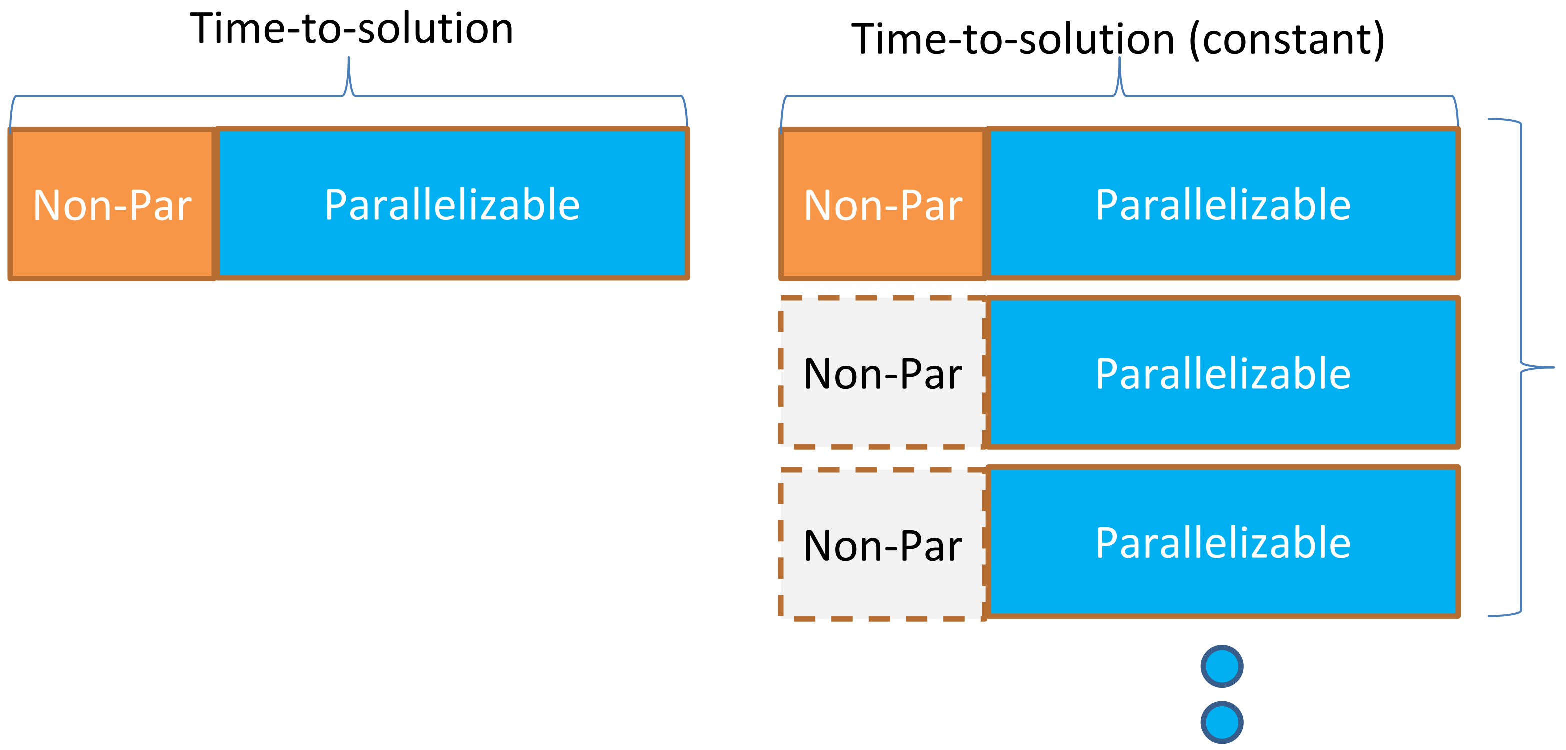
Accelerators vs. Amdahl's Law & Gustafson's Law (1)

- **Accelerators are subject to Amdahl's law (strong scaling)**



For accelerators to work, non-accelerated portion must be as small as possible e.g. GPU-CPU, CPU processing must be minimized

- **Large-scale parallel computing subject to Gustafson's law (weak scaling)**

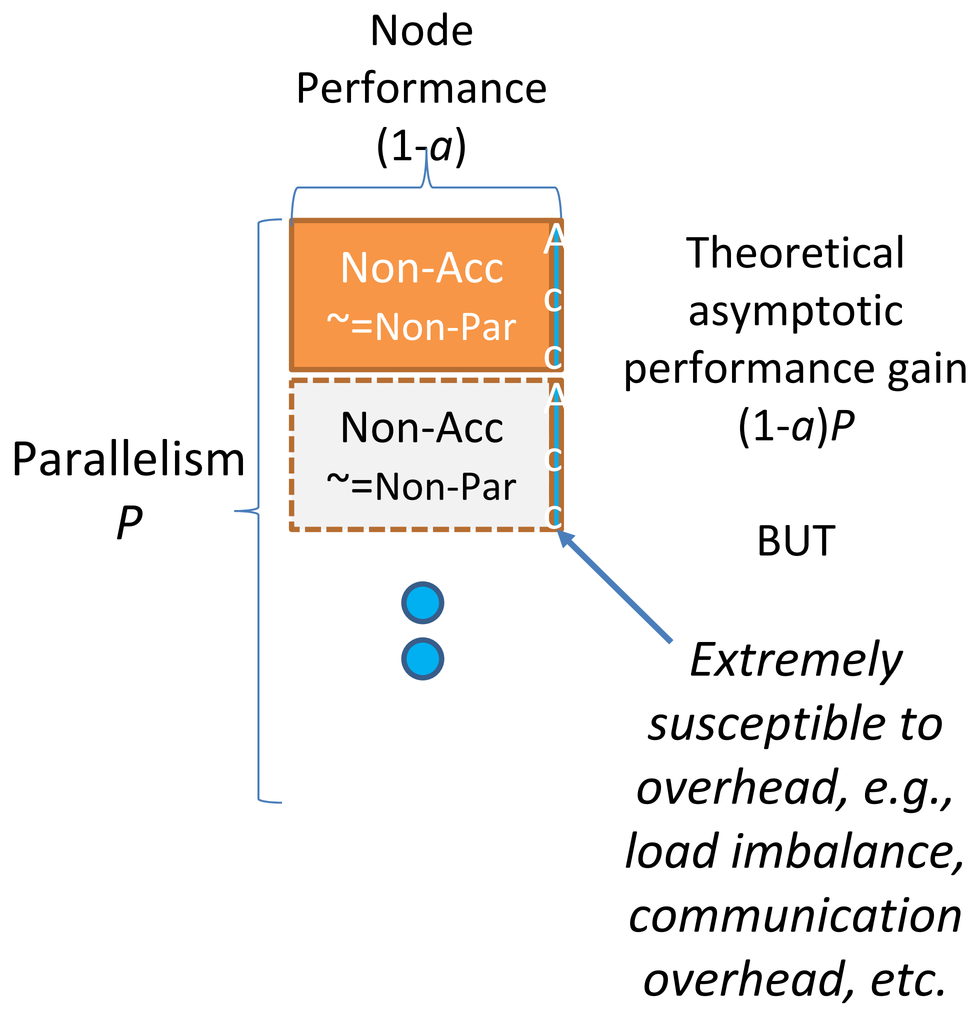


Parallelism: P
Problem size: xP
(weak scaling)

Performance: ~= xP for large P, if non-parallelizable overhead would be minimized, e.g., w/load balancing, communication minimization, etc. => *entails uniform, well balanced processing for every node*

Accelerators vs. Amdahl's Law & Gustafson's Law (2)

● Combining Amdahl's law and Gustafson's law in a supercomputer


Node Performance (1-*a*)
Non-Acc ~=Non-Par
ACC
Non-Acc ~=Non-Par
ACC
Parallelism *P*
Theoretical asymptotic performance gain (1-*a*)*P*
BUT
Extremely susceptible to overhead, e.g., load imbalance, communication overhead, etc.

Principles of Accelerated Supercomputer:

- *Maximizing acceleration under Amdahl*
  => Dominant processing done on the same accelerator on every node
  BAD:"intra-node" heterogeneous processing
- *Extremely uniform load balancing*
  => SPMD over uniform accelerators the best
  BAD: heterogeneous task parallelism over multiple types of accelerators
- *Minimize parallelization overhead e.g. communication*
  => tight communication coupling of accelerated components, on-chip > on-package > on node > different machines
  BAD: any segregation entailing data movement, poor interconnect, etc.

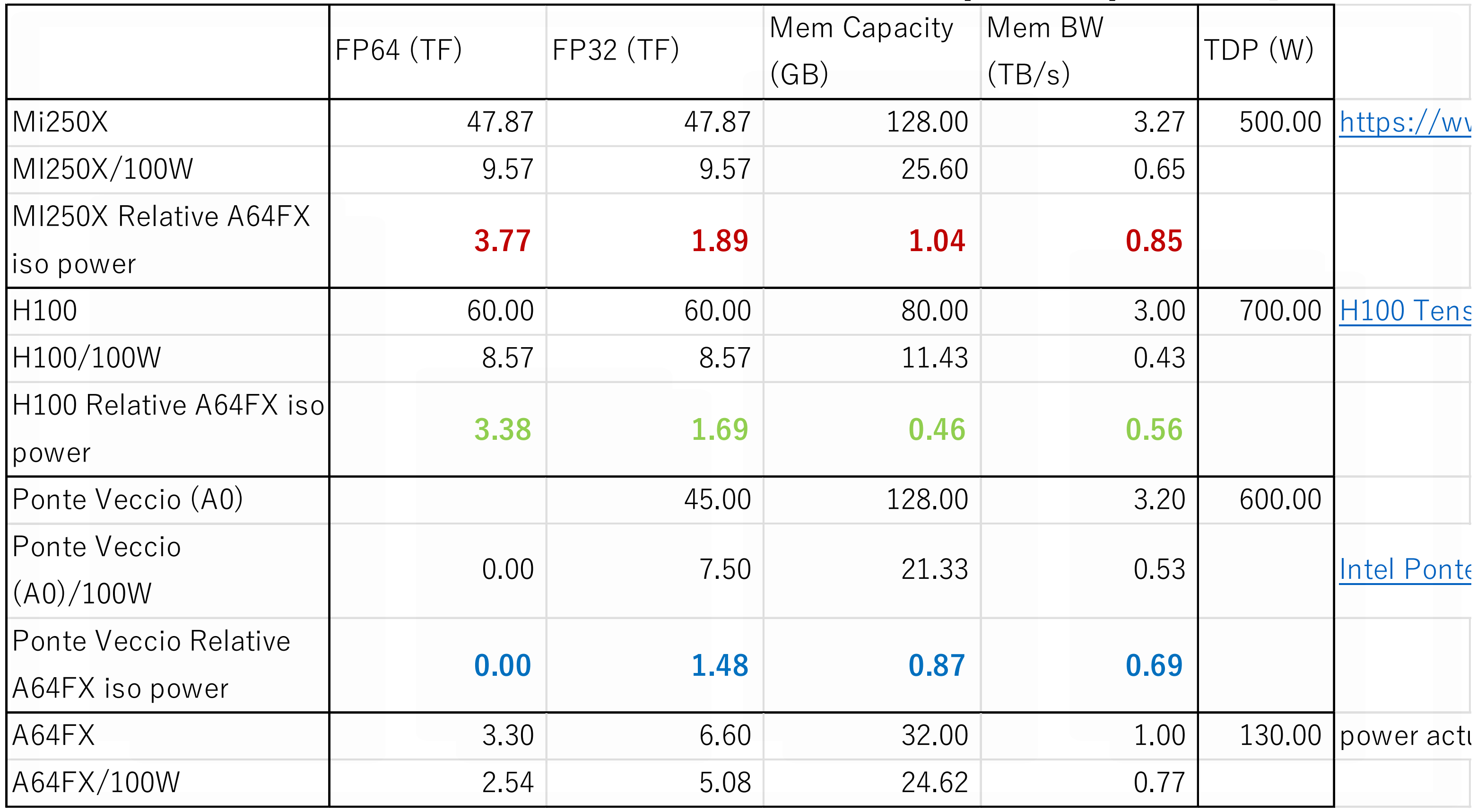
Accelerators vs. Amdahl's Law & Gustafson's Law (3)

- **It is no accident that, every successful large-scale accelerated supercomputers (esp. GPU machines) are**
  - built with a singular node configuration across the entire machine
  - tight coupling and robust interconnect (& I/O) to sustain maximum bandwidth in/out of accelerator processor
  - dominant processing on the GPU for maximum performance
  - SPMD with very good load balancing (incl. data parallel DNN training)
  - Tsubame2/3, Tianhe-2A, Titan/Summit, Piz-Daint, ABCI, Fugaku, Frontier, Lumi, Aurora, …

  - … and this is the consequence of physical laws, so will continue to be applicable to future machines (no extreme heterogeneity, asynchrony, …)

Current GPUs nor their trajectory are not so promising…

- **Top-end HPC/AI GPUs Circa 2022-23 relative to A64FX (2019), iso power**
  - Modest ↑ FP32
  - Flat~modest ⤵ Mem capacity
  - Modest ⤵ Mem BW
  - Not much gain for majority of HPC / digital twin apps



	FP64 (TF)	FP32 (TF)	Mem Capacity (GB)	Mem BW (TB/s)	TDP (W)	
Mi250X	47.87	47.87	128.00	3.27	500.00	https://wv
MI250X/100W	9.57	9.57	25.60	0.65		
MI250X Relative A64FX iso power	**3.77**	**1.89**	**1.04**	**0.85**		
H100	60.00	60.00	80.00	3.00	700.00	H100 Tens
H100/100W	8.57	8.57	11.43	0.43		
H100 Relative A64FX iso power	**3.38**	**1.69**	**0.46**	**0.56**		
Ponte Veccio (A0)		45.00	128.00	3.20	600.00	
Ponte Veccio (A0)/100W	0.00	7.50	21.33	0.53		Intel Ponte
Ponte Veccio Relative A64FX iso power	**0.00**	**1.48**	**0.87**	**0.69**		
A64FX	3.30	6.60	32.00	1.00	130.00	power actu
A64FX/100W	2.54	5.08	24.62	0.77		

- Compare Fugaku (160K A64FX @ 20MW) vs. Frontier (40K Mi250X + 20K CPU @ 30MW)
- **3 years after A64FX/Fugaku, GPU-based US Exascale machines will be fantastic in AI/DL, modest gain in HPC compute bound apps (FP32/FP64 mixed), no gain or less performant in BW bound apps (subject to verification in various benchmarks)**

"Multiple Heterogeneous Domain Specific Accelerator" Considered Harmful

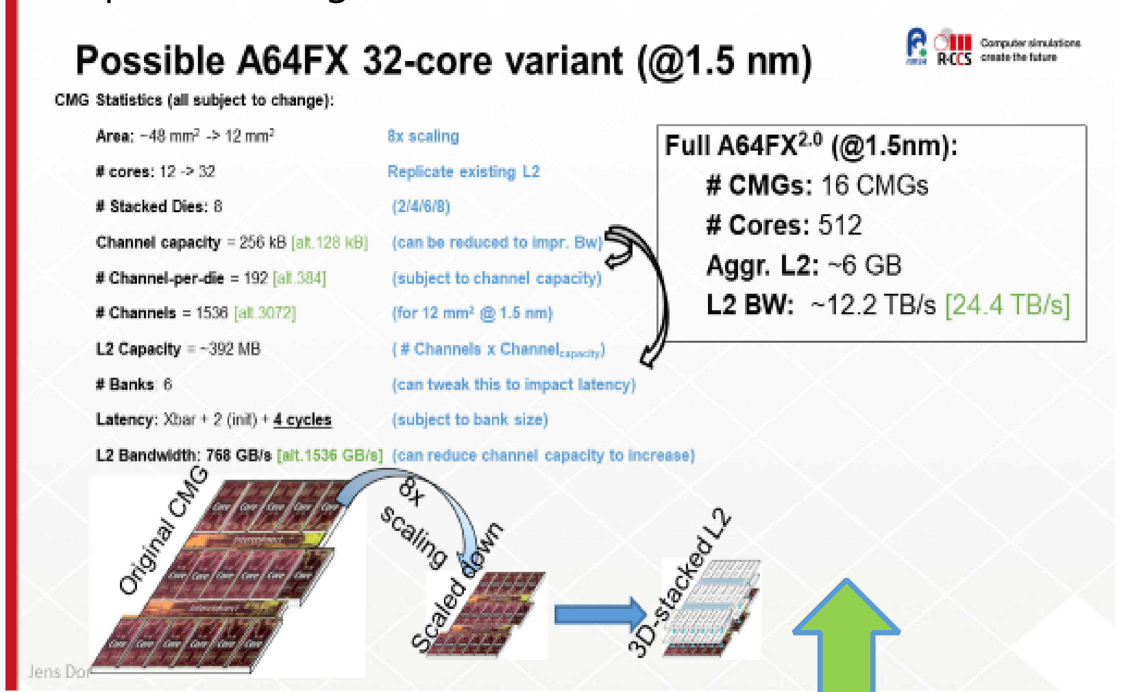
- Amahl's law also will hit communication time and energy/power consumption
- Even if we achieve considerable speedup with low energy on the accelerator, moving the data around to be processed by other accelerators will be hit with the Amdahl's law in communication time and power/energy consumption
  - Neither can be brought down, the more distance the signal travels from on-chip towards inter-rack or inter IDC, becoming the overall overhead factor
- Thus the right approach to minimize the effect of the Amdahl's law is to do SoC or even CPU integration of acceleration features, NOT PLETHORA OF DOMAIN-SPECIFC HETEROGENEOUS ACCELERATOR CHIPS&SYSTEMS
  - Again, Fugaku / A64FX was designed with this principle

- **<u>Accelerator should focus on strong scaling (in fact whole machine)</u>**

All is not Rosy: Modernizing & Downselecting Application & Algorithm Types

- **Compute bound via matrix/tensor HW**
  - Fairly low utilization
  - Low memory capacity (O(n^k))
  - Easy to encapsulate in library etc.
- **Latency bound via standard localization & hiding techniques**
  - Good single thread / low latency communication HW
  - Multithreading/latency hiding
  - Latency-avoiding / localization algorithms

Domke et. al. "At the Locus of Performance: A Case Study in Enhancing CPUs with Copious 3D-Stacked Cache" https://arxiv.org/abs/2204.02235



Possible A64FX 32-core variant (@1.5 nm)

Computer simulations create the future

CMG Statistics (all subject to change):

Area: ~48 mm² -> 12 mm²	8x scaling
# cores: 12 -> 32	Replicate existing L2
# Stacked Dies: 8	(2/4/6/8)
Channel capacity = 256 kB [alt.128 kB]	(can be reduced to impr. Bw)
# Channel-per-die = 192 [alt.384]	(subject to channel capacity)
# Channels = 1536 [alt.3072]	(for 12 mm² @ 1.5 nm)
L2 Capacity = ~392 MB	(# Channels x Channel$_{capacity}$)
# Banks: 6	(can tweak this to impact latency)
Latency: Xbar + 2 (init) + 4 cycles	(subject to bank size)
L2 Bandwidth: 768 GB/s [alt.1536 GB/s]	(can reduce channel capacity to increase)

Full A64FX$^{2.0}$ (@1.5nm):
- # CMGs: 16 CMGs
- # Cores: 512
- Aggr. L2: ~6 GB
- L2 BW: ~12.2 TB/s [24.4 TB/s]

- **BW bound via 3D stacked near memory & photonics**
  - Tiered memory, extreme high BW memory is capacity limited c.f. FLOPS (see figure)
  - Require algorithmic changes and innovations, generic (eg temporal blocking), customized, …
  - *Some apps/algorithms may not survive the change (eg traditional unstructured mesh…)*

Investigating the non-Quantum Future FLOPS to BYTES for future acceleration? (1)

- **Increasing FLOPS via increasing the number of ALUs no longer viable**
  - Compute power = ALU logic switching power + data movement between ALUs and registers/memory
  - ALU logic power saturation faster than lithography saturation
    - No more acceleration of pure FLOPS
    - Only way to increase performance at low level is logic simplification, e.g., lower precision, alternative numerical formats
    - At higher levels, decreasing the # of numerical operations very effective => sparse (iterative) methods (general HPC), network compaction (AI), algorithmic pruning (HPC & AI)

Investigating the non-Quantum Future FLOPS to BYTES for future acceleration? (2)

- **Data movement has its own problems but promising w/ new device and packaging tech + architectures & algorithms to exploit them**
  - Devices & Packaging
    - 3-D stacking of memory + logic
    - Photonic interconnect
    - Dense and fast memory devices from SRAM to MRAM
  - Architecture
    - Large & high bandwidth local memory processor (very large L1/L2)
    - Customized datapaths for frequent compute patters – stencils/convolution, matrix, FFT, tensor operations, ... => can they be generalized? Micro dataflow in a core?
    - Coarse grained dataflow (CGRA)? => optimize data movement in general over standard CPU/GPU(SMT Vector)
    - Near memory processing
- **FLOPS to BYTES!**
  - Same motivation as embedded computing

Our Project: Exploring versatile HPC architecture and system software technologies to achieve 100x performance by 2028

Problems to be solved and goals to be achieved

- General-purpose computer architectures that will accelerate a wide range of applications in the post-Moore era have not yet been established.
- What is a feasible approach for versatile HPC systems based on bandwidth improvement?
- **Goal:** to explore architectures that can achieve 100x performance in a wide range of applications around 2028

Approaches and subtasks

- Exploration of future CPU node architectures and necessary technologies

Subtask1.1 Performance characterization and modeling with benchmarks to identify directions for exploration and improvement (Riken R-CCS)

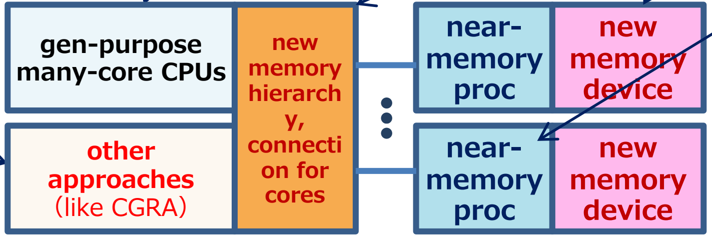
Subtask2 Exploring innovative memory architectures with ultra-deep and ultra-wide bandwidth (Tokyo Tech.)

Subtask1.2 Exploring a reconfigurable vector data-flow architecture (CGRA) that can exploit increased data transfer capability (Rijen R-CCS)

Subtask3 Exploring near-memory computing for highly effective bandwidth and cooling efficiency for general purpose computing (U-Tokyo)

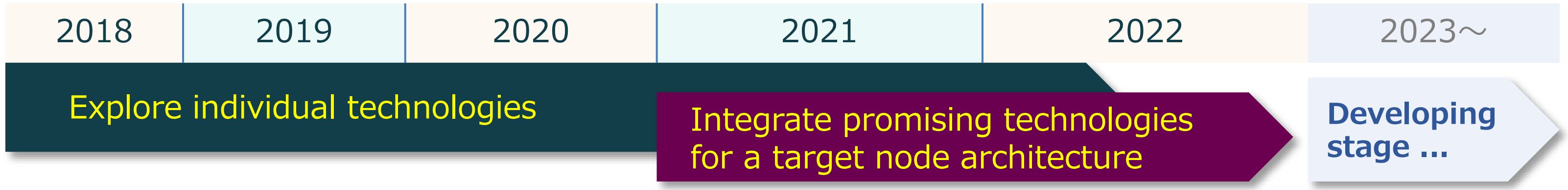
Planned

Subtask4 Exploration of node architectures as extension of existing many-core CPUs with non von-Neumann methods (unnamed company)


gen-purpose many-core CPUs
other approaches (like CGRA)
new memory hierarchy, connection for cores
near-memory proc
new memory device
near-memory proc
new memory device

Exemlar FLOPS to BYTES Architecture

Plan


2018
2019
2020
2021
2022
2023～
Explore individual technologies
Integrate promising technologies for a target node architecture
Developing stage …

Non-Quantum Future Towards Strong Scaling (1)

- **Assume constant memory per core**
- **#cores ~ total problem (total machine (memory)) size *n* ~ core performance**
- **Modern massively parallel architectures: core performance constant, performance gains ~ increasing #cores in system, runtime T ~ problem complexity / core performance**
- **Compute-bound codes, O(n^k) complexity where k > 1 : runtime T ~ n^{k-1} , so increasing total machine size increases *T*, even w/ constant memory per core**
- **Memory-bound codes, O(*n*) complexity, runtime *T* ~ # memory controllers**
  - At core level, # memory controllers (e.g. access to cache) ~ #cores so runtime remains constant with increasing cores (weak scaling).
  - However, at chip level (external memory access), memory controllers are constant even with #cores increase, so T ~ #cores (no scaling)
    - Increasing memory size further per core meaningless, since *T* ~ *n*
- **Maintaining memory size per core, let alone increase, will not lead to effective performance gains, diminishing Gustafson's Law**

Non-Quantum Future Towards Strong Scaling (2)

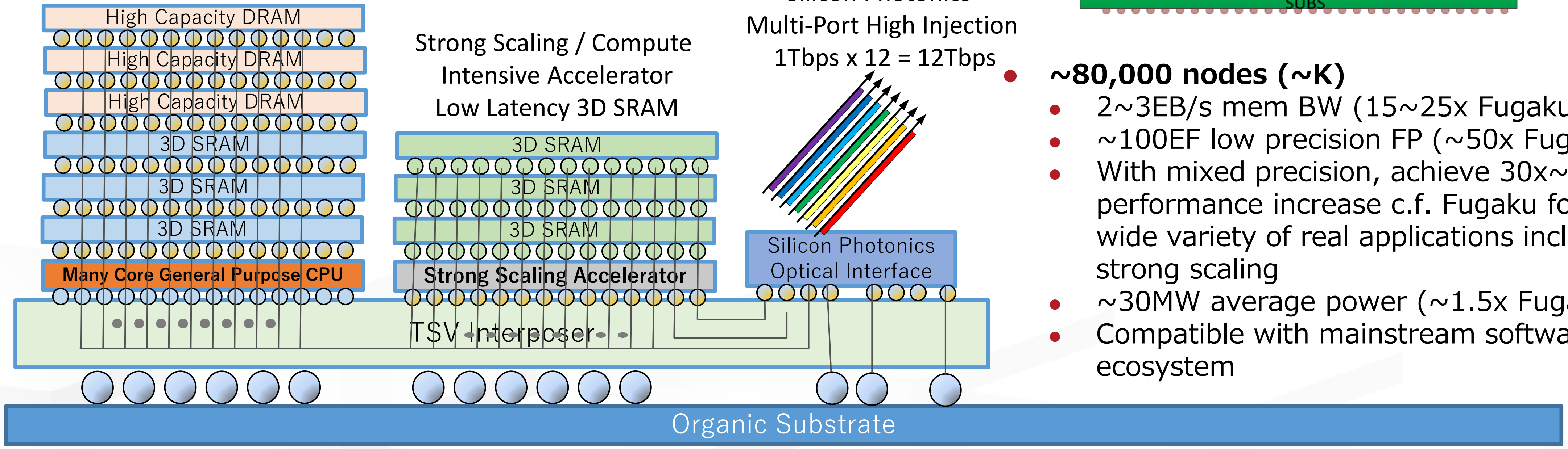
- **Even traditional weak scaling codes will need to strong scale**
  - Architectural requirements: memory high BW / low latency => small capacity
  - **Science requirements: from demonstrative big runs to *real R&D***
    - **Ensemble of multiple smaller problem sizes**
    - **Time to solution >> problem size**
  - **If Gustafson's law is well satisfied (e.g., well load balanced),** then strong scaling will work up to the point of bad load balance and/or non-parallel region becoming significant
- **Some apps inherently strong scaling and may benefit from accelerator**
  - E.g. Molecular Dynamics, c.f., Anton
- **Most apps (esp. BW sensitive) must be prepared to strong scale at algorithms level, or at least deal with hierarchical memory**
  - **Advanced localization e.g. temporal blocking, putting only BW sensitive data in fast memory, memory compression (incl. low rank approximation…)**

2028~30 Strawman Non-Quantum Next-Gen FugakuNEXT Architecture

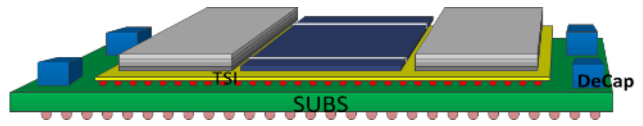
High Bandwidth / High Memory Capacity
General-Purpose Many-Core CPU
High Bandwidth SRAM + Large Capacity DRAM or NVM

Strong Scaling / Compute Intensive Accelerator
Low Latency 3D SRAM

Silicon Photonics
Multi-Port High Injection
1Tbps x 12 = 12Tbps



1.5 nm UV fabrication



- **~80,000 nodes (~K)**
  - 2~3EB/s mem BW (15~25x Fugaku)
  - ~100EF low precision FP (~50x Fugaku)
  - With mixed precision, achieve 30x~100x performance increase c.f. Fugaku for wide variety of real applications including strong scaling
  - ~30MW average power (~1.5x Fugaku)
  - Compatible with mainstream software ecosystem

- General purpose CPU w/3D Stack memory for high bandwidth apps, >20TB/s SRAM bandwidth, FP64/FP32 Scalable with multiple tiled architecture (could be 40TB/s)
- CGRA accelerator w/high compute intensity for strong scaling apps + compute intensive apps + Deep Learning FP32/19/16 > 1PF per node, very low latency configuration of compute pipelines for MD, DL Inference, etc. for strong scaling

- Direct Chip-Chip Interconnect with DWDM Silicon Photonics
- Low arity switches for multi-dimensional torus, multi-channel network injection ports

Observations for NextGen CPU

- **Similar result using large L3 obtained by Ltaief et. al. [SC21] (see below)**
- **For majority of codes memory bandwidth bound => dramatic increase in performance by large capacity L2/L3 dedicated to core(s) via 3D, then increasing core count & SRAM capacity with lithography shrink**
- **Much R&D needed to fit existing codes into this model (semi-) automatically**
  - HW support for strong scaling => low latency intra-chip NW, fast messaging,
  - Various algorithms, compilers & libraries & frameworks & tools etc. support to 'fit' problems into smaller memory, including:
    - Data compression incl. low rank approximation [SC21]
    - Hierarchical data partitioning/restructuring, to cluster BW sensitive data onto faster memory
    - Latency hiding incl. temporal blocking over hierarchical memory
    - Load balancing to maintain Gustafson's law
  - Ultimately, may require changes in the underlying numerics/solvers in the apps
    - But once done the code will be future proof

[SC21] Hatem Ltaief, Jesse Cranney, Damien Gratadour, Yuxi Hong, Laurent Gatineau and David Keyes, "Meeting the Real-Time Challenges of Ground-Based Telescopes Using Low-Rank Matrix Computations", ACM/IEEE Supercomputing 21, the ACM Press, Nov. 2021.

What should a strong scaling accelerator look like?

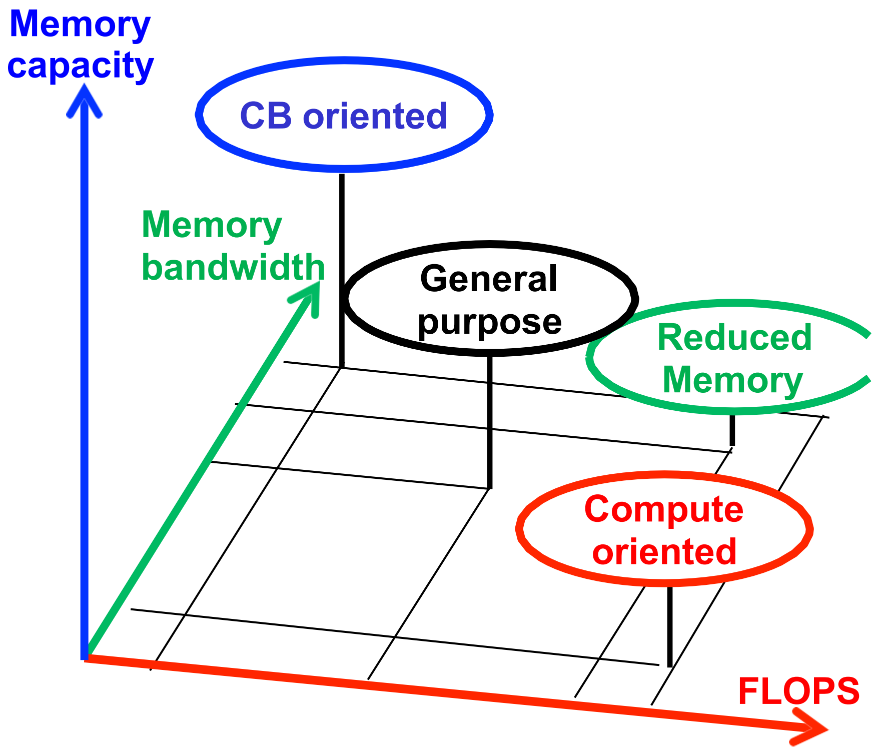
- **Properties**
  - Configurable datapaths that synchronize at clock level
  - Large SFU blocks aka CGRA---low precision matrix engines, FFTs, various DL operators, …
  - Compute intensive SFUs must be ‘densely’ packed to compete in per chip performance with weak scaling chips when it is used in weak scaling mode (e.g., large scale MM in CNN)
- **Some Candidates**
  - Commercial CGRA e.g., Xylinx ACAP
  - High performance dataflow/CGRA in research e.g., Intel CSA
  - GPUs with clock-level synchronization (c.f., atomics)
  - Outgrowth of FPGA w/very large SFUs

Backup Slides

SDHPC (2011-2012) Candidate of ExaScale Architecture

https://www.exascale.org/mediawiki/images/a/aa/Talk-3-kondo.pdf

- Four types of architectures are considered
  - General Purpose (GP)
    - Ordinary CPU-based MPPs
    - e.g.) K-Computer, GPU, Blue Gene, x86-based PC-clusters
  - Capacity-Bandwidth oriented (CB)
    - With expensive memory-I/F rather than computing capability
    - e.g.) Vector machines
  - Reduced Memory (RM)
    - With embedded (main) memory
    - e.g.) SoC, MD-GRAPE4, Anton
  - Compute Oriented (CO)
    - Many processing units
    - e.g.) ClearSpeed, GRAPE-DR



SDHPC (2011-2012) Performance Projection

- Performance projection for an HPC system in 2018
  - Achieved through continuous technology development
  - Constraints: 20 – 30MW electricity & 2000sqm space

Node Performance	Total CPU Performance (PetaFLOPS)	Total Memory Bandwidth (PetaByte/s)	Total Memory Capacity (PetaByte)	Byte / Flop
General Purpose	200~400	20~40	20~40	0.1
Capacity-BW Oriented	50~100	50~100	50~100	1.0
Reduced Memory	500~1000	250~500	0.1~0.2	0.5
Compute Oriented	1000~2000	5~10	5~10	0.005

Network

	Injection	P-to-P	Bisection	Min Latency	Max Latency
High-radix (Dragonfly)	32 GB/s	32 GB/s	2.0 PB/s	200 ns	1000 ns
Low-radix (4D Torus)	128 GB/s	16 GB/s	0.13 PB/s	100 ns	5000 ns

Storage

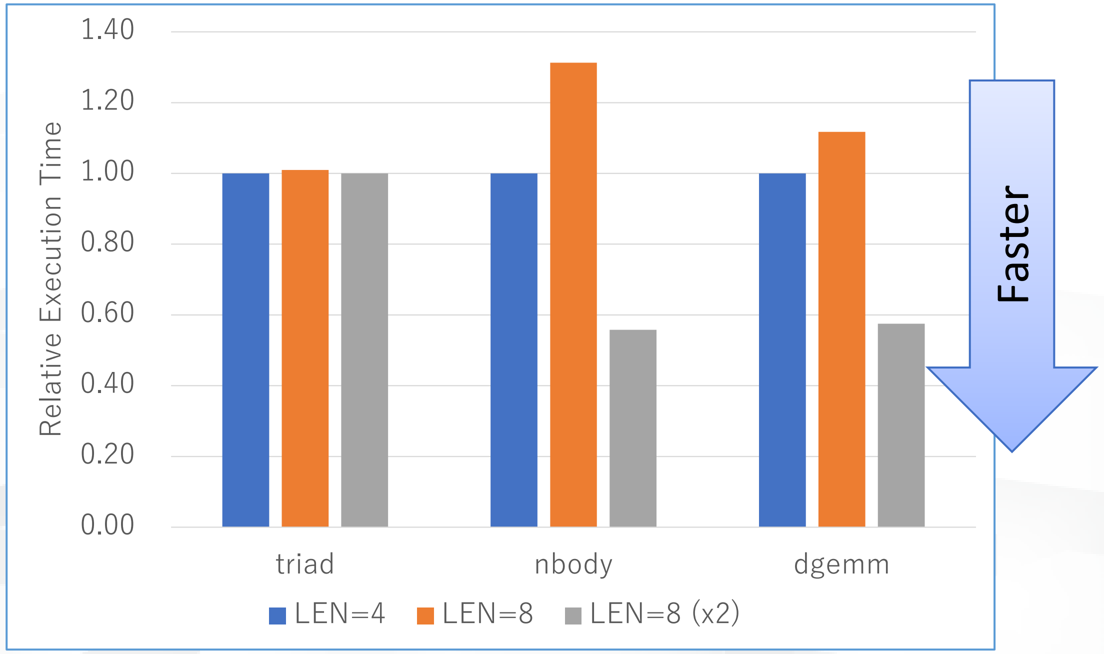
Total Capacity	Total Bandwidth
1 EB	10TB/s
100 times larger than main memory	For saving all data in memory to disks within 1000-sec.

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



Technologies and Architectural Parameters to be determined by Codesign

- **Basic Architecture Design (by Feasibility Studies)**
  - Manycore approach, O3 cores, some parameters on chip configuration and SIMD
- **Instruction Set Architecture and SIMD Instructions**
  - Fujitsu collaborated with Arm, contributing to the design of the SVE as a lead partner
- **Chip configuration**
- **Memory technology**
  - DDR, HBM, HMC
- **Cache structure**
- **Out of order (O3**
- **Enhancement fo**
- **Interconnect between Nodes**
  - SerDes, topologies "Tofu" or other network?

✓ The number of cores in a CMG
✓ The number of CMGs in a chip
s to shared L2 in a CMG
, the size, and throughp
vork-on-chip to connect
✓ The die size of the chip
✓ The number of chips in a node

SC20 technical paper. "Co-Design for A64FX Manycore Processor and "Fugaku""

M. Sato, Y. Ishikawa, H. Tomita, Y. Kodama, T. Odajima, M. Tsuji, H. Yashiro, M. Aoki, N. Shida, I. Miyoshi,K. Hirai, A. Furuya, A. Asato, K. Morita, T. Shimizu

Lessons Learned from Fugaku

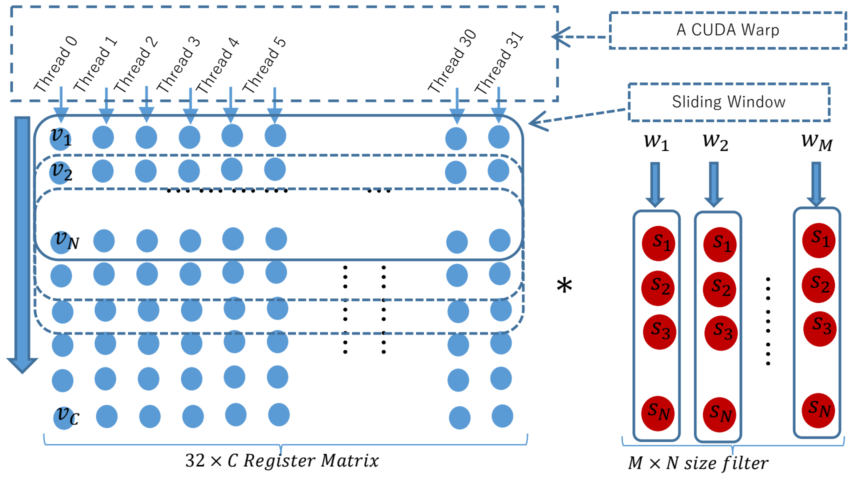
- **Positives: proper project vision and management**
  - **<u>General purpose</u>** low power CPU w/good FLOPs and high BW
    - Arm ecosystem extremely important, for programming, tools, & apps
  - **Aggressive R&D+adoption of new (risky) technologies:** on-die HBM2, Embedded ~400Gbps partially optical switchless interconnect, mainframe RAS, low power etc.
  - **Co-design** and co-working at (inter-)nationally
    - Some evolutions to cope with massive parallelism (K=>Fugaku)
    - Addition of modern architecture features e.g. FP16
- **Shortcomings: lack of widespread commercial adoption**
  - Co-design: focused too much on target app optimization (only)
  - Immaturity of software stack esp. compilers & libraries w/SVE
  - Still too focused on classic HPC for industry & cloud adoption
  - Failed to look at modern apps: data, AI, entertainment, mobility
  - Failed to 'deprecate' classes of algorithms towards Post-Moore
- **What elements can we learn for sustained perf. improvements**

GPUs do have some internal clock-level synchronization: "Pushing the Limits for 2D Convolution Computation On GPUs"

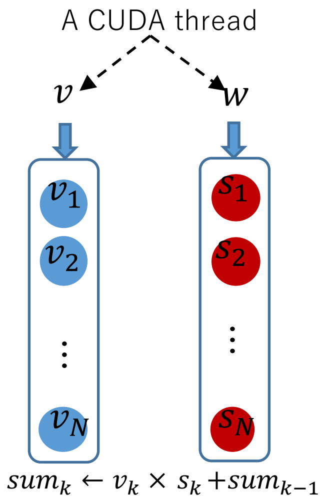
[Chen et. al., ACM/IEEE SC19]

Concept adopted fully by Intel Xe GPU OneAPI

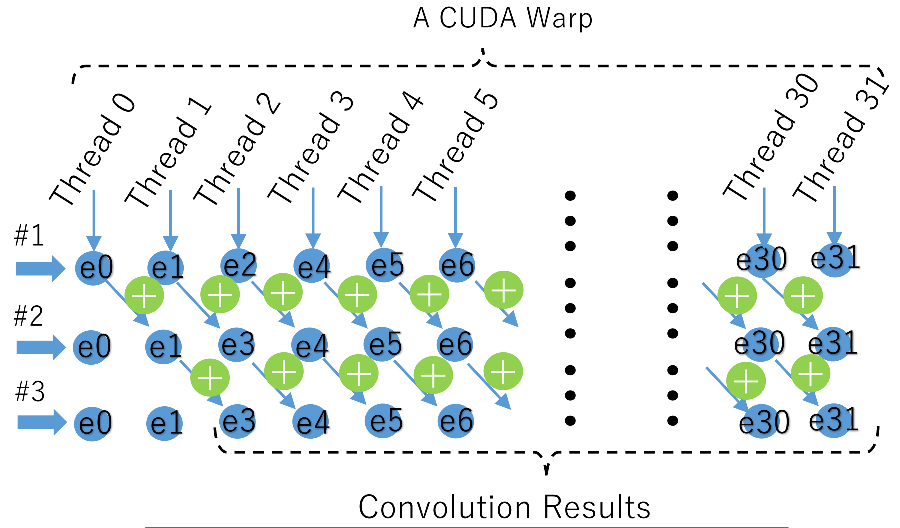
- **Background of 2D convolution**
  - Convolution on CUDA-enabled GPUs is essential for Deep Learning workload
  - A typical memory-bound problem with regular access
- **Method**



(1) Register Cache



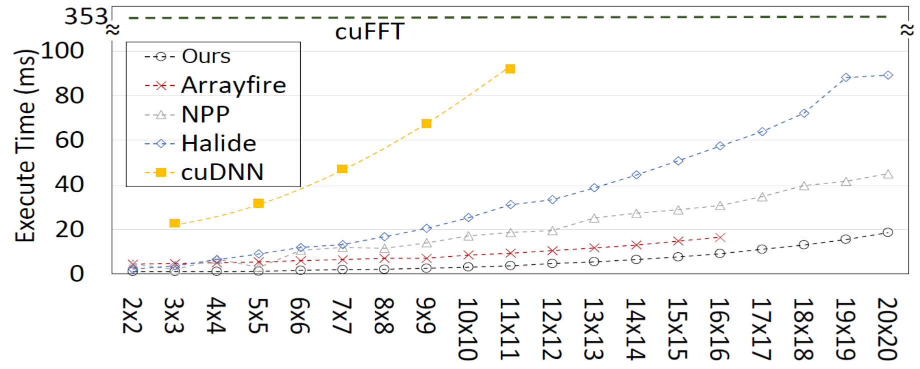
(2) Compute partial sums



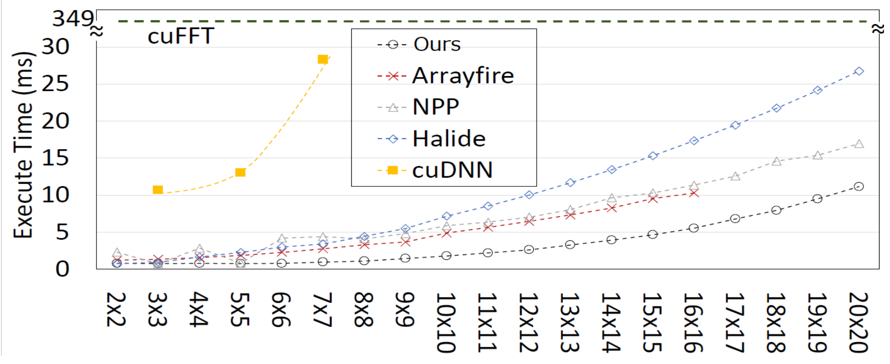
(3) Transfer partial sums

- **Evaluation**
  - a single Tesla **P100** and **V100** GPUs
  - Single precision

Point: Vector lane shuffle datapath can fully emulate Systolic Array efficiently



Evaluation on Tesla **P100** GPU

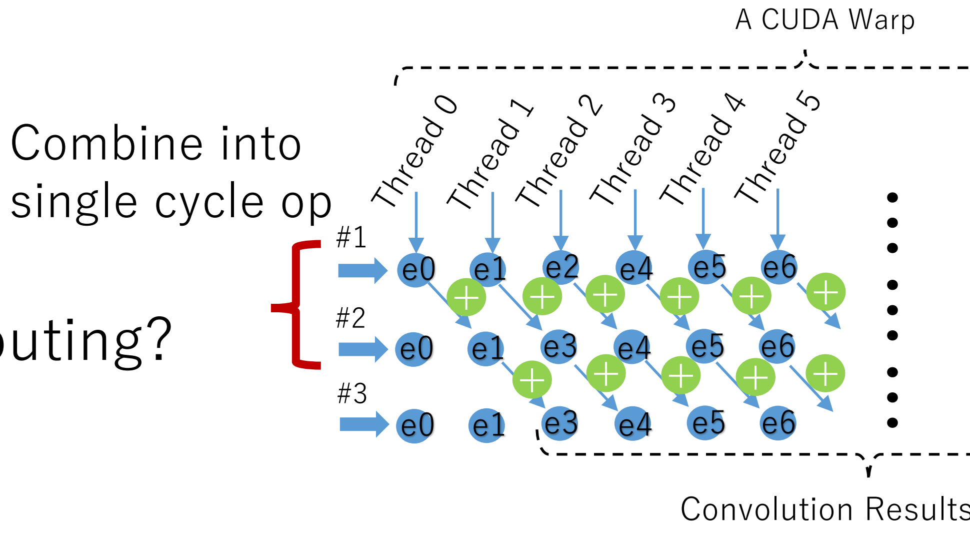


Evaluation on Tesla **V100** GPU

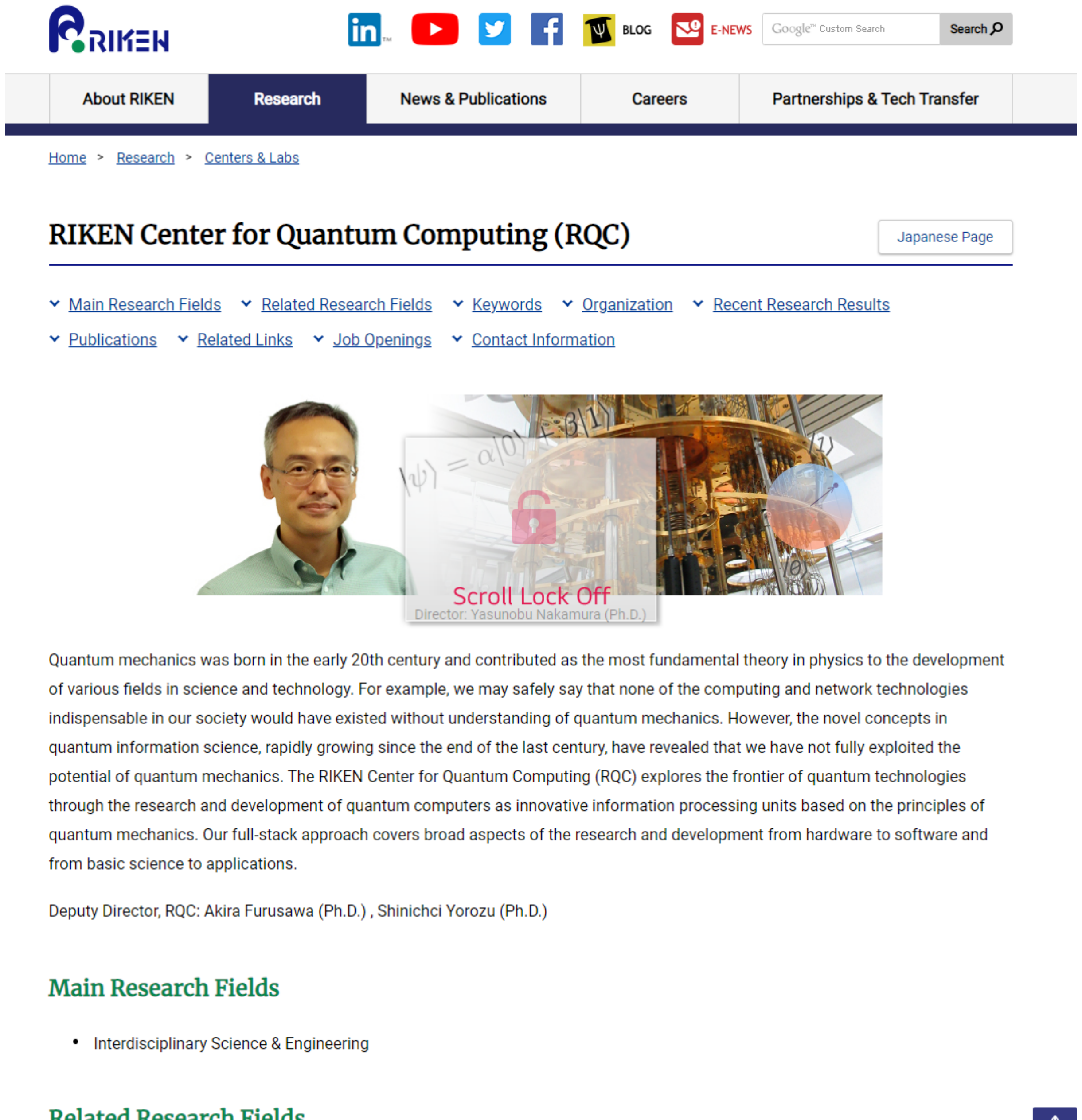
[1] Peng Chen, Mohamed Wahib, Shinichiro Takizawa, Satoshi Matsuoka. Pushing the Limits for 2D Convolution Computation On CUDA-enabled GPUs

Lessons learned from the SSA work

- Existing vector CPUs already embed internal datapaths to emulate SA ops efficiently, with clock-level synchronization
  - Vector lane (Warp) shuffle
  - Note that it does not increase FLOPS as # of ALUs are x1 or x2 vector lanes => speedup due to data movement optimization and clock level synchronization leading to strong scaling.
- Questions
  - Are there ways to maintain the data movement advantage and increasing FLOPS? (increase # ALU with datapaths), consistent with major compute patterns?
  - Are there other datapaths for other major compute patterns? (MM, FFT, DL, etc.)
  - What are the silicon tradeoffs for datapaths? => are they worth the cost for the overall application portfolio
  - Can strong scaling be extended to inter-core computing? (not just atomics)



Investigating the Quantum Future with HPC



Home > Research > Centers & Labs

RIKEN Center for Quantum Computing (RQC)

Japanese Page

Main Research Fields · Related Research Fields · Keywords · Organization · Recent Research Results · Publications · Related Links · Job Openings · Contact Information

Director: Yasunobu Nakamura (Ph.D.)

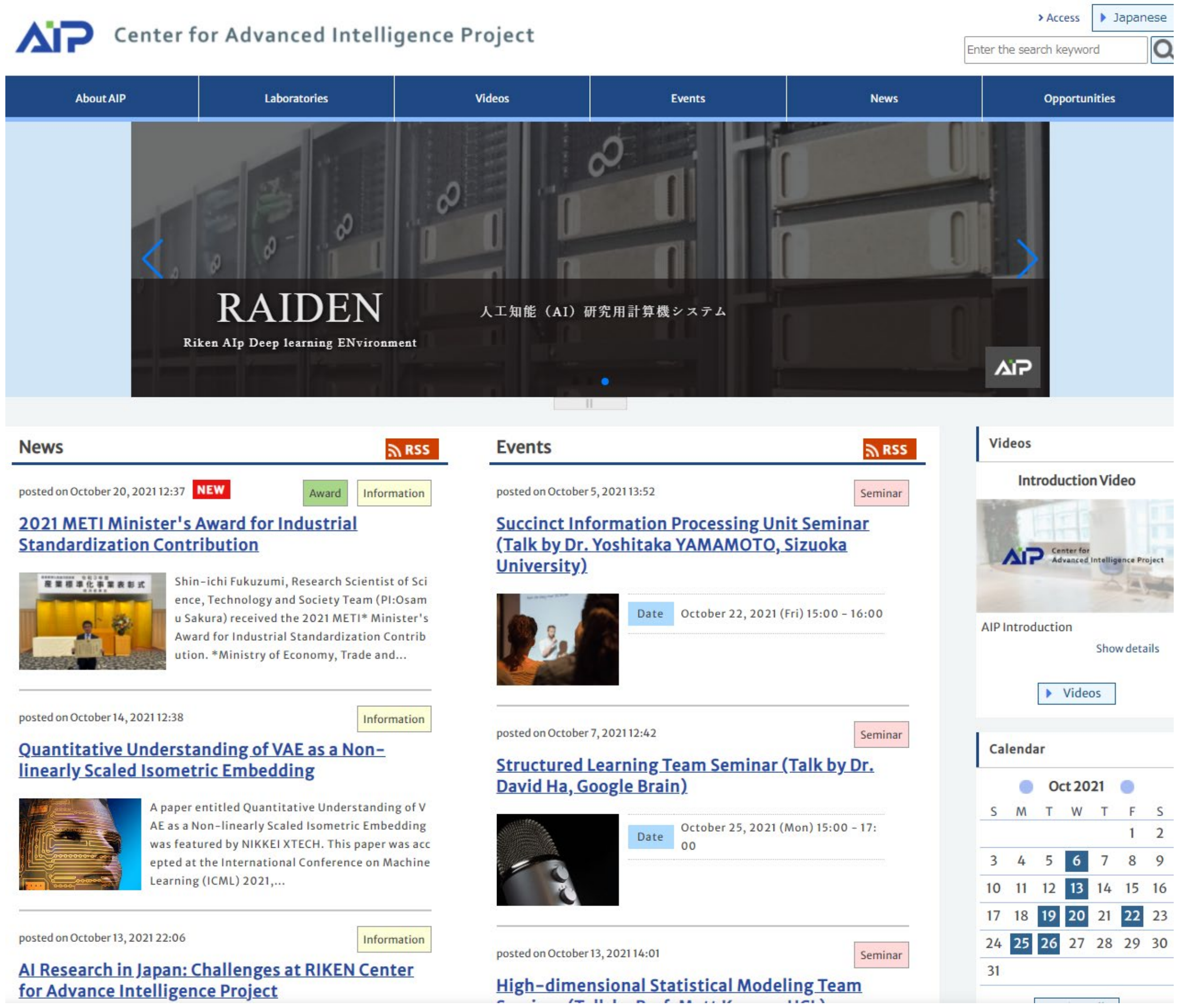
Quantum mechanics was born in the early 20th century and contributed as the most fundamental theory in physics to the development of various fields in science and technology. For example, we may safely say that none of the computing and network technologies indispensable in our society would have existed without understanding of quantum mechanics. However, the novel concepts in quantum information science, rapidly growing since the end of the last century, have revealed that we have not fully exploited the potential of quantum mechanics. The RIKEN Center for Quantum Computing (RQC) explores the frontier of quantum technologies through the research and development of quantum computers as innovative information processing units based on the principles of quantum mechanics. Our full-stack approach covers broad aspects of the research and development from hardware to software and from basic science to applications.

Deputy Director, RQC: Akira Furusawa (Ph.D.) , Shinichci Yorozu (Ph.D.)

Main Research Fields

- Interdisciplinary Science & Engineering

Related Research Fields



Center for Advanced Intelligence Project
Access
Japanese
Enter the search keyword
About AIP
Laboratories
Videos
Events
News
Opportunities
RAIDEN
人工知能（AI）研究用計算機システム
Riken AIp Deep learning ENvironment

News RSS

posted on October 20, 2021 12:37 NEW Award Information

2021 METI Minister's Award for Industrial Standardization Contribution

Shin-ichi Fukuzumi, Research Scientist of Science, Technology and Society Team (PI:Osamu Sakura) received the 2021 METI* Minister's Award for Industrial Standardization Contribution. *Ministry of Economy, Trade and...

posted on October 14, 2021 12:38 Information

Quantitative Understanding of VAE as a Non-linearly Scaled Isometric Embedding

A paper entitled Quantitative Understanding of VAE as a Non-linearly Scaled Isometric Embedding was featured by NIKKEI XTECH. This paper was accepted at the International Conference on Machine Learning (ICML) 2021,...

posted on October 13, 2021 22:06 Information

AI Research in Japan: Challenges at RIKEN Center for Advance Intelligence Project

Events RSS

posted on October 5, 2021 13:52 Seminar

Succinct Information Processing Unit Seminar (Talk by Dr. Yoshitaka YAMAMOTO, Sizuoka University)

Date October 22, 2021 (Fri) 15:00 - 16:00

posted on October 7, 2021 12:42 Seminar

Structured Learning Team Seminar (Talk by Dr. David Ha, Google Brain)

Date October 25, 2021 (Mon) 15:00 - 17:00

posted on October 13, 2021 14:01 Seminar

High-dimensional Statistical Modeling Team

Videos

Introduction Video

AIP Introduction

Show details

Videos

Calendar

Oct 2021

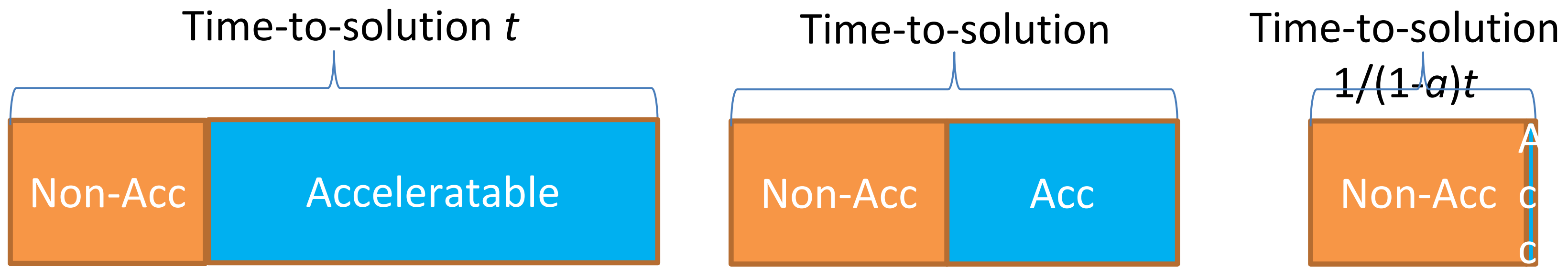
S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

What are the applications we desire quantum accelerator?

- **Applications that are infeasible to solve on conventional computers due to high complexity => impractical time-to-solution**
  - Material science- first principles simulations (wave functions)
  - Difficult higher-order problems difficult to solve with conventional means due to high complexity : O(n^k) where k > 4
  - Other examples: cryptography (much harder)
- **Applications that can be solved with existing computers but beneficial on quantum computers due to cheaper OPEX/CAPX**
  - Optimization problems – TSP and variants
  - Some classes of AI/DL – variational solvers, quantum learning
- **They are important applications, OTOH the list is unfortunately not very long (and likely will not be…)**

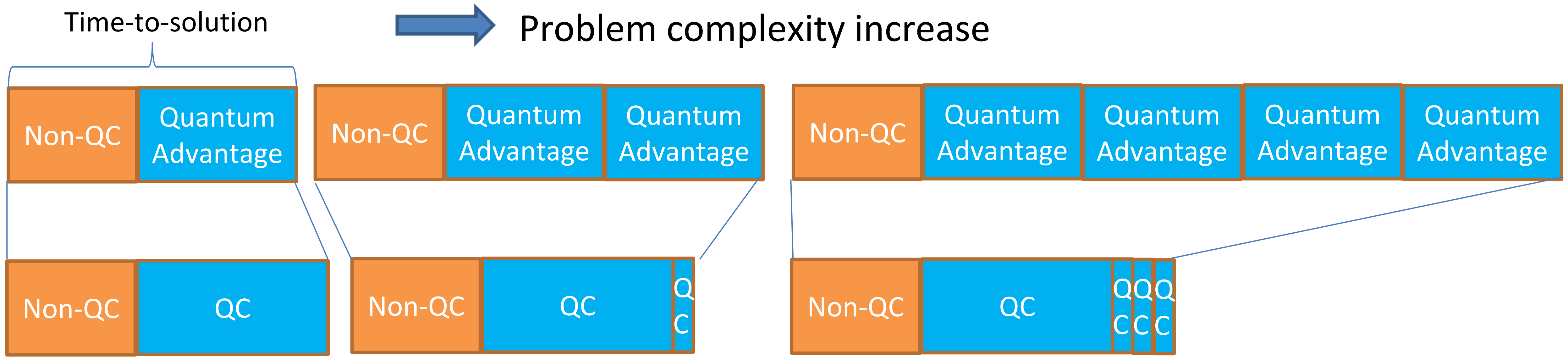
Quantum Computers as Amdahl Accelerators

- **Accelerators are subject to Amdahl's law (strong scaling)**



For accelerators to work, non-accelerated portion must be as small as possible

- **Possible (polynomial?) speedup for NP-Hard problems, exponential speedup for HSP problems**



Research for Quantum Computing/Computer (QC) @ R-CCS

① Development of large-scale QC simulators using Fugaku

- Bracket simulator(R-CCS Ito Team) Large-medium scale (#qubits<50)
- Qulacs simulator (RQC Fujii Team) Medium-small scale (#qubits<30)
- QC simulator designed using Tensor-network (R-CCS yunoki Team)

◆ Development supported by Program "The enhancement of Fugaku useability" for Fugaku CPU resource.

Integrated

② Design of Hybrid programming environment for integration of QC (simulator) and classic HPC supercomputer

- Workflow and task-parallel programming model for offloading for QC (R-CCS Sato Team)
- Design and implementation for a common framework such as Qibo(IHPC, Singapore)

Execute

Tech Transfer

Collaboration with RIKEN Center for Quantum Computing (RQC)

③ Design of QC algorithm and Development of QC applications for QC and HPC hybrid computing

- Target：**Material simulation for the optimization of ground state of molecules by VQE method using more than 40 qubits of QC simulator on Fugaku.**
- **It is expected to be used for the real QC developed by RQC.**

◆ Supported by a special program in the by Program "The enhancement of Fugaku useability" for Fugaku CPU resource.

◆ Access to the external real QC (IBM Q, D-WAVE)

④ Research on the architecture to accelerate QC simulation

◆ To accelerate QC research, the technology for high-speed QC simulation is important. (R-CCS Kondo Team)

◆ The acquisition of GPU-based system (NVIDIA A100)

◆ Expected to use the outcome for Fugaku Next

⑤ Inter-national collaborations

- IHPC(Singapore)　・CEA (France)

If you are excited about future of HPC…

- “Feasibility Study 2.0” for Fugaku NEXT starting Apr 2022, ~$4m USD/y
- We are hiring!
- Team/Unit leaders (digital twins, possibly more in future)
- Various researcher and post-doc positions
- For details ‘google’ Riken R-CCS Home page

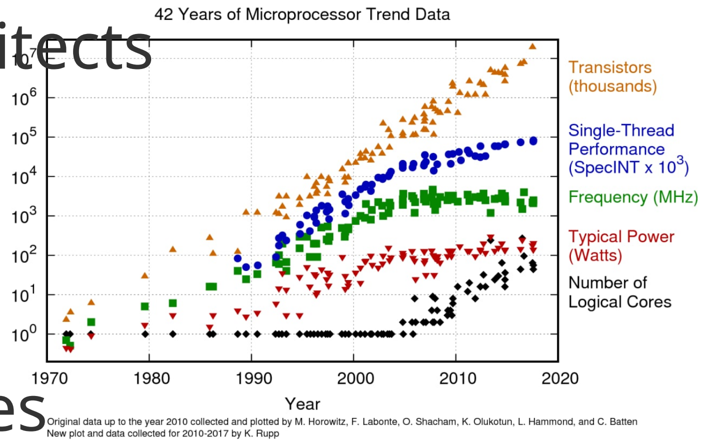
Future architecture perferformance analysis (including AI) for future systems – Building a new methodology @ Riken R-CCS & partners

Future systems

- Methodology to design future systems
- New&better co-design for between doman scientists and system architects

Simulation targets

- Apps, Miniapps, Kernels
- AI models, layers, primitives
- ‘Octopods’*



Investigation components

- Vector extentions
- Matrix engines
- Memory subsystems
- Strong Scaling Accl.

Tools

- Simulators: Riken simulator, Gem5, SST,
- Instrumentation: PIN, DynamoRIO
- Benchmarks: ‘Continuous benchmarking platform’

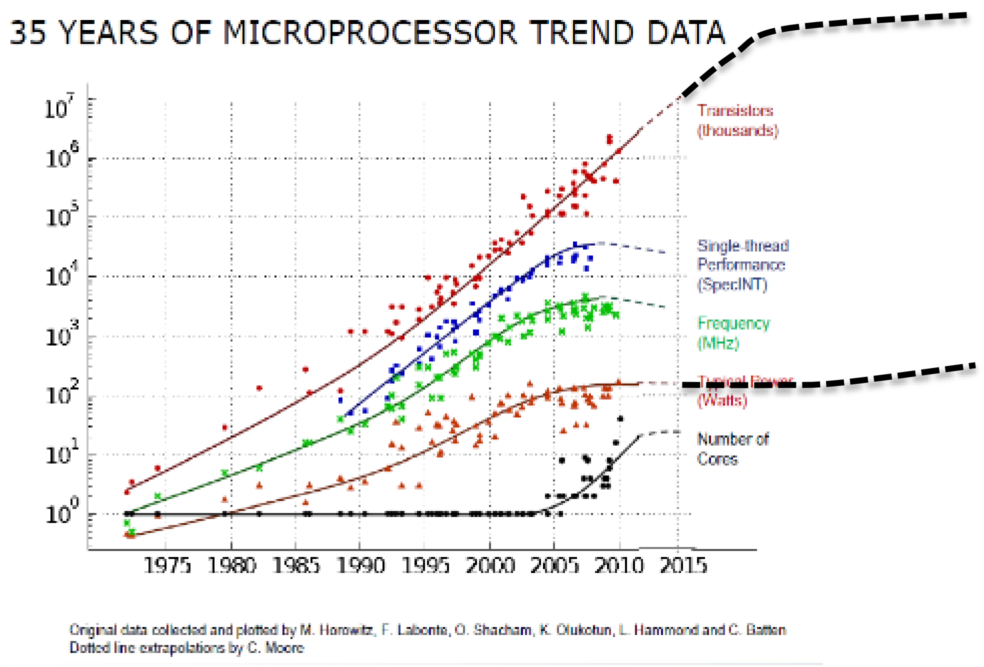
References: “Preparing for the Future – Rethinking Proxy Apps”
Satoshi Matsuoka, Jens Domke, Mohamed Wahib, Aleksandr Drozd, Ray Bair, Andrew A. Chien, Jeffrey S. Vetter, and John Shalf, to be published as CiSE article, 2022.

New Efforts at R-CCS towards Non-Quantum Future

- **New!: Comprehensive benchmarking platform effort**
  - Collect benchmarks and machines incl. Fugaku also x86&GPU
  - Construct a platform to do all benches x all machines benchmarking
  - Make all benchmarks be repeatedly executable so that new instrumentations can be done easily
  - Couple with architectural simulators to conduct what-if analysis
- **New!: Enhancing system software robustness, contributing to compilers and other performance OSS tools (Continuous Benchmarking)**
  - Make Fugaku be performance robust, not focus on co-design apps
  - OSS as future dev platform e.g. LLVM and contribute result to community
  - New optimizations for new architectures before actual HW
  - ‘Platform the benchmarks’ – allow ‘continuous’ benchmarking, archive results automatically, track applications, system SW & HW evolutions, etc.

2028: Post-Moore Era

～2015	25 years of sustained scaling in the Manycore period (Post-Dennard scaling)
2016～	Difficulty in advancing Moore's law
2025～	Post-Moore Era The end of transistor-power advancement



Challenge: **Exploration of computer architectures that will enable performance improvement even around the year 2028**

Key to sustained performance improvement:

FLOPS to Bytes, "data movement-centric architecture"

- ✓ Reconfigurable, data-driven, vector computing
- ✓ Ultra-deep and ultra-wide bandwidth memory architectures
- ✓ Optical networks
- ✓ system software, programing, algorithms that correspond to new architectures