The first “exascale” supercomputer Fugaku – HPC, BD & AI

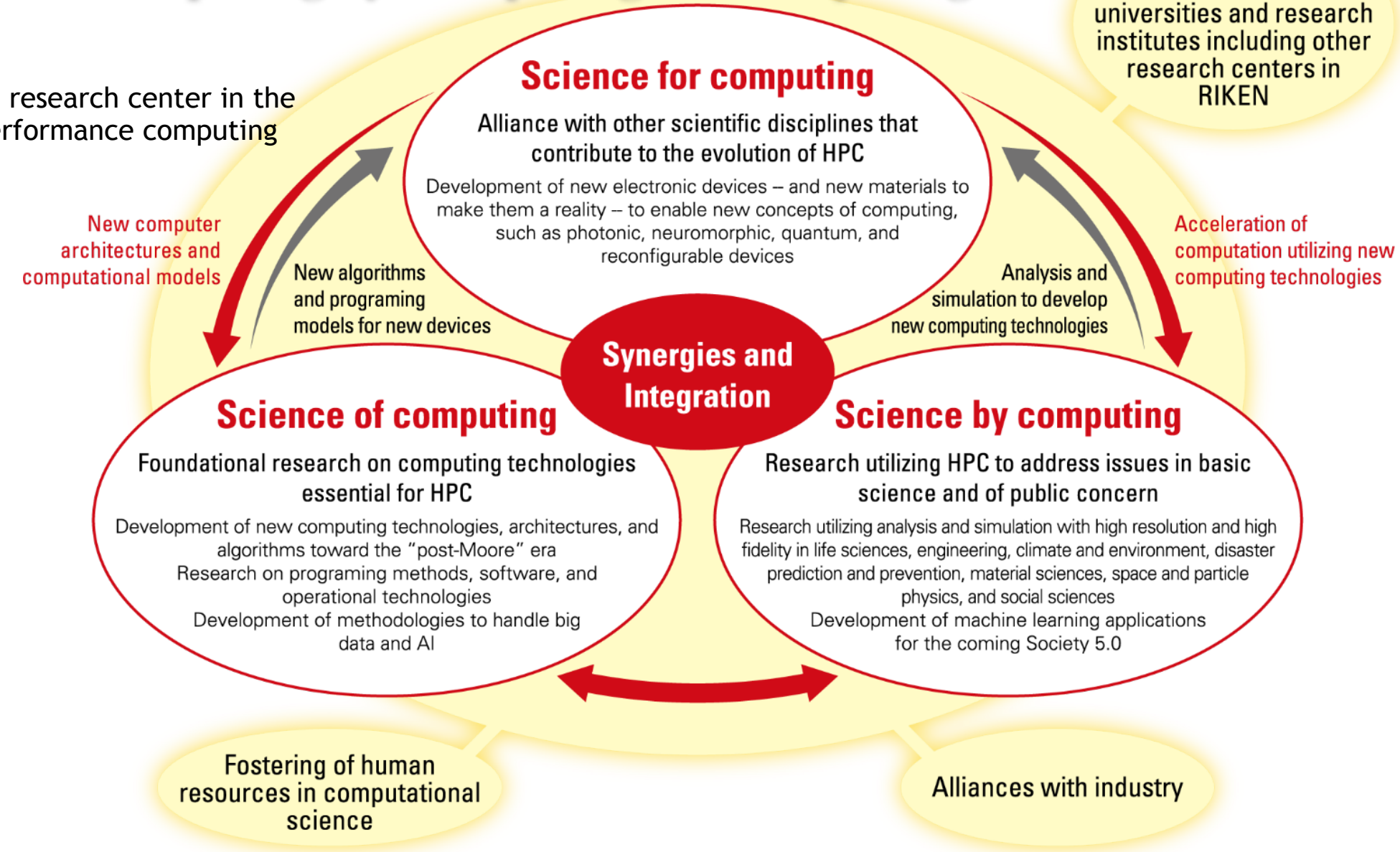
- ***Satoshi Matsuoka***
- ***Director, RIKEN Center for Computational Science***
- **20191106 France-Germany-Japan Presentation @ Tokyo**

RIKEN R-CCS

Science of Computing by Computing for Computing

R-CCS

International core research center in the science of high performance computing (HPC)



Challenges Ahead for R-CCS

- **1. Launching, Operating, and Improving 'Fugaku' – the first 'Exascale' Supercomputer for Simulation, Big Data and AI**

- **2. Extreme improvements in convergence of HPC for AI**
  - Improving processor performance for inference & training
  - Extreme data parallelism for extreme scaling
  - Incorporating model parallelism for performance and ultra large neural networks

  … and AI for HPC (challenges in apps & algorithms)

- **3. Big data with IoT and HPC convergence --- how to process data WITHOUT moving or storing them**
  - Not just traditional compression, filtering…
- **4. Post-Moore computing towards 2030s --- sustainable future for HPC, Big Data, and AI (and Fugaku-Next)**

Challenges Ahead for R-CCS

Today's Talk

- **1. Launching, Operating, and Improving 'Fugaku' – the first 'Exascale' Supercomputer for Simulation, Big Data and AI**

- **2. Extreme improvements in convergence of HPC for AI**
  - Improving processor performance for inference & training
  - Extreme data parallelism for extreme scaling
  - Incorporating model parallelism for performance and **ultra large neural networks beyond 10s GByte**

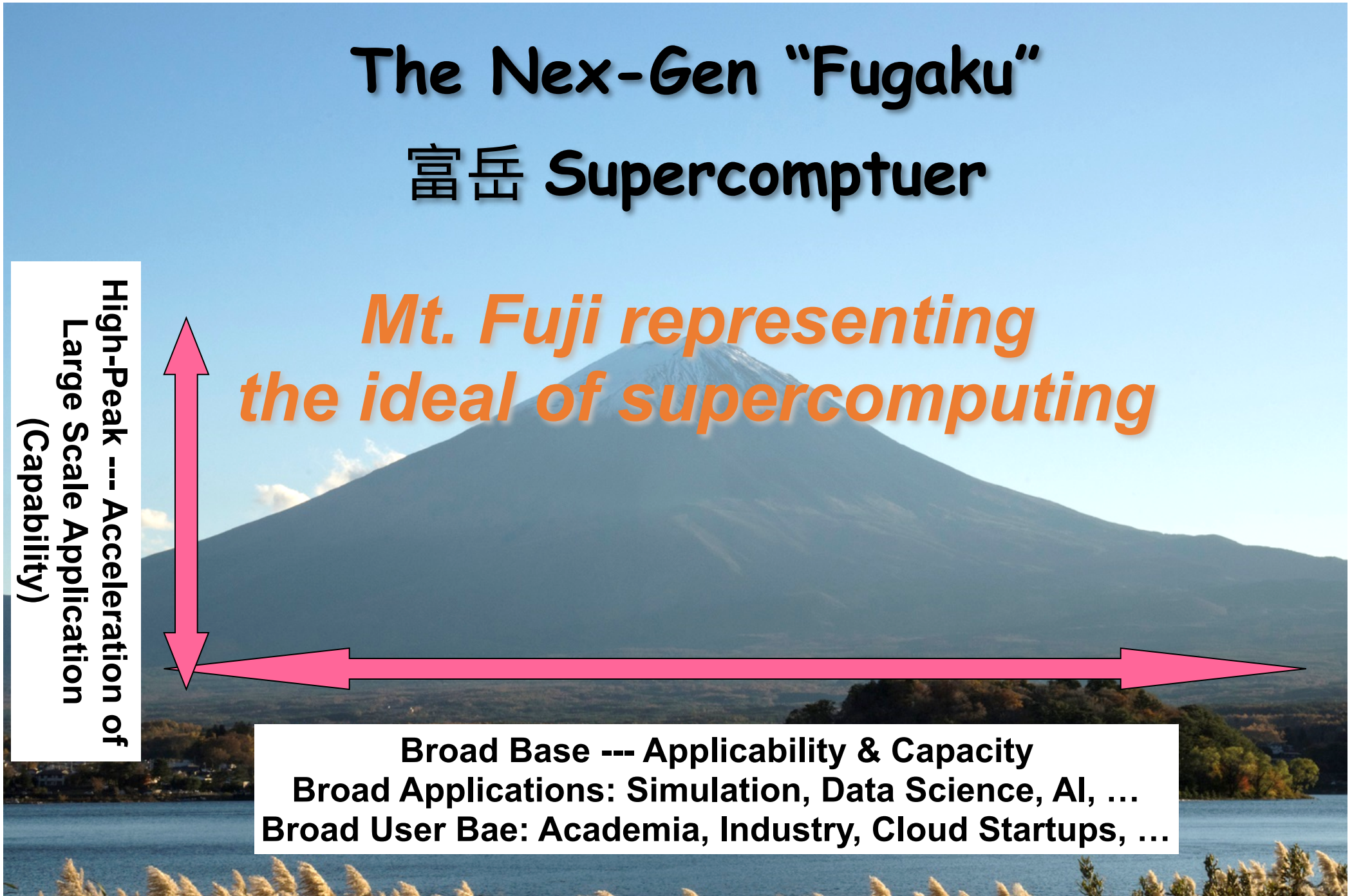
  … and AI for HPC (challenges in apps & algorithms)

- **3. Big data with IoT and HPC convergence --- how to process data WITHOUT moving or storing them**
  - Not just traditional compression, filtering…
- **4. Post-Moore computing towards 2030s --- sustainable future for HPC, Big Data, and AI (and Fugaku-Next)**

The ‘Fugaku’ Supercomputer, Successor to the K-Computer

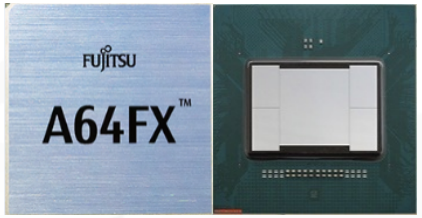
The Nex-Gen "Fugaku"

富岳 Supercomptuer

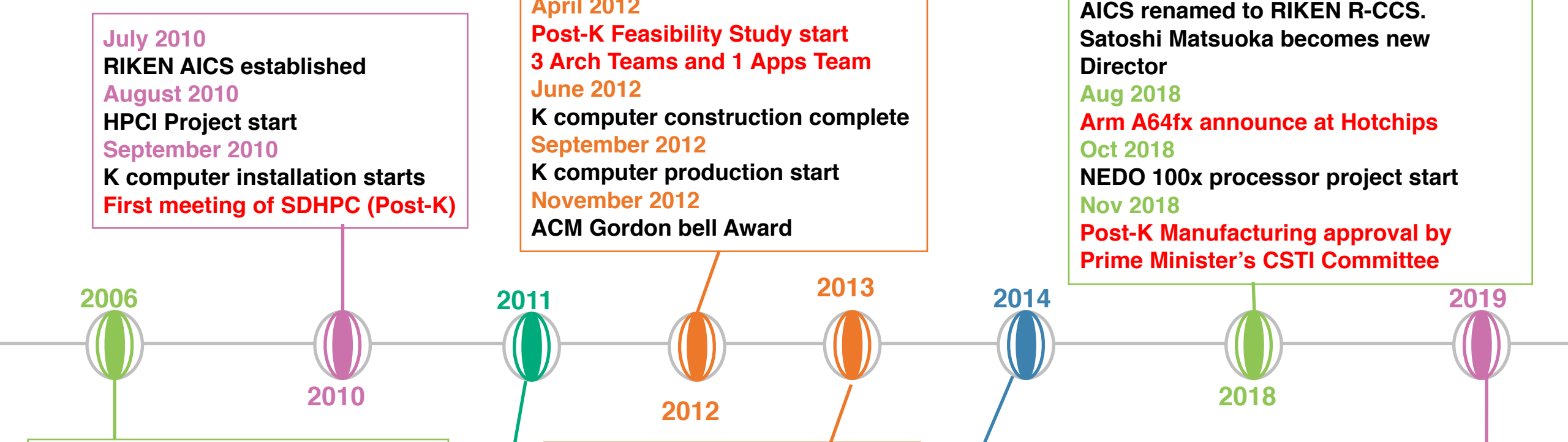


Arm64fx & Fugaku 富岳 /Post-K are:

- **Fujitsu-Riken design A64fx ARM v8.2 (SVE), 48/52 core CPU**
  - ***HPC Optimized:*** Extremely high package high memory BW (1TByte/s), on-die Tofu-D network BW (~400Gbps), high SVE FLOPS (~3Teraflops), various AI support (FP16, INT8, etc.)
  - Gen purpose CPU – Linux, Windows (Word), other SCs/Clouds
  - Extremely power efficient – > **10x power/perf efficiency for CFD benchmark** over current mainstream x86 CPU
- **Largest and fastest supercomputer to be ever built circa 2020**
  - > 150,000 nodes, superseding LLNL Sequoia
  - > 150 PetaByte/s memory BW
  - Tofu-D 6D Torus NW, 60 Petabps injection BW (10x global IDC traffic)
  - 25~30PB NVMe L1 storage
  - The first 'exascale' machine (not exa64bitflops =>apps perf.)
  - **Acceleration of HPC, Big Data, and AI to extreme scale**



Brief History of R-CCS towards Fugaku



January 2006
Next Generation Supercomputer Project (K Computer) start

July 2010
RIKEN AICS established
August 2010
HPCI Project start
September 2010
K computer installation starts
First meeting of SDHPC (Post-K)

June 2011
#1 on Top 500
November 2011
#1 on Top 500 > 10 Petaflops
ACM Gordon Bell Award
End of FY 2011 (March 2012)
SDHPC Whitepaper

April 2012
Post-K Feasibility Study start
3 Arch Teams and 1 Apps Team
June 2012
K computer construction complete
September 2012
K computer production start
November 2012
ACM Gordon bell Award

End of FY2013 (Mar 2014)
Post-K Feasibility Study Reports

April 2014
Post-K project start
June 2014
#1 on Graph 500

April 2018
AICS renamed to RIKEN R-CCS.
Satoshi Matsuoka becomes new Director
Aug 2018
Arm A64fx announce at Hotchips
Oct 2018
NEDO 100x processor project start
Nov 2018
Post-K Manufacturing approval by Prime Minister's CSTI Committee

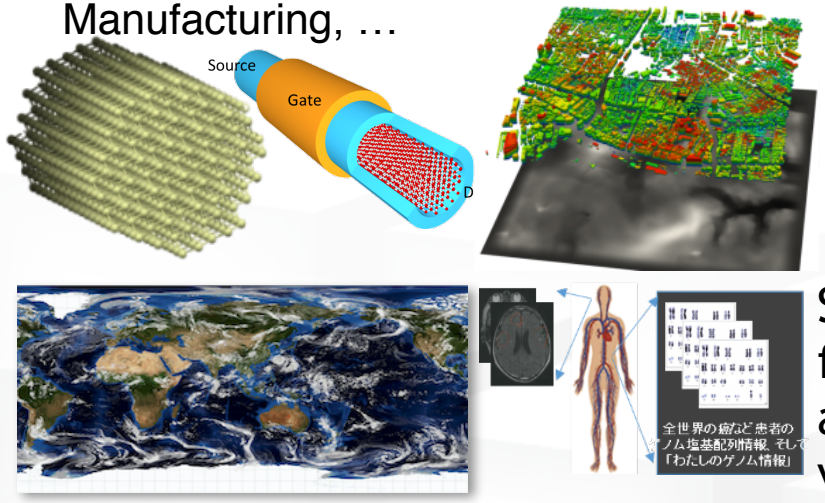
March 2019
Post-K Manufacturing start
May 2019
Post-K named "Supercomputer Fugaku"
July 2019
Post-Moore Whitepaper start
Aug 2019
K Computer shutdown
Dec 2019
Fugaku installation start (planned)

Co-Design Activities in Fugaku

Multiple Activities since 2011

Science by Computing

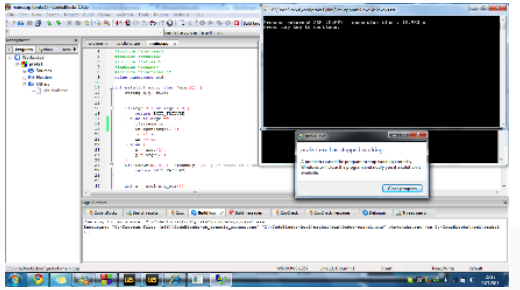
・9 Priority Application Areas: High Concern to General Public: Medical/Pharma, Environment/Disaster, Energy, Manufacturing, …



Select representatives from 100s of applications signifying various computational characteristics

Science of Computing

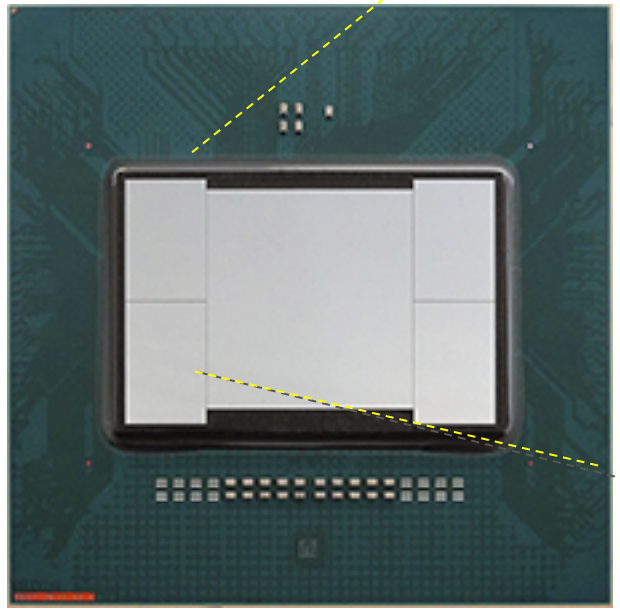
Design systems with parameters that consider various application characteristics

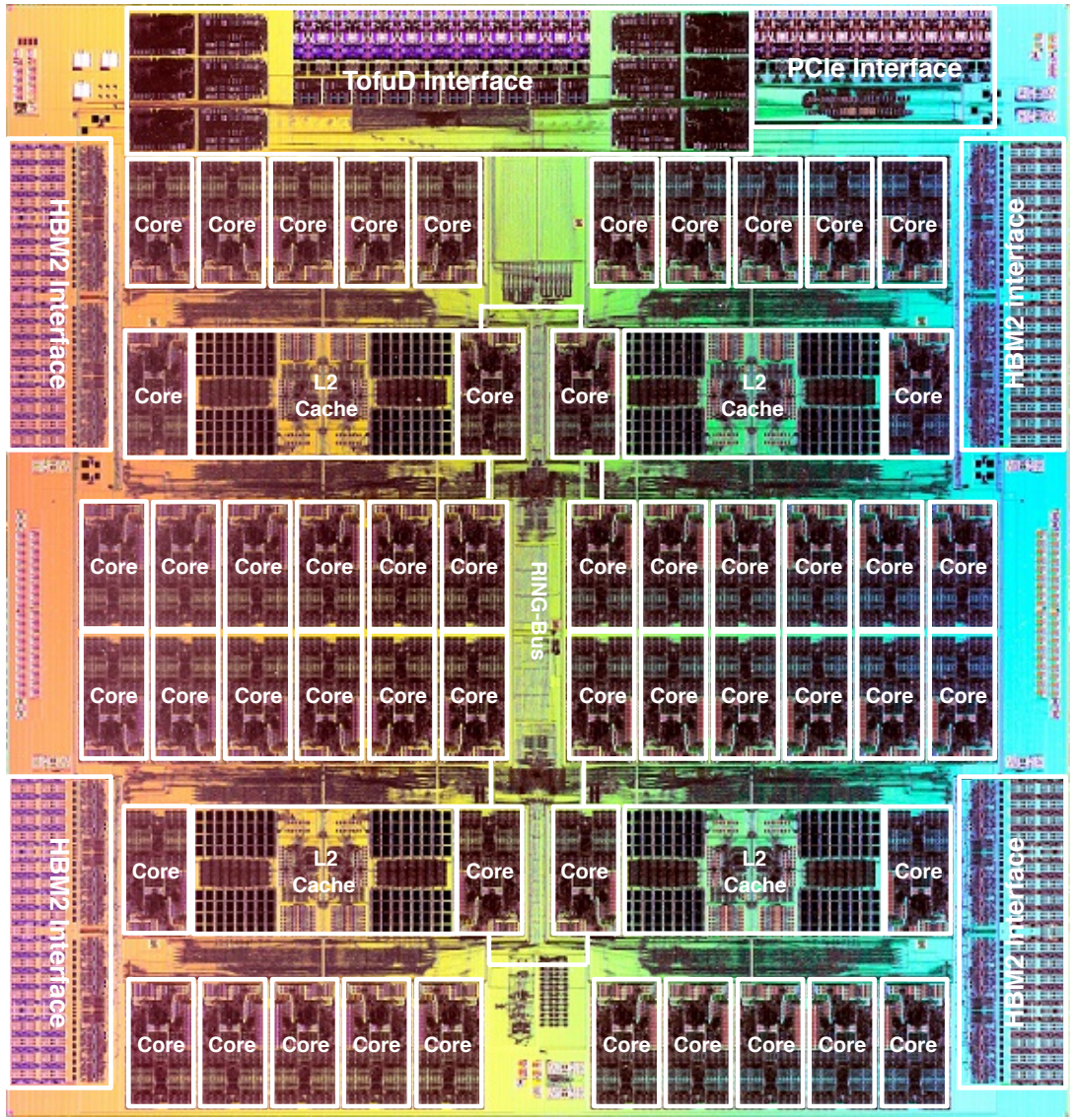


- **Extremely tight collabrations between the Co-Design apps centers, Riken, and Fujitsu, etc.**
- **Chose 9 representative apps as “target application” scenario**
- **Achieve up to x100 speedup c.f. K-Computer**
- **Also ease-of-programming, broad SW ecosystem, very low power, …**

A64FX Leading-edge Si-technology

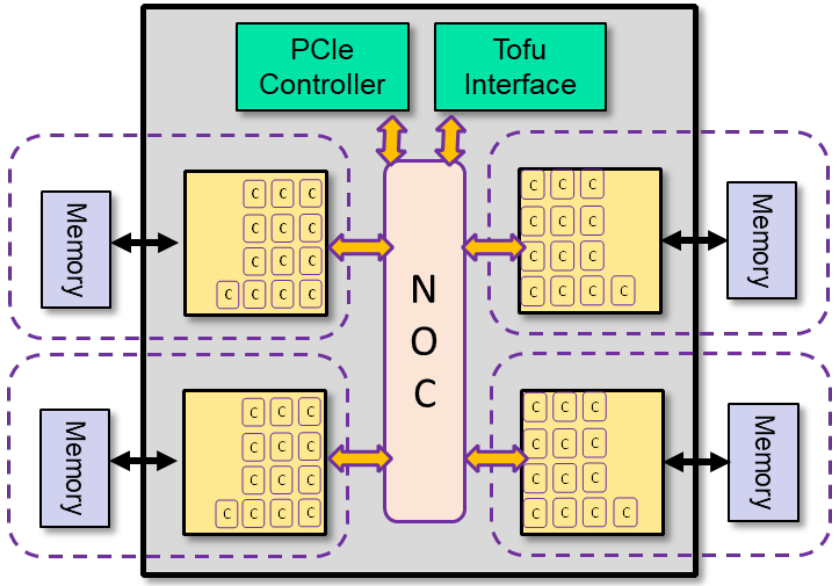
- TSMC 7nm FinFET & CoWoS
  - Broadcom SerDes, HBM I/O, and SRAMs
  - 8.786 billion transistors
  - 594 signal pins





Fugaku's FUjitsu A64fx Processor is…

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



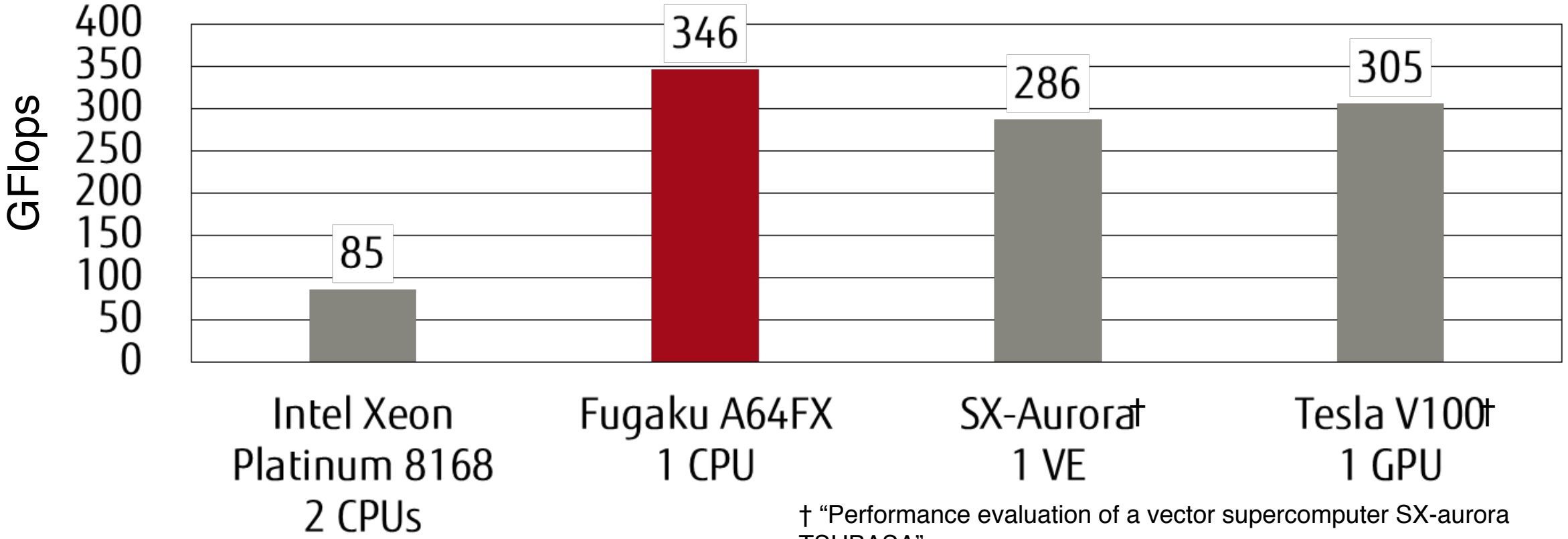
- **…but also an accelerated GPU-like processor**
  - SVE 512 bit x 2 vector extensions (ARM & Fujitsu)
    - Integer (1, 2, 4, 8 bytes) + Float (16, 32, 64 bytes)
  - Cache + scratchpad-like local memory (sector cache)
  - HBM2 on package memory – Massive Mem BW (Bytes/DPF ~0.4)
    - Streaming memory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

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

“Fugaku” CPU Performance Evaluation (2/3)

FUJITSU

- Himeno Benchmark (Fortran90)
  - Stencil calculation to solve Poisson’s equation by Jacobi method

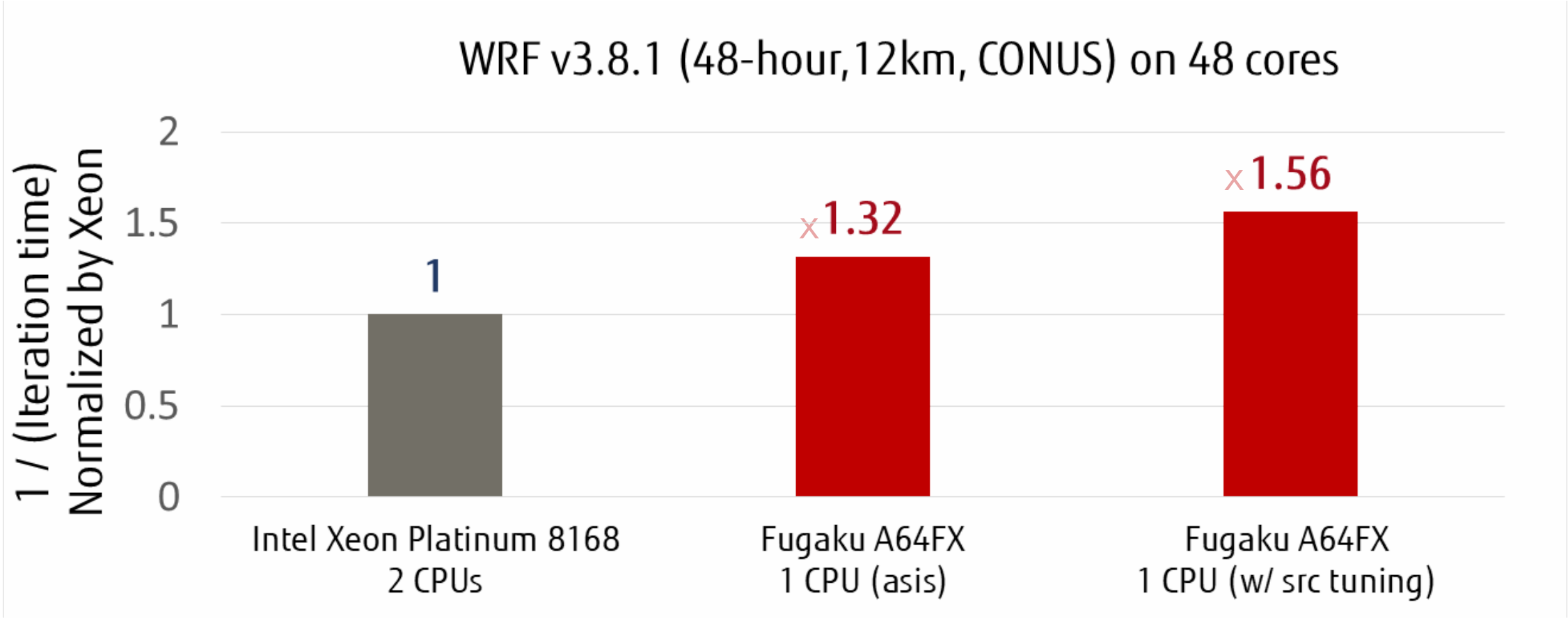


† “Performance evaluation of a vector supercomputer SX-aurora TSUBASA”, SC18, https://dl.acm.org/citation.cfm?id=3291728

“Fugaku” CPU Performance Evaluation (3/3)

FUJITSU

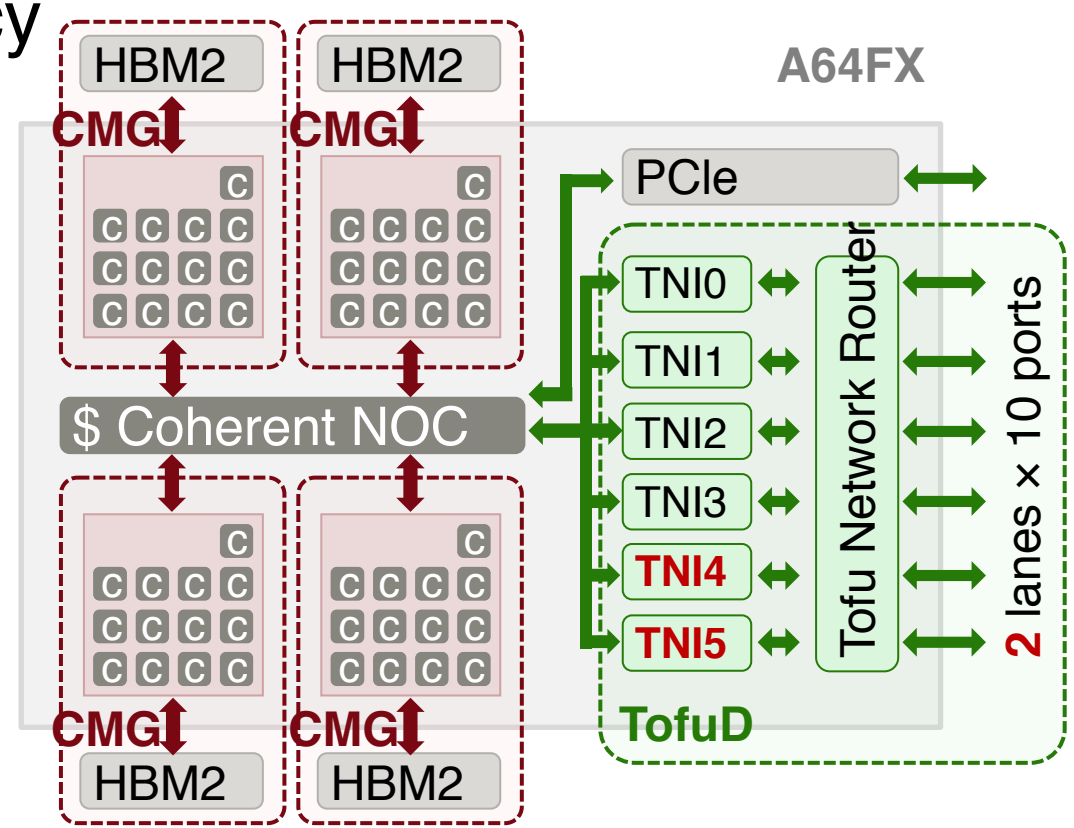
- WRF: Weather Research and Forecasting model
  - Vectorizing loops including IF-constructs is key optimization
  - Source code tuning using directives promotes compiler optimizations



A64FX: Tofu interconnect D

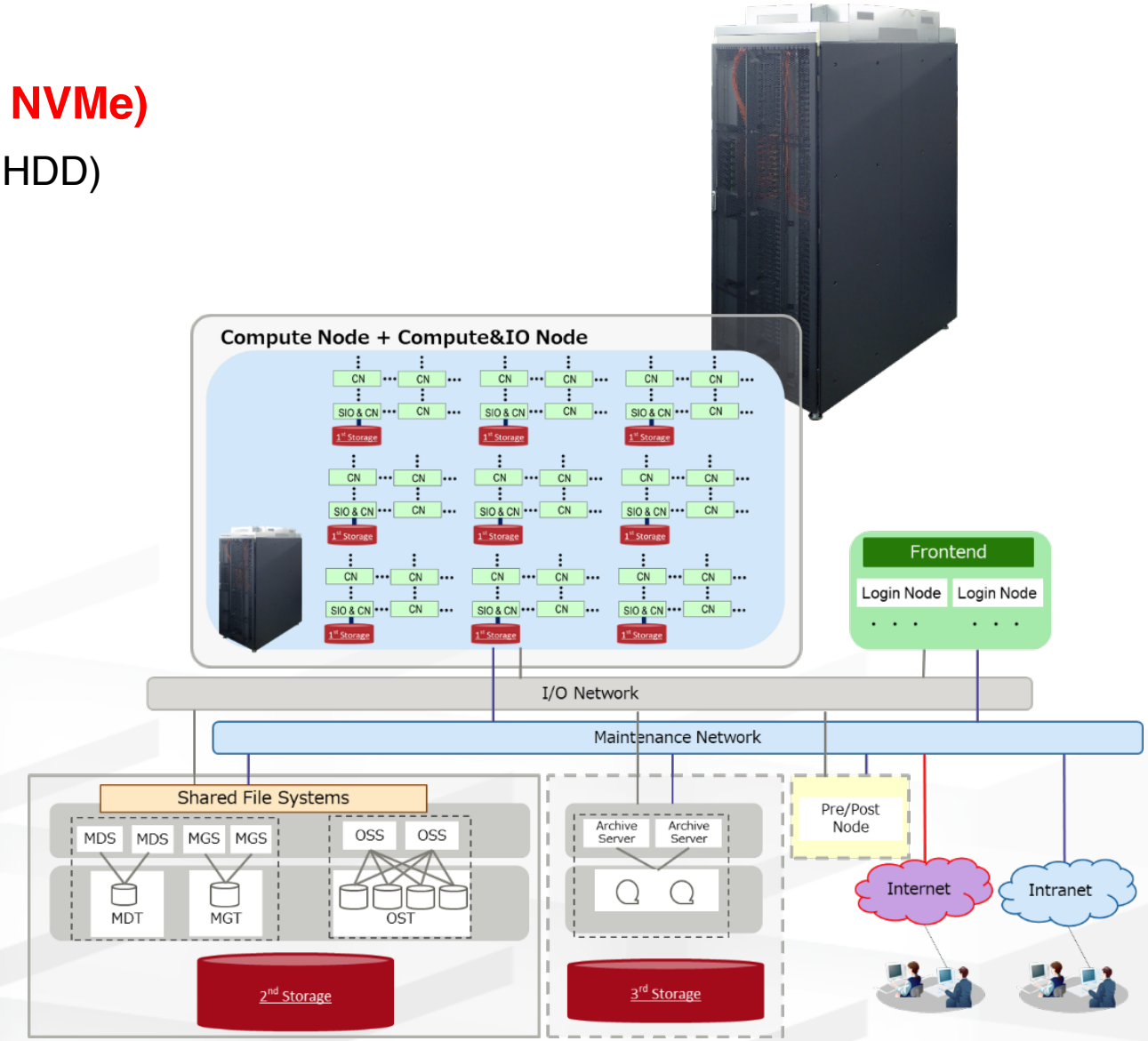
- Integrated w/ rich resources
  - Increased TNIs achieves higher injection BW & flexible comm. patterns
  - Increased barrier resources allow flexible collective comm. algorithms
- Memory bypassing achieves low latency
  - Direct descriptor & cache injection

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



Overview of Fugaku System & Storage

- **3-level hierarchical storage**
  - 1st Layer: GFS Cache + Temp FS **(25~30 PB NVMe)**
  - 2nd Layer: Lustre-based GFS (a few hundred PB HDD)
  - 3rd Layer: Off-site Cloud Storage
- **Full Machine Spec**
  - **>150,000 nodes**
    ~8 million High Perf. Arm v8.2 Cores
  - **> 150PB/s memory BW**
  - **Tofu-D 10x Global IDC traffic @ 60Pbps**
  - **~10,000 I/O fabric endpoints**
  - **> 400 racks**
  - **~40 MegaWatts Machine+IDC**
    PUE ~ 1.1 High Pressure DLC
  - **NRE pays off: ~= 15~30 million state-of-the art competing CPU Cores for HPC workloads (both dense and sparse problems)**



Fugaku Performance Estimate on 9 Co-Design Target Apps

Performance target goal

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

Peak performance to be achieved

	PostK	K
Peak DP (double precision)	>400+ Pflops (34x +)	11.3 Pflops
Peak SP (single precision)	>800+ Pflops (70x +)	11.3 Pflops
Peak HP (half precision)	>1600+ Pflops (141x +)	--
Total memory bandwidth	>150+ PB/sec (29x +)	5,184TB/sec

Geometric Mean of Performance Speedup of the 9 Target Applications over the K-Computer

> 37x+

As of 2019/05/14

Category	Priority Issue Area	Performance Speedup over K	Application	Brief description
Health and longevity	1. Innovative computing infrastructure for drug discovery	**125x +**	GENESIS	MD for proteins
	2. Personalized and preventive medicine using big data	8x +	Genomon	Genome processing (Genome alignment)
Disaster prevention and Environment	3. Integrated simulation systems induced by earthquake and tsunami	45x +	GAMERA	Earthquake simulator (FEM in unstructured & structured grid)
	4. Meteorological and global environmental prediction using big data	**120x +**	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	40x +	NTChem	Molecular electronic simulation (structure calculation)
	6. Accelerated development of innovative clean energy systems	35x +	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	30x +	RSDFT	Ab-initio simulation (density functional theory)
	8. Development of innovative design and production processes	25x +	FFB	Large Eddy Simulation (unstructured grid)
Basic science	9. Elucidation of the fundamental laws and evolution of the universe	25x +	LQCD	Lattice QCD simulation (structured grid Monte Carlo)

Fugaku Programming Environment

- **Programing Languages and Compilers provided by Fujitsu**
  - Fortran2008 & Fortran2018 subset
  - C11 & GNU and Clang extensions
  - C++14 & C++17 subset and GNU and Clang extensions
  - OpenMP 4.5 & OpenMP 5.0 subset
  - Java

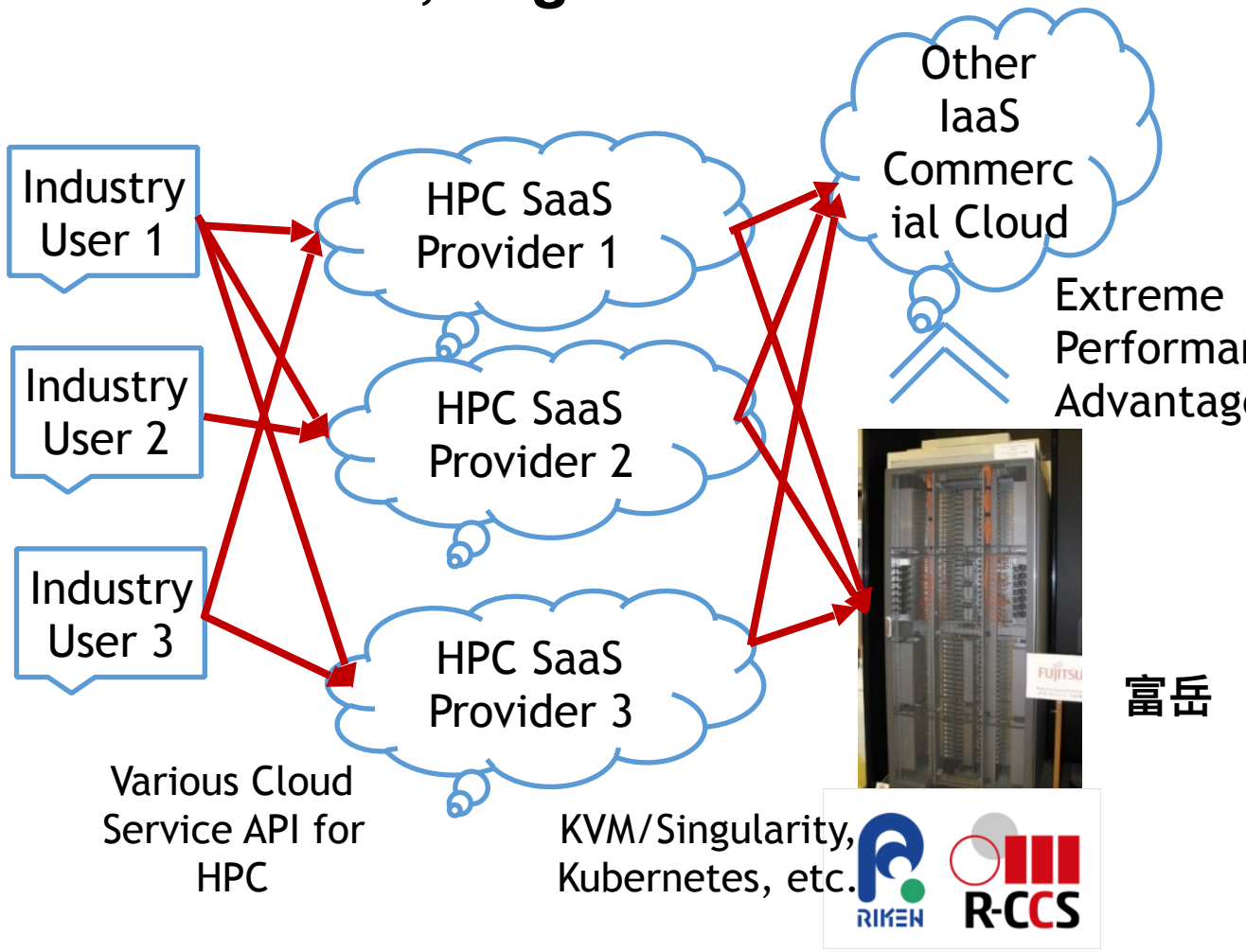
GCC and LLVM will be also available

- **Parallel Programming Language & Domain Specific Library provided by RIKEN**
  - XcalableMP
  - FDPS (Framework for Developing Particle Simulator)
- **Process/Thread Library provided by RIKEN**
  - PiP (Process in Process)

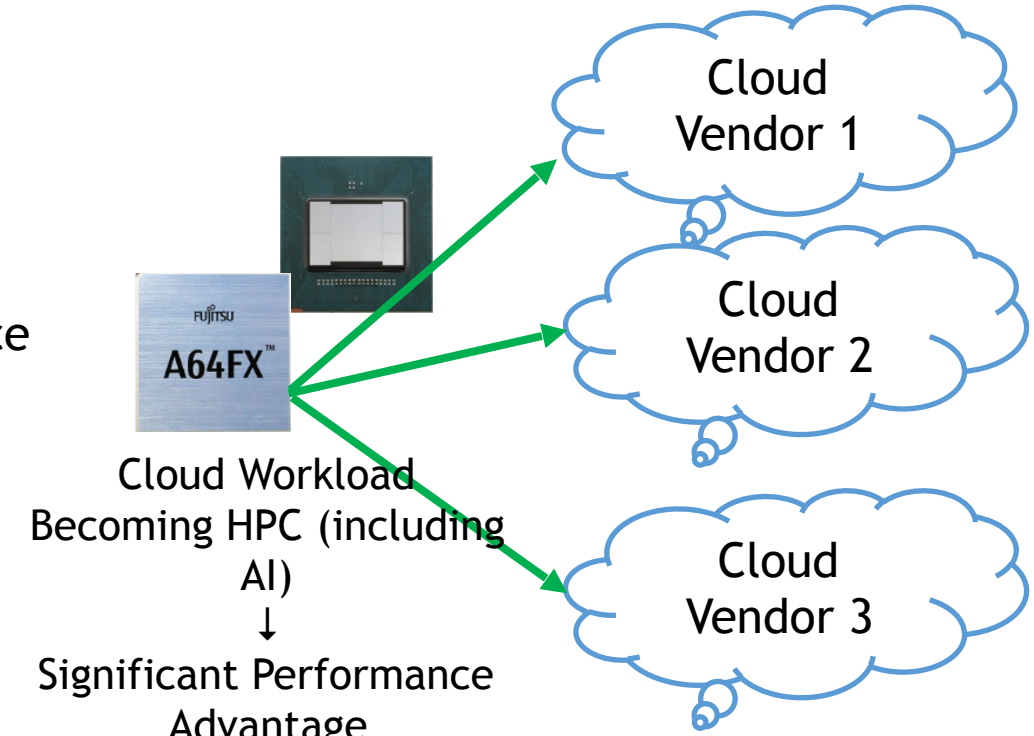
- **Script Languages provided by Linux distributor**
  - E.g., Python+NumPy, SciPy
- **Communication Libraries**
  - MPI 3.1 & MPI4.0 subset
    - Open MPI base (Fujitsu), MPICH (RIKEN)
  - Low-level Communication Libraries
    - uTofu (Fujitsu), LLC(RIKEN)
- **File I/O Libraries provided by RIKEN**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

Fugaku Cloud Strategy

- **Industry use of Fugaku via intermediary cloud SaaS vendors, Fugaku as IaaS**



- **A64fx and other Fugaku Technology being incorporated into the Cloud**



A64fx in upcoming Stony Brook Cray System

National Science Foundation
WHERE DISCOVERIES BEGIN

SEARCH

HOME | RESEARCH AREAS | FUNDING | AWARDS | DOCUMENT LIBRARY | NEWS | ABOUT NSF

Awards

- Search Awards
- Recent Awards
- Presidential and Honorary Awards
- About Awards

How to Manage Your Award

- Grant Policy Manual
- Grant General Conditions
- Cooperative Agreement Conditions
- Special Conditions
- Federal Demonstration Partnership
- Policy Office Website

Award Abstract #1927880

Category II : Ookami: A high-productivity path to frontiers of scientific discovery enabled by exascale system technologies

NSF Org:	OAC Office of Advanced Cyberinfrastructure (OAC)
Initial Amendment Date:	July 11, 2019
Latest Amendment Date:	August 29, 2019
Award Number:	1927880
Award Instrument:	Cooperative Agreement
Program Manager:	Robert Chadduck OAC Office of Advanced Cyberinfrastructure (OAC) CSE Direct For Computer & Info Scie & Enginr
Start Date:	October 1, 2019
End Date:	September 30, 2024 (Estimated)
Awarded Amount to Date:	$2,780,373.00
Investigator(s):	Robert Harrison robert.harrison@stonybrook.edu (Principal Investigator) Barbara Chapman (Co-Principal Investigator) Matthew Jones (Co-Principal Investigator) Alan Calder (Co-Principal Investigator)
Sponsor:	SUNY at Stony Brook WEST 5510 FRK MEL LIB Stony Brook, NY 11794-0001 (631)632-9949
NSF Program(s):	Innovative HPC
Program Reference Code(s):	
Program Element Code(s):	7619

ABSTRACT

The State University of New York proposes to procure and operate for at least four years the first computer outside of Japan with the A64fx processor developed by Fujitsu for the Japanese path to exascale computing (i.e., computers capable of 10^18 operations per second). The ARM-based, multi-core, 512-bit SIMD-vector processor with ultrahigh-bandwidth memory promises to retain familiar and successful programming models while achieving very high performance for a wide range of applications including simulation and big data. The testbed significantly extends current NSF-sponsored HPC technologies and will enable the community to evaluate and demonstrate the potential of this technology for deployment in multiple settings. Through integration with NSF's Extreme Science and Engineering Discovery Environment (XSEDE), the system will be widely accessible and fully leverages existing cyber infrastructure including the XDMoD monitoring system.

What does this mean for science? Compared with the best CPUs anticipated during the deployment period, A64fx offers 2-4x better performance on memory-intensive applications such as sparse-matrix solvers found in many engineering and physics codes.

HPC wire

Since 1987 - Covering the Fastest Computers in the World and the People Who Run Them

- Home
- Technologies
- Sectors
- AI/ML/DL
- Exascale
- Specials
- Resource Library
- Podcast
- Events
- Job Bank

Cray ARM-based 'Ookami' to Serve as Testbed for Computational Studies at Stony Brook

August 16, 2019

STONY BROOK, N.Y., August 16, 2019 – A $5 million grant from the National Science Foundation (NSF) to the Institute of Advanced Computational Science (IACS) at Stony Brook University will enable researchers nationwide to test future supercomputing technologies and advance computational and data-driven research on the world's most pressing challenges.

Serving as a testbed for advanced computer technologies, the Ookami system is expected to signal a new generation of high-speed U.S. supercomputers. Using a Cray ARM-based system, Ookami will deliver remarkably high performance for scientific applications, in part due to its blazing-fast memory. Robert J. Harrison, PhD, professor of applied mathematics and statistics and director of IACS, expects that these advanced technologies will enable researchers to more quickly and effectively conduct computational investigations. The project is led by IACS faculty in partnership with co-PI Matt Jones, PhD at the State University of New York at Buffalo, whose team will lead the capture of detailed operational metrics and provision of extensive

Ookami

- Test bed for NSF researchers
  - First planned deployment of the Post-K processor outside of Japan
- Collaboration with Riken CCS
  - http://www.riken.jp/en/research/labs/r-ccs/
- Installation 3Q 2020
- $5M award NSF OAC 1942140 for purchase and operations

Node	
Processor	A64FX
#Cores	48+4
Peak DP	2.76 TOP/s
Peak INT8	22.08 TOP/s
Memory	32GB@1TB/s
System	
#Nodes	176
Peak DP	486 TOP/s
Peak INT8	3886 TOP/s
Memory	5.6 TB
Disk	0.5 PB
Comms	IB HDR-100

Pursuing Convergence of HPC & AI (1)

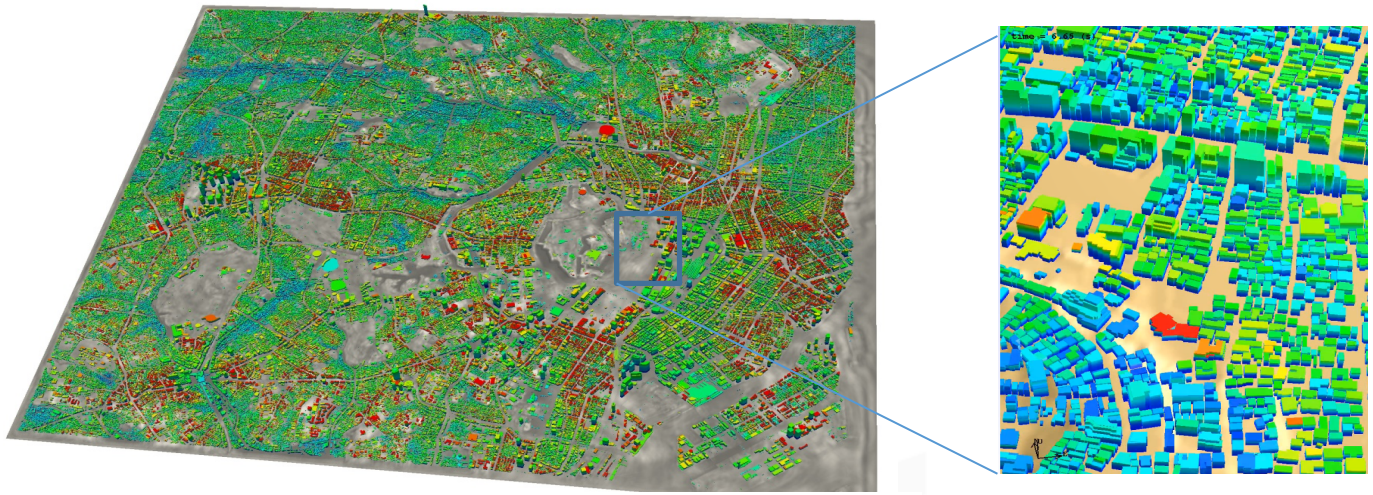
- **Acceleration of Simulation (first principles methods) with AI (empirical method) : *AI for HPC***
  - Interpolation & Extrapolation of long trajectory MD
  - Reducing parameter space on Paretho optimization of results
  - Adjusting convergence parameters for iterative methods etc.
  - AI *replacing* simulation when exact physical models are unclear, or excessively costly to compute
- **Acceleration of AI with HPC: *HPC for AI***
  - HPC Processing of training data -data cleansing
  - Acceleration of (Parallel) Training: Deeper networks, bigger training sets, complicated networks, high dimensional data…
  - Acceleration of Inference: above + real time streaming data
  - Various modern training algorithms: Reinforcement learning, GAN, Dilated Convolution, etc.

R-CCS Pursuit of Convergence of HPC & AI (2)

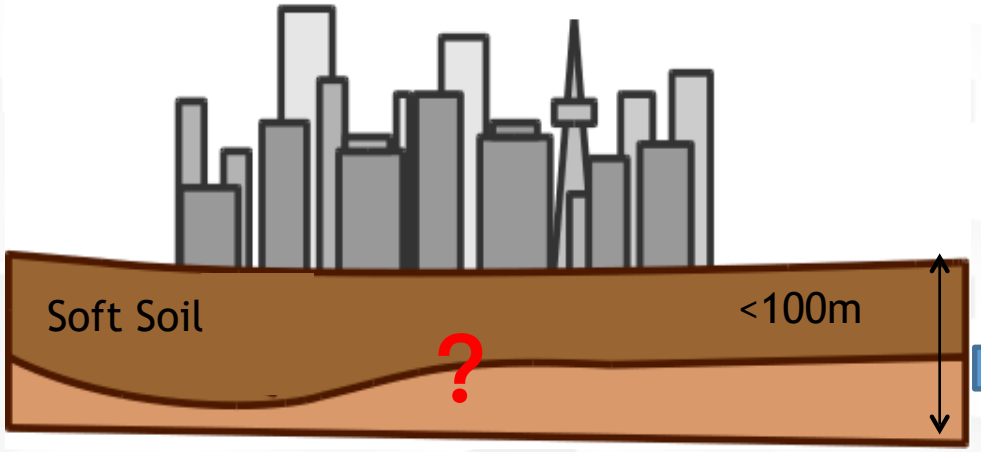
- **Acceleration of Simulation (first principles methods) with AI (empirical method) : *AI for HPC***
  - ***Most*** R-CCS research & operations teams investigating use of AI for HPC
  - 9 priority co-design issues area teams also extensive plans
  - Essential to deploy AI/DL frameworks efficiently & at scale on A64fx/Fugaku

- **Acceleration of AI with HPC: *HPC for AI***
  - New teams instituted in Science of Computing to accelerate AI
    - Kento Sato (High Performance Big Data Systems)
    - Satoshi Matsuoka (High Performance AI Systems)
    - Masaaki Kondo Next Gen (High Performance Architecture)
  - **NEW: Optimized AI/DL Library via port of DNNL (MKL-DNN)**
    - **Arm Research + Fujitsu Labs + Riken R-CCS + others**
    - **First public ver. by Mar 2020, TensorFlow, PyTorch, Chainer, etc.**

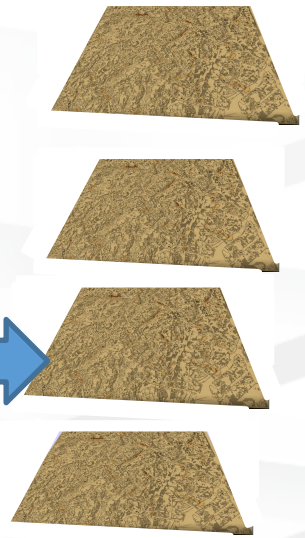
Large Scale simulation and AI coming together [Ichimura et. al. Univ. of Tokyo, IEEE/ACM SC17 Best Poster 2018 Gordon Bell Finalist]



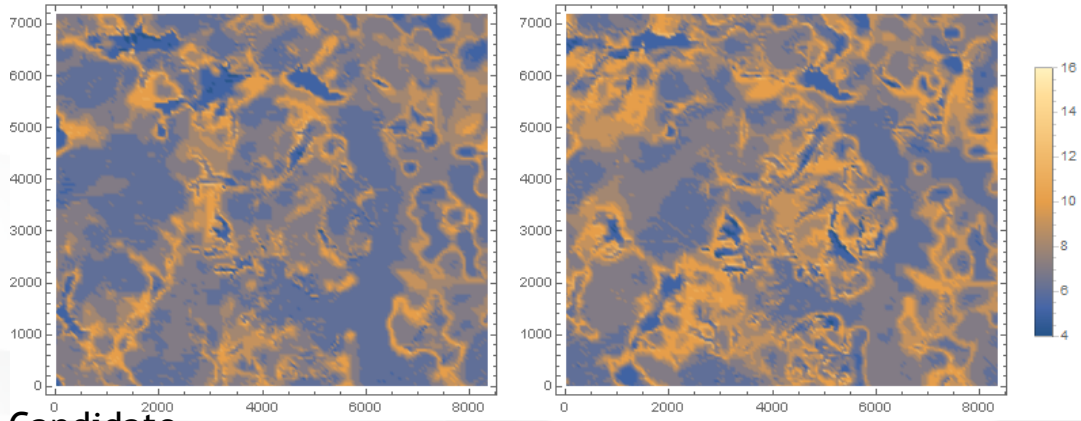
130 billion freedom earthquake of entire Tokyo on K-Computer (2018 ACM Gordon Bell Prize Finalist, SC16,17 Best Poster)



Earthquake



Too Many Instances



Candidate Underground Structure 1

AI Trained by Simulation to generate candidate soft soil structure

Candidate Underground Structure 2

Convergence of HPC & AI in Modsim

- **Performance modeling and prediction with AI (empirical method)** ***AI for modsim of HPC systems***
  - *C.f. GEM5 simulation – first principle perf. modeling*
  - AI Interpolation & Extrapolation of system performance
  - Objective categorization of benchmarks
  - Optimizing system performance using machine learning

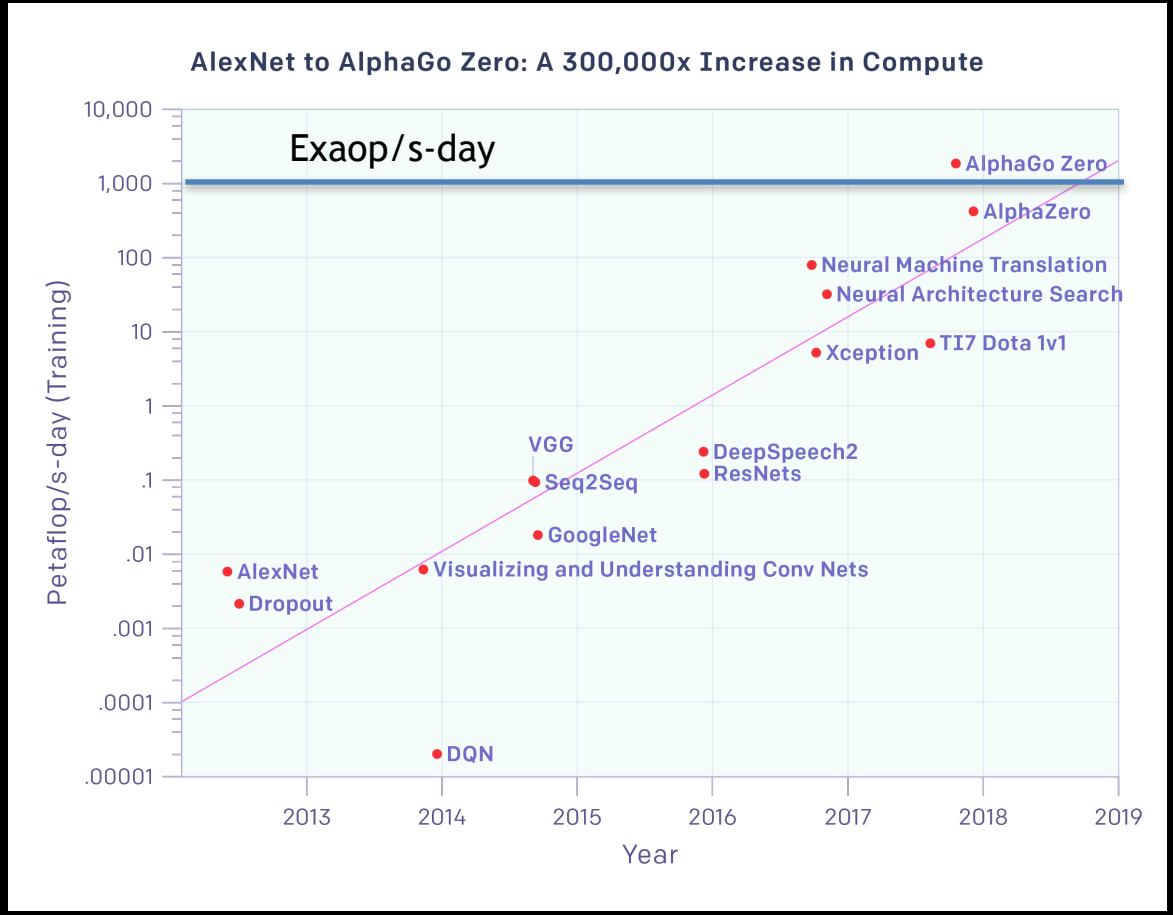
- **Performance Modeling of AI esp. Machine Learning** ***HPC modsim techniques for AI***
  - Perf. modeling of Deep Neural Networks on HPC machines
  - Large scaling of Deep Learning on large scale machines
  - Optimization of AI algorithms using perf modeling
  - Architectural survey and modeling of future AI systems

Deep Learning Meets HPC
6 orders of magnitude compute increase in 5 years

[Slide Courtesy Rick Stevens @ ANL]

Exascale Needs for Deep Learning

- Automated Model Discovery
- Hyper Parameter Optimization
- Uncertainty Quantification
- Flexible Ensembles
- Cross-Study Model Transfer
- Data Augmentation
- Synthetic Data Generation
- Reinforcement Learning

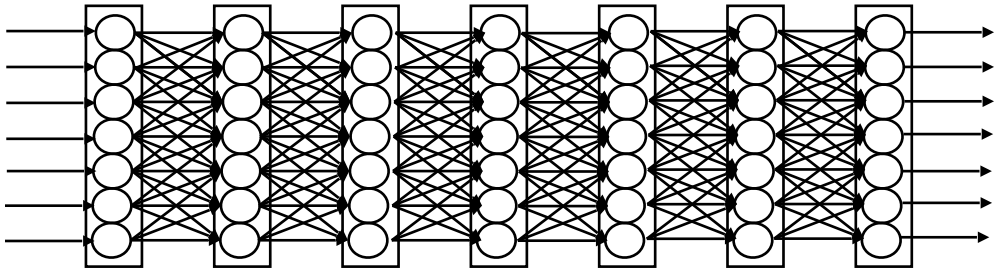


4 Layers of Parallelism in DNN Training

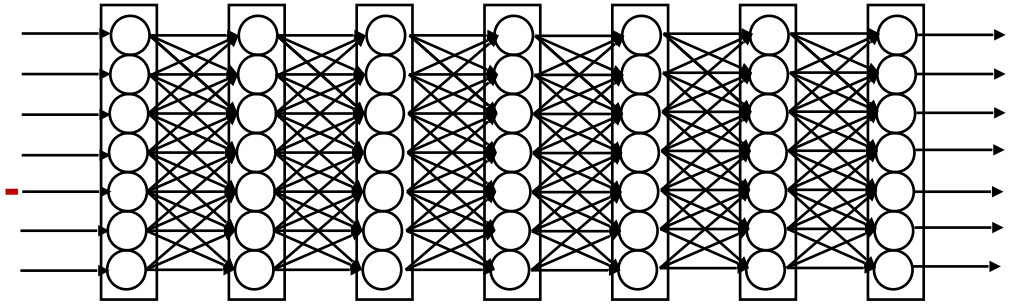
- Hyper Parameter Search
  - Searching optimal network configs & parameters
  - Parallel search, massive parallelism required
- Data Parallelism
  - Copy the network to compute nodes, feed different batch data, average => network reduction bound
  - TOFU: Extremely strong reduction, x6 EDR Infiniband

Inter-Node

- Model Parallelism (domain decomposition)
  - Split and parallelize the layer calculations in propagation
  - Low latency required (bad for GPU) -> strong latency tolerant cores + low latency TOFU network



- Intra-Chip ILP, Vector and other low level Parallelism
  - Parallelize the convolution operations etc.
  - SVE FP16+INT8 vectorization support + extremely high memory bandwidth w/ HBM2



- Post-K could become world's biggest & fastest platform for DNN training!

Intra-Node

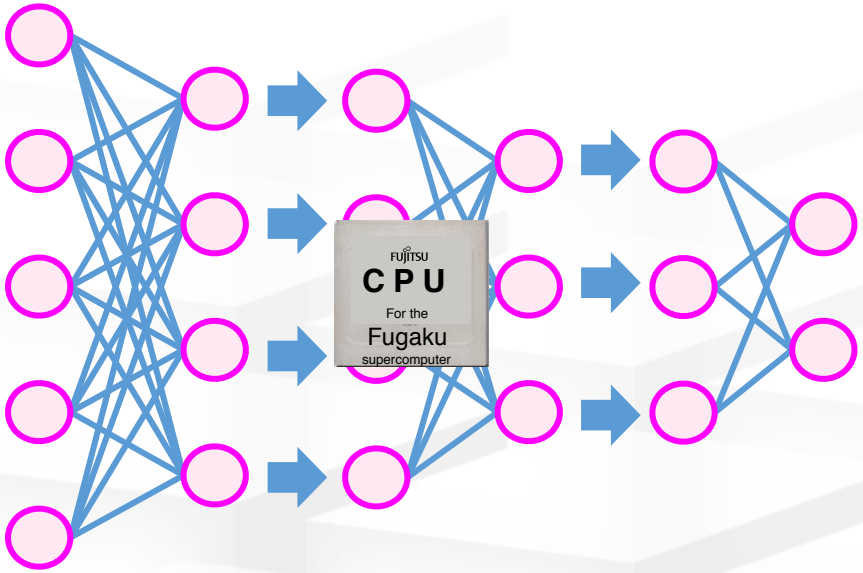
Massive amount of total parallelism, only possible via supercomputing

Massive Scale Deep Learning on Fugaku

Fugaku Processor

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

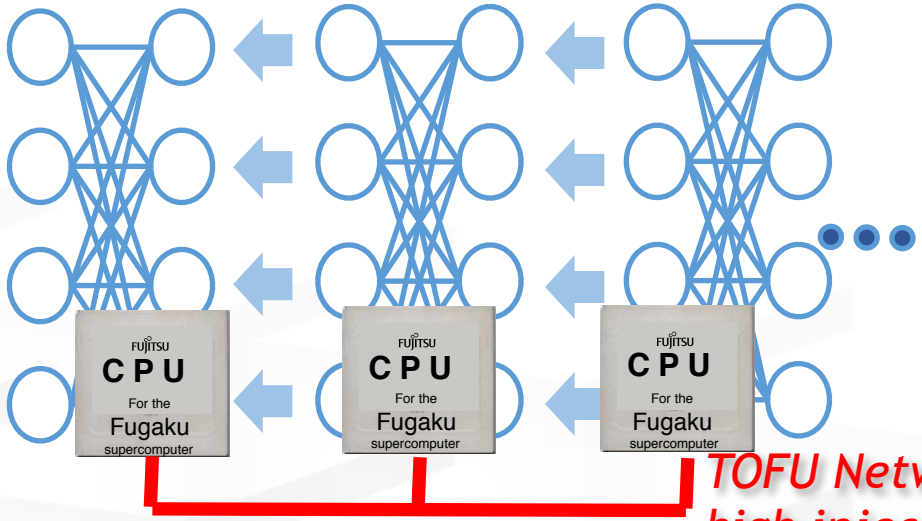
High Performance DNN Convolution



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

Unprecedened DL scalability

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

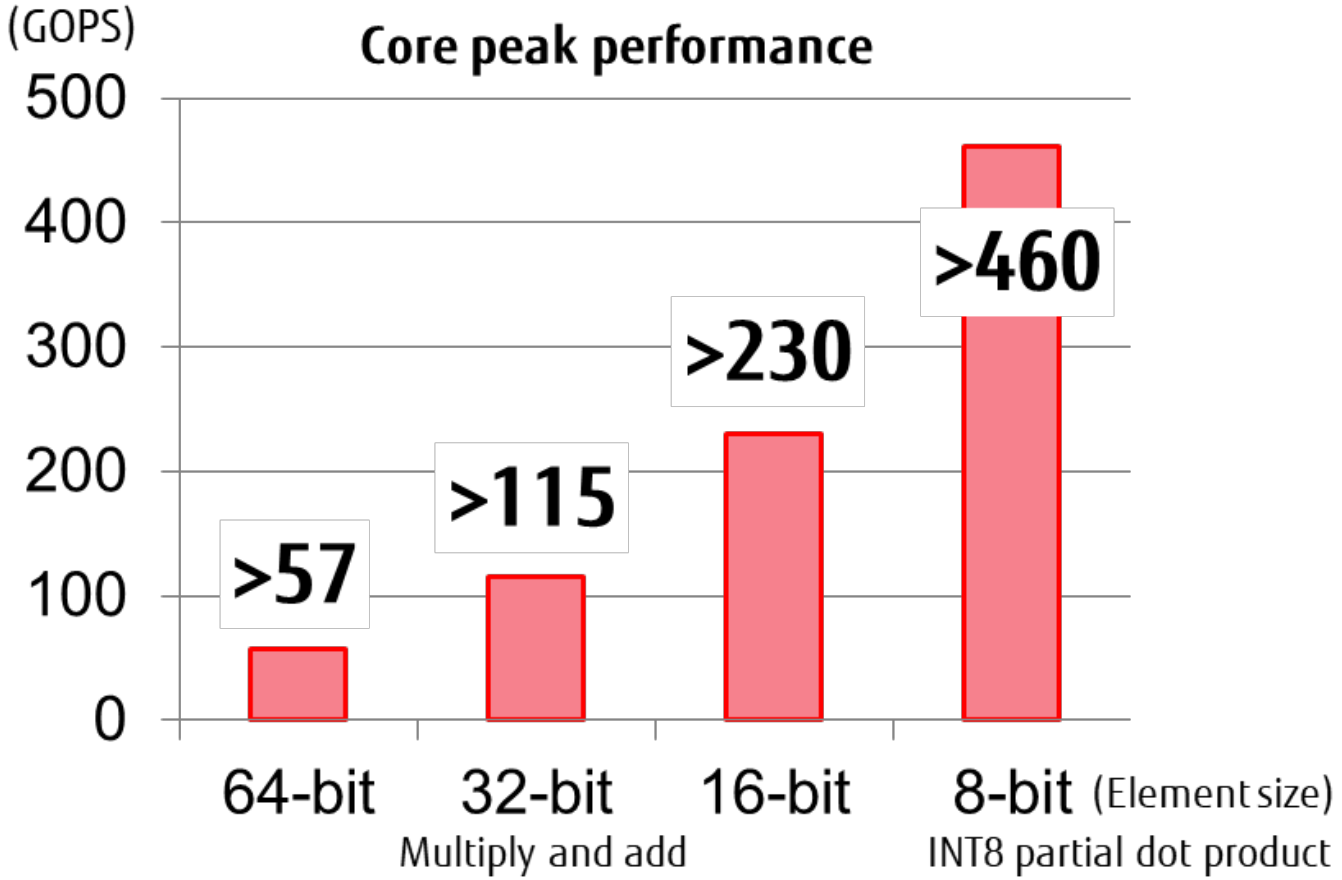


TOFU Network w/ high injection BW for fast reduction

Unprecedented Scalability of Data/

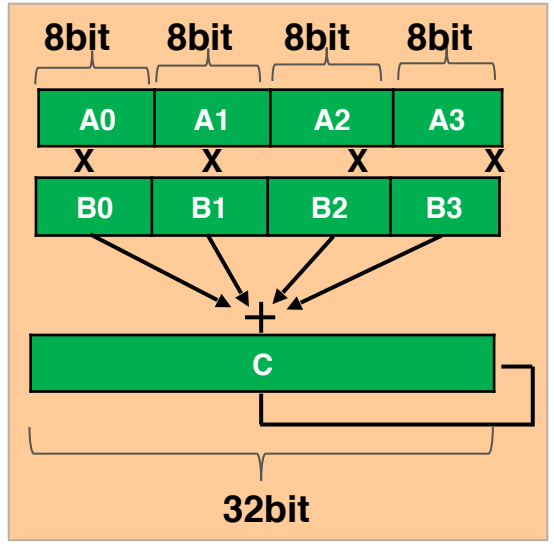
A64FX technologies: Core performance

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

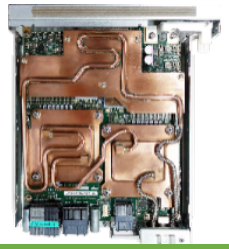


INT8 partial dot product

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



“Isopower” Comparsion with the Best GPU



	NVIDIA Volta v100	Fujitsu A64fx (2 A0 chip nodes)
Power	400 W (incl. CPUs, HCAs DGX-1)	“similar”
Vectorized MACC Formats	FP 64/32/16, INT 32(?)	FP 64/32/16, INT 32/16/8 w/INT32 MACC
Multi-node Linpack	5.9 TF / chip (DGX-1)	> 5.3 TF / 2 chip blade
Flops/W Linpack	15.1 GFlops/W (DGX-2)	> 15 Glops/W
Stream Triad	855 GB/s	1.68 TB / s
Memory Capacity	16 / 32 GB	64 GB (32 x 2)
AI Performance	125 (peak) / ~95 (measured) Tflops FP16 Tensor Cores	~48 TOPS (INT8 MACC peak)
Price	~$11,000 (SXM2 32GB board only) ~$13,000 (DGX-1, per 16GB GPU)	Talk to Fujitsu ☺

Large Scale Public AI Infrastructures in Japan

Inference 838.5PF
Training 86.9 PF

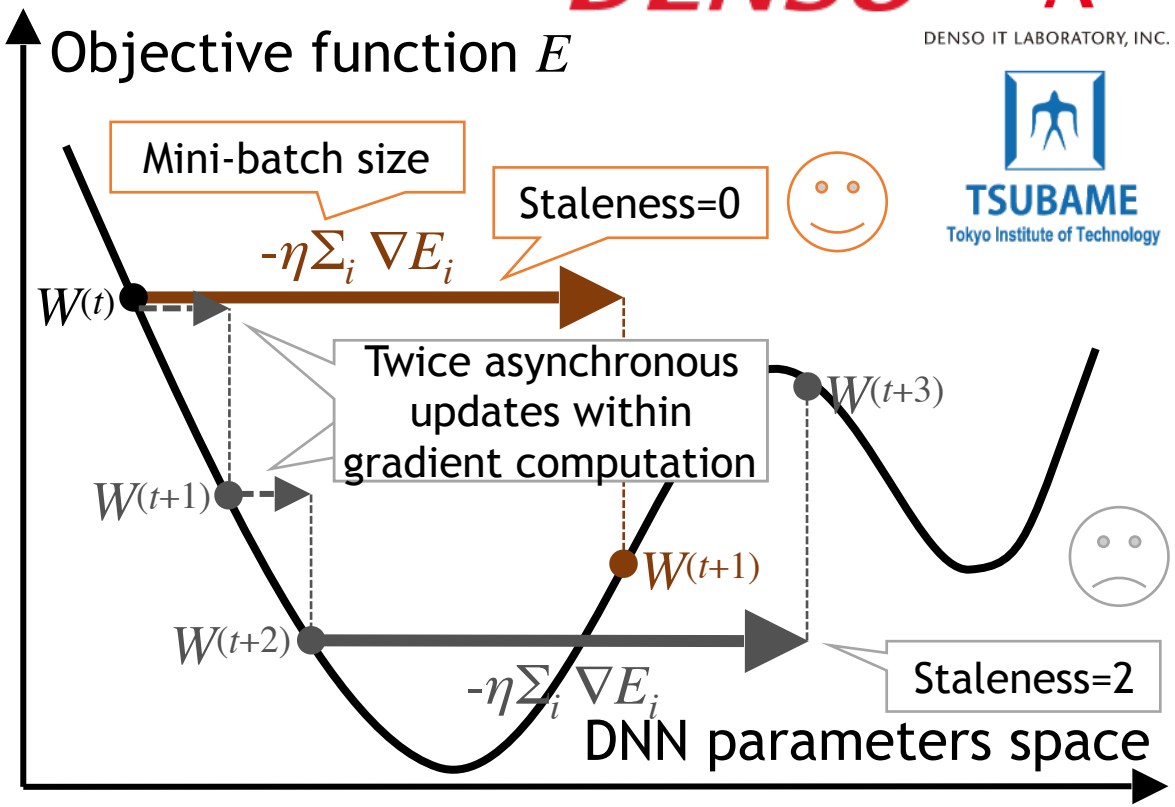
vs. Summit
Inf. 1/4
Train. 1/5

	Deployed	Purpose	AI Processor	Inference Peak Perf.	Training Peak Perf.	Top500 Perf/Rank	Green500 Perf/Rank
Tokyo Tech. TSUBAME3	July 2017	HPC + AI Public	NVIDIA P100 x 2160	45.8 PF (FP16)	22.9 PF / 45.8PF (FP32/FP16)	8.125 PF #22	13.704 GF/W #5
U-Tokyo Reedbush-H/L	Apr. 2018 (update)	HPC + AI Public	NVIDIA P100 x 496	10.71 PF (FP16)	5.36 PF / 10.71PF (FP32/FP16)	(Unranked)	(Unranked)
U-Kyushu ITO-B	Oct. 2017	HPC + AI Public	NVIDIA P100 x 512	11.1 PF (FP16)	5.53 PF/11.1 PF (FP32/FP16)	(Unranked)	(Unranked)
AIST-AIRC AICC	Oct. 2017	AI Lab Only	NVIDIA P100 x 400	8.64 PF (FP16)	4.32 PF / 8.64PF (FP32/FP16)	0.961 PF #446	12.681 GF/W #7
Riken-AIP Raiden	Apr. 2018 (update)	AI Lab Only	NVIDIA V100 x 432	54.0 PF (FP16)	6.40 PF/54.0 PF (FP32/FP16)	1.213 PF #280	11.363 GF/W #10
AIST-AIRC ABCI	Aug. 2018	AI Public	NVIDIA V100 x 4352	544.0 PF (FP16)	65.3 PF/544.0 PF (FP32/FP16)	19.88 PF #7	14.423 GF/W #4
NICT (unnamed)	Summer 2019	AI Lab Only	NVIDIA V100 x 1700程度	~210 PF (FP16)	~26 PF/~210 PF (FP32/FP16)	????	????
C.f. US ORNL Summit	Summer 2018	HPC + AI Public	NVIDIA V100 x 27,000	3,375 PF (FP16)	405 PF/3,375 PF (FP32/FP16)	143.5 PF #1	14.668 GF/W #3

Predicting Statistics of Asynchronous SGD Parameters for a Large-Scale Distributed Deep Learning System on GPU Supercomputers

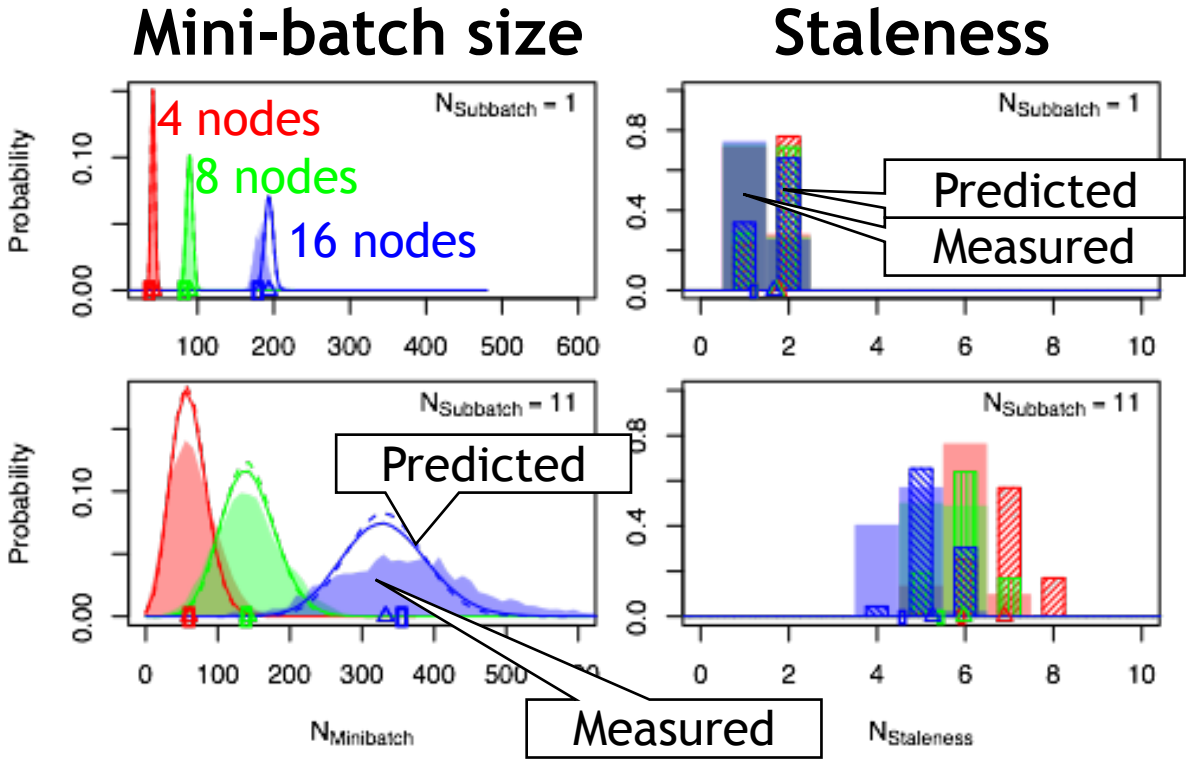
Background

- In large-scale Asynchronous Stochastic Gradient Descent (ASGD), mini-batch size and gradient staleness tend to be large and unpredictable, which increase the error of trained DNN



Proposal

We propose a empirical performance model for an ASGD deep learning system SPRINT which considers probability distribution of mini-batch size and staleness



($N_{Subbatch}$: # of samples per one GPU iteration)

- Yosuke Oyama, Akihiro Nomura, Ikuro Sato, Hiroki Nishimura, Yukimasa Tamatsu, and Satoshi Matsuoka, "**Predicting Statistics of Asynchronous SGD Parameters for a Large-Scale Distributed Deep Learning System on GPU Supercomputers**", in proceedings of 2016 IEEE International Conference on Big Data (IEEE BigData 2016), Washington

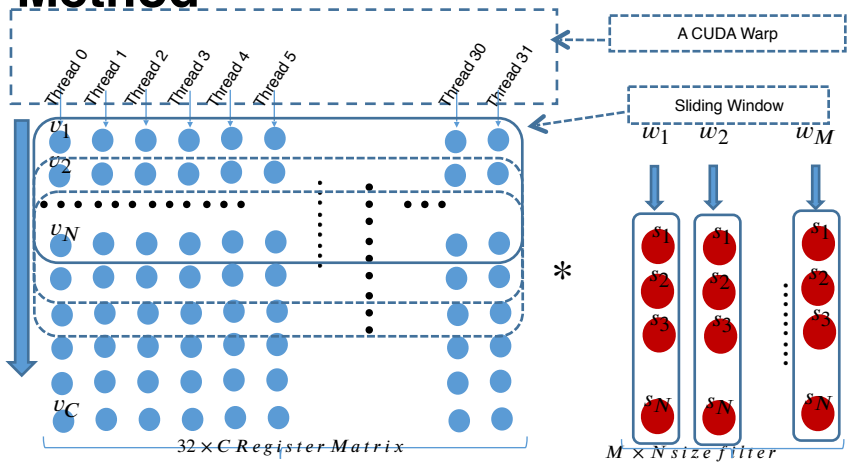
Pushing the Limits for 2D Convolution Computation On GPUs

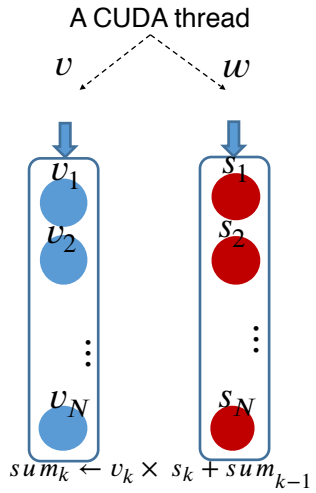
[To appear SC19]

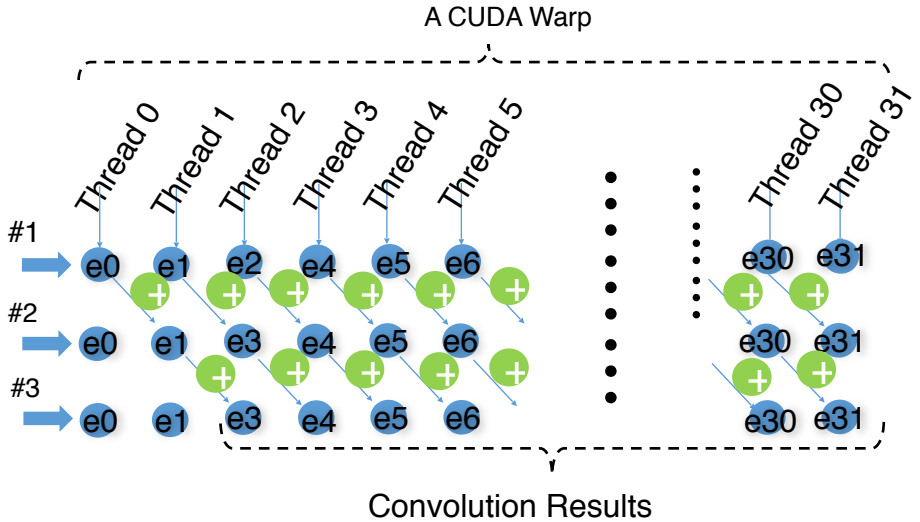
- **Background of 2D convolution**
  - Convolution on CUDA-enabled GPUs is essential for Deep Learning workload
  - A typical memory-bound problem with regular access

Also applicable to vector processor with shuffle ops, e.g. A64FX

- **Method**





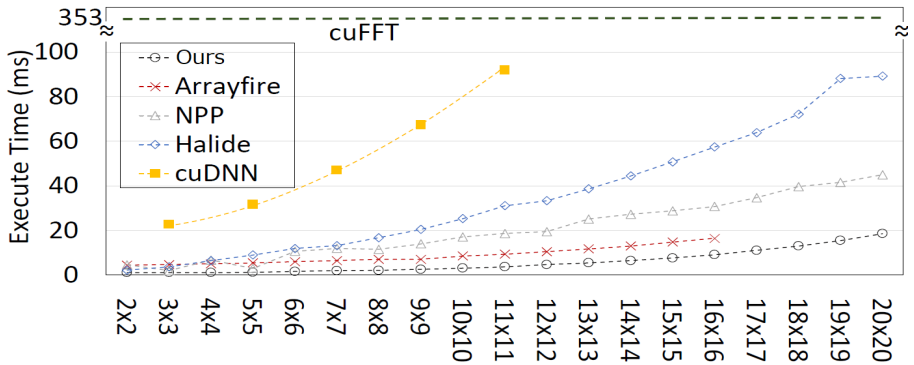


(1) Register Cache

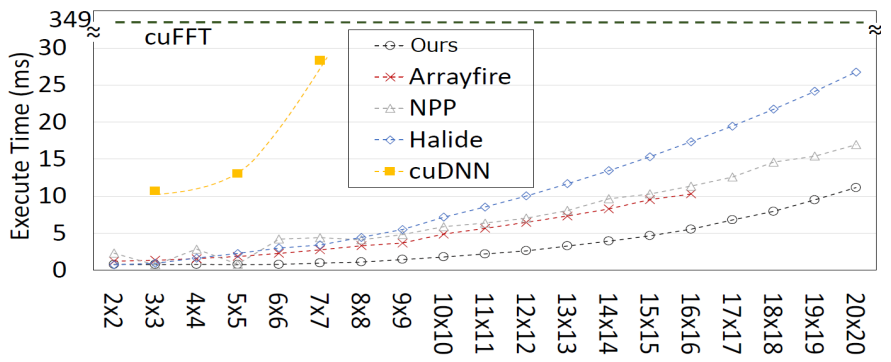
(2) Compute partial sums

(3) Transfer partial sums

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



Evaluation on Tesla **P100** GPU

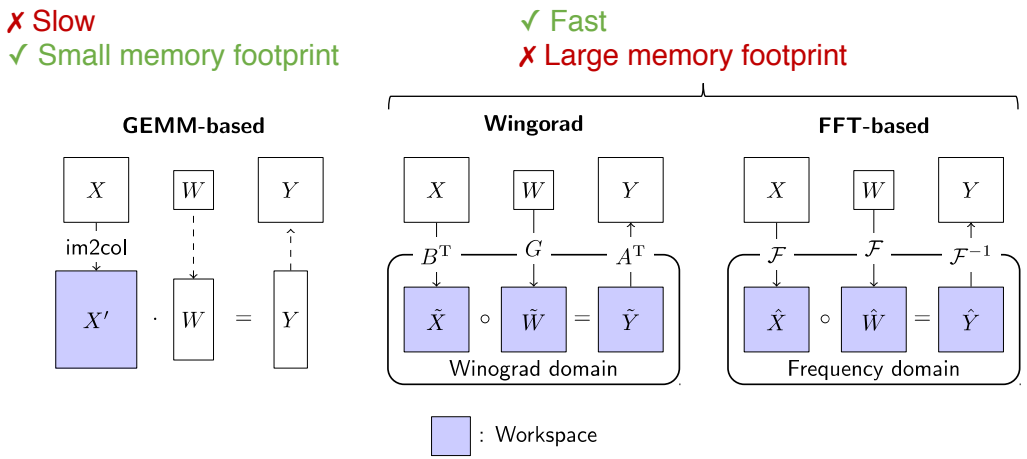


Evaluation on Tesla **V100** GPU

[1] Peng Chen, Mohamed Wahib, Shinichiro Takizawa, Satoshi Matsuoka. Pushing the Limits for 2D Convolution Computation On CUDA-enabled GPUs. 第163回ハイパフォーマンスコンピューティング研究会, Mar. 2018.

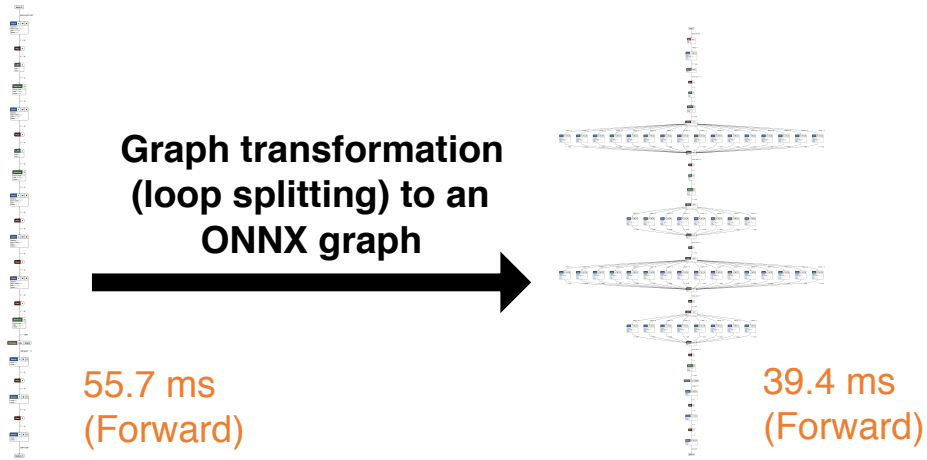
Applying Loop Transformations/Algorithm Optimizations to Deep Learning Kernels on cuDNN [1] and ONNX [2]

- **Motivation**: How can we use faster convolution algorithms (FFT and Winograd) with a small workspace memory for CNNs?
- **Proposal**: μ-cuDNN, a wrapper library for cuDNN, which applies loop splitting to convolution kernels based on DP and integer LP techniques
- **Results**: μ-cuDNN achieves significant speedups in multiple levels of deep learning workloads, achieving 1.73x of average speedups for DeepBench's 3×3 kernels and 1.45x of speedup for AlexNet on Tesla V100



Convolution algorithms supported by cuDNN

- **Motivation**: How can we extend μ-cuDNN to support arbitrary types of layers, frameworks and loop dimensions?
- **Proposal**: Apply graph transformations on the top of the ONNX (Open Neural Network eXchange) format
- **Results**: 1.41x of speedup for AlexNet on Chainer only with graph transformation and Squeezing 1.2x of average speedup for DeepBench's 3x3 kernels by multi-level splitting



AlexNet before/after the transformation

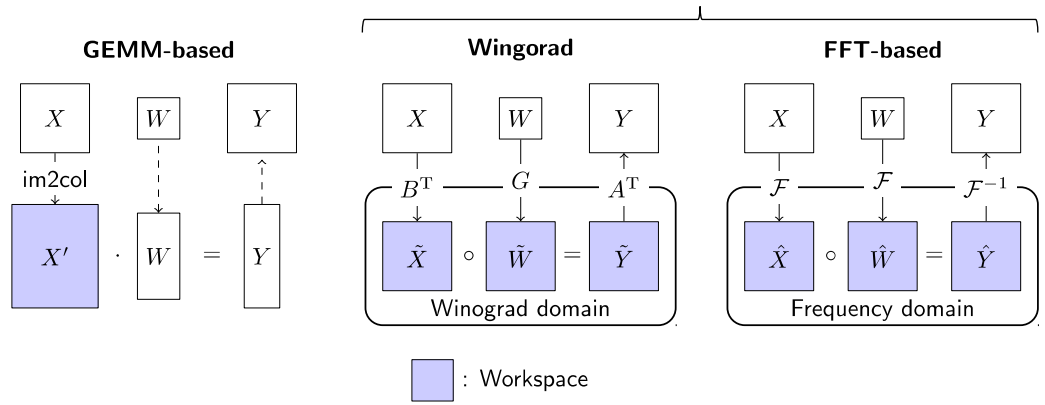
[1] Yosuke Oyama, Tal Ben-Nun, Torsten Hoefler, Satoshi Matsuoka, Accelerating Deep Learning Frameworks with Micro-batches, In proceedings of IEEE Cluster 2018, Belfast UK, Sep. 10-13, 2018.
[2] (To appear) Yosuke Oyama, Tal Ben-Nun, Torsten Hoefler, Satoshi Matsuoka, Applying Loop Transformations to Deep Neural Networks on ONNX, 情報処理学会研究報告, 2019-HPC-170. In 並列/分散/協調処理に関するサマーワークショップ (SWoPP2019), Jul. 24-26, 2019.

μ-cuDNN: Accelerating Deep Learning Frameworks with Micro-batches [1]

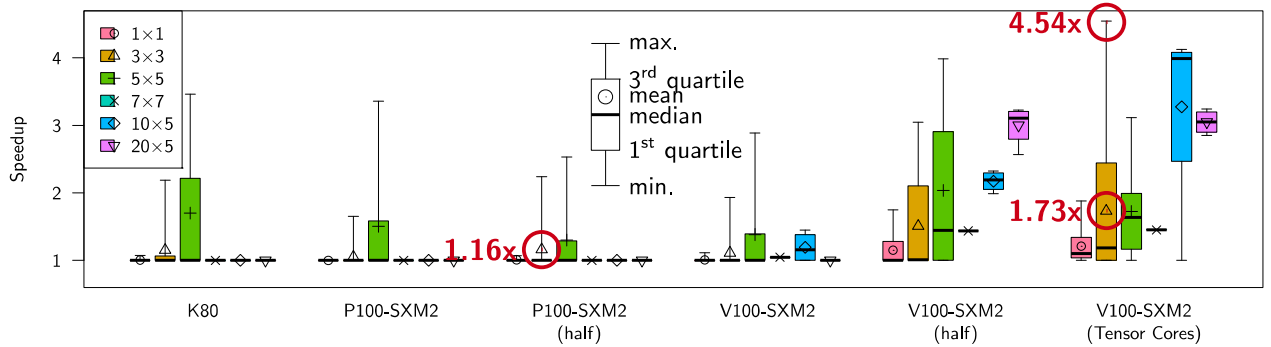
- **Motivation**: How can we use faster convolution algorithms (ex. FFT and Winograd) with a small workspace memory for Convolutional Neural Networks (CNNs)?
- **Proposal**: μ-cuDNN, a wrapper library for the math kernel library cuDNN which is applicable for most deep learning frameworks
  - μ-cuDNN applies loop splitting by using dynamic programming and integer linear programming techniques
- **Results**: μ-cuDNN achieves significant speedups in multiple levels of deep learning workloads
  - 1.16x, 1.73x of average speedups for DeepBench's 3×3 kernels on Tesla P100 and V100 respectively
  - achieves 1.45x of speedup (1.60x w.r.t. convolutions alone) for AlexNet on V100

✗ Slow
✓ Small memory footprint

✓ Fast
✗ Large memory footprint



Convolution algorithms supported by cuDNN



Relative speedups of DeepBench's forward convolution layers against cuDNN

[1] Yosuke Oyama, Tal Ben-Nun, Torsten Hoefler, Satoshi Matsuoka, Accelerating Deep Learning Frameworks with Micro-batches, In proceedings of IEEE Cluster 2018, Belfast UK, Sep. 10-13, 2018.

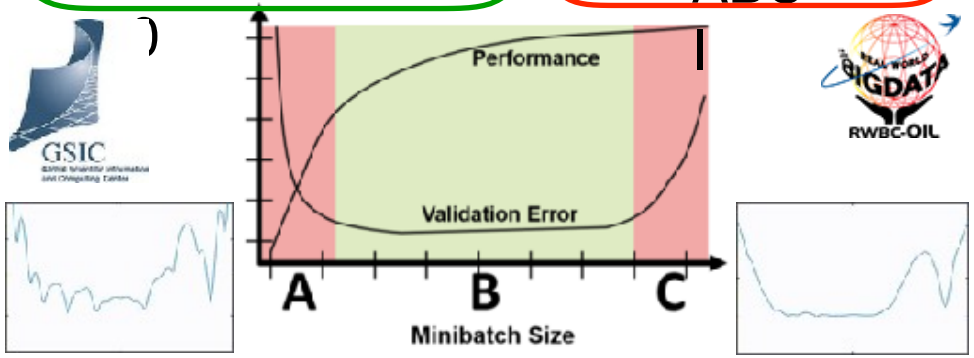
Training ImageNet in Minutes

Rio Yokota, Kazuki Osawa, Yohei Tsuji, Yuichiro Ueno, Hiroki Naganuma, Shun Iwase, Kaku Linsho, Satoshi Matsuoka Tokyo Institute of Technology/Riken

+Akira Naruse (NVIDIA)

TSUBAME3.

ABC

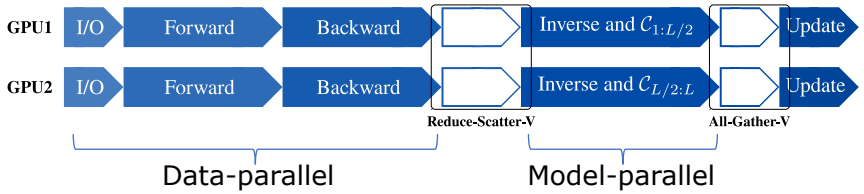


Source Ben-nun & Hoefler https://arxiv.org/pdf/

	#GPU	time
Facebook	512	30 min
Preferred Networks	1024	15 min
UC Berkeley	2048	14 min
Tencent	2048	6.6 min
Sony (ABCI)	~3000	3.7 min
Google (TPU/GCC)	1024	2.2 min
TokyoTech/NVIDIA/Riken (ABCI)	4096	? min

Accelerating DL with 2nd Order Optimization and Distributed Training [Tsuji et al.] => *Towards 100,000 nodes scalability*

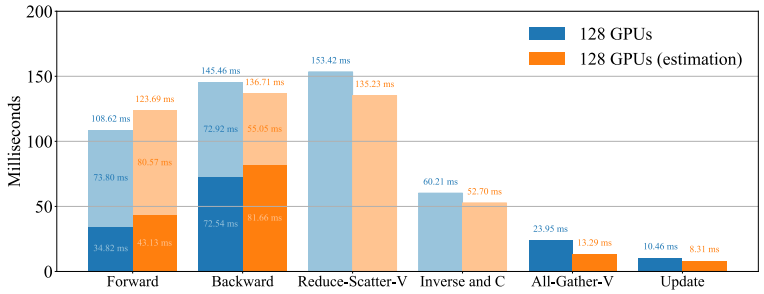
- Background
  - Large complexity of DL training.
  - Limits of data-parallel distributed training.
  - *> How to accelerate the training further?*
- Method
  - Integration of two techniques: 1) data- and model-parallel distributed training, and 2) K-FAC, an approx 2nd order optimization.
- Evaluation and Analysis
  - Experiments on ABCI supercomputer.
  - Up to 128K batch size w/o accuracy degradation.
  - Finish training in 35 epochs/10 min/ 1024 GPUs in 32K batch size.
  - A performance tuning / modeling.



Design our hybrid parallel distributed K-FAC

	Batch size	# Iterations	Accuracy
Goyal et al.	8K	14076	76.3%
Akiba et al.	32K	3519	75.4%
Ying et al.	64K	1760	75.2%
Ours	**128K**	**978**	**75.0%**

Comparison with related work (ImageNet/ResNet-50)



Time prediction with the performance model

Osawa et al., Large-Scale Distributed Second-Order Optimization Using Kronecker-Factored Approximate Curvature for Deep Convolutional Neural Networks, CVPR 2019

Fast ImageNet Training

2018/12/2 Assert

Top 10 Arxiv Papers Today in Computer Science

2.06 Mikeys

#1. Second-order Optimization Method for Large Mini-batch: Training ResNet-50 on ImageNet in 35 Epochs

Kazuki Osawa, Yohei Tsuji, Yuichiro Ueno, Akira Naruse, Rio Yokota, Satoshi Matsuoka

Large-scale distributed training of deep neural networks suffer from the generalization gap caused by the increase in the effective mini-batch size. Previous approaches try to solve this problem by varying the learning rate and batch size over epochs and layers, or some ad hoc modification of the batch normalization. We propose an alternative approach using a second-order optimization method that shows similar generalization capability to first-order methods, but converges faster and can handle larger mini-batches. To test our method on a benchmark where highly optimized first-order methods are available as references, we train ResNet-50 on ImageNet. We converged to 75% Top-1 validation accuracy in 35 epochs for mini-batch sizes under 16,384, and achieved 75% even with a mini-batch size of 131,072, which took 100 epochs.

more | pdf | html

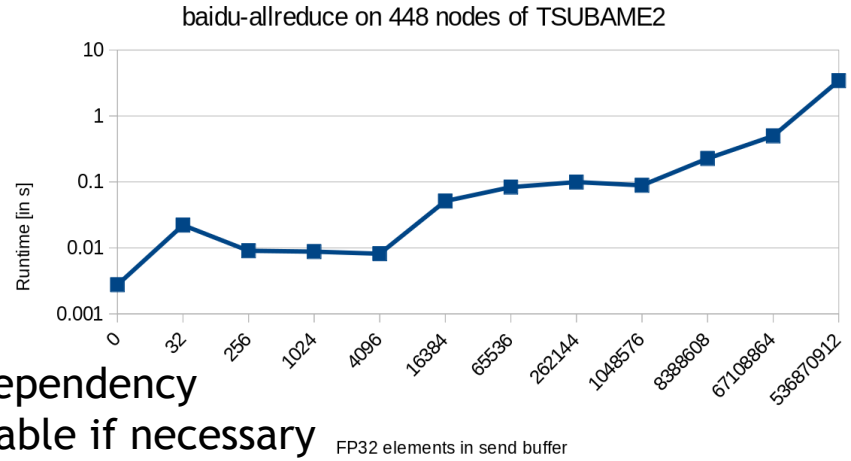
Figures

None.

Our measurements (following DeepBench specs) for Interconnect on Tsubame 2.5 and K computer

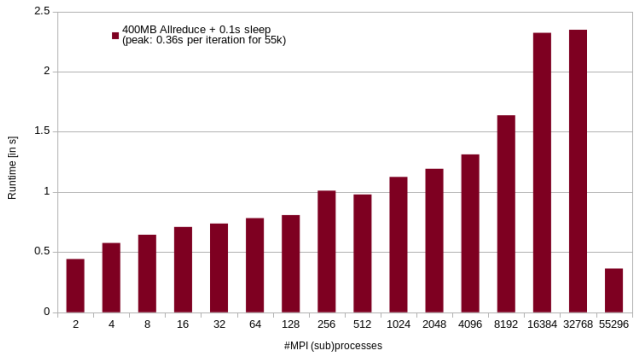
Baidu's Allreduce Benchmark

- hardcoded steps: 0B→2GiB
- hardcoded #iterations per msg size
- GPU/Cuda-dependency easily removable if necessary



Our "sleepy-allreduce" (modified Intel IMB)

- emulated DL training
- alternating 400MiB Allreduce and 0.1s sleep for compute



Nvidia's Collective Comm. Library (NCCL) Tests

- benchmark GPU collectives for DL frameworks which use NCCCL as backend
- Example visualization:

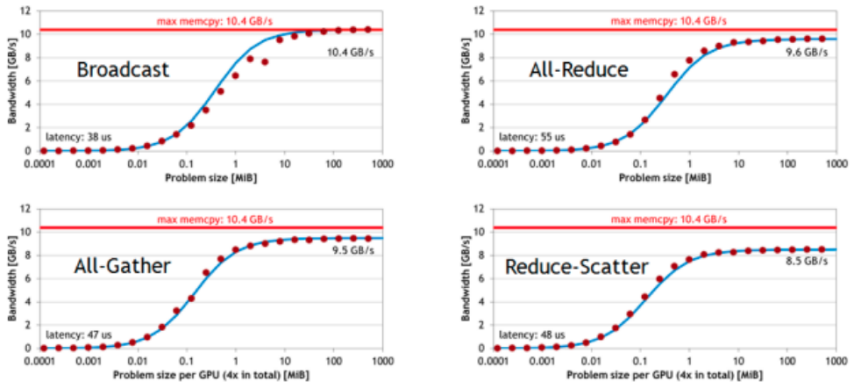


Fig. from Nvidia Devblog

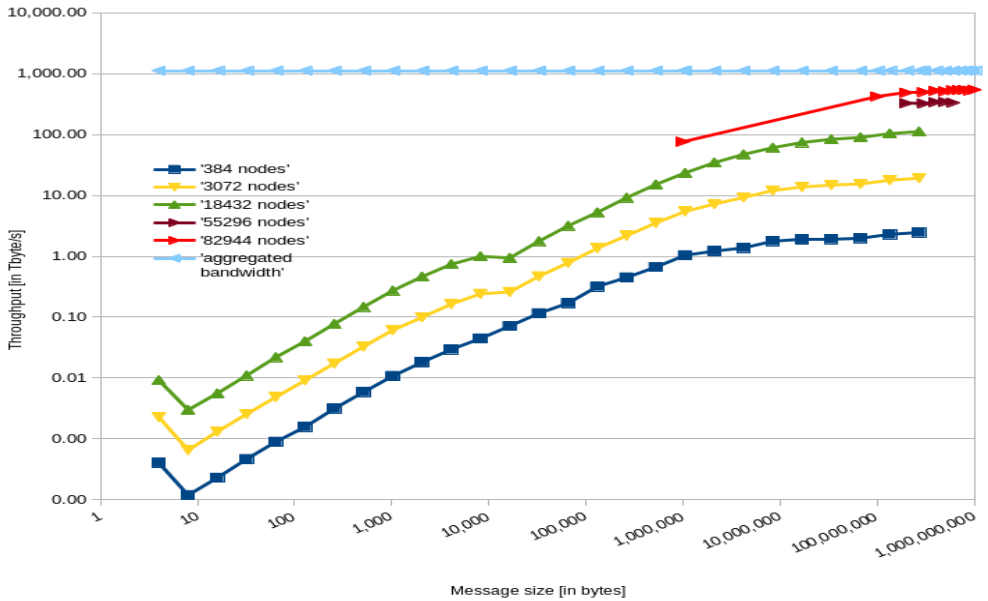
Others:

- Tensorflow's allreduce benchmark (see Tf_cnn_benchmarks for details; needs very recent TF)
- PFN has benchmark/data for ChainerMN / PFN-Proto (see their blogpost; unknown if open-source)

Common/Generic Interconnect Benchmarks

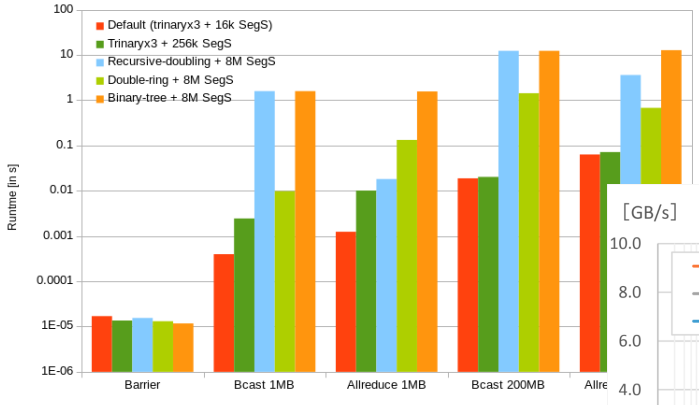
Intel MPI Benchmarks (IMB)

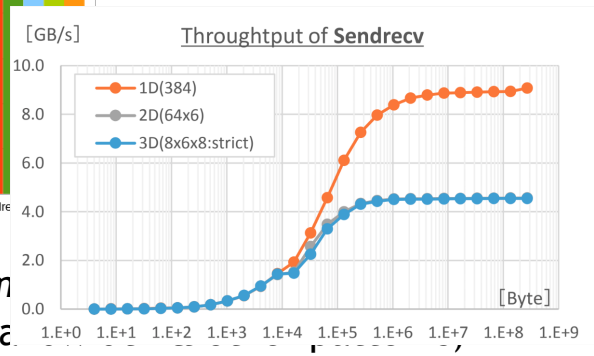
- IMB and OSU benchmarks very similar
- testing many P2P, collectives, MPI-I/O functions
- Default comm. size range from 0B→4MiB (power-2 steps; can be modified manually)
- MPI-Allreduce example for K:



OSU Micro-Benchmarks (from Ohio-State Univ)

- MPI collectives relevant for DL training + p2p BMs:





Many other, less common MPI (m

- Netgauge (for eff. bisection ba
- BenchIT suite (incl. MPI BMs; more fine-granular steps)
- SPEC MPI2007 (more application-centric)
- etc.

Optimizing Collective Communication in DL Training (1 of 3)

- Reducing **training time** of large-scale AI/DL on GPUs-system.
  - Time for inference = O(seconds)
  - **Time for training = O(hours or days)**
- Computation is one of the bottleneck factors
  - Increasing the batch size and learning in **parallel**
    - Training ImageNet in 1 hour [1]
    - Training ImageNet in ~20 minutes [2]
- Communication also can become a bottleneck
  - Due to **large message sizes**

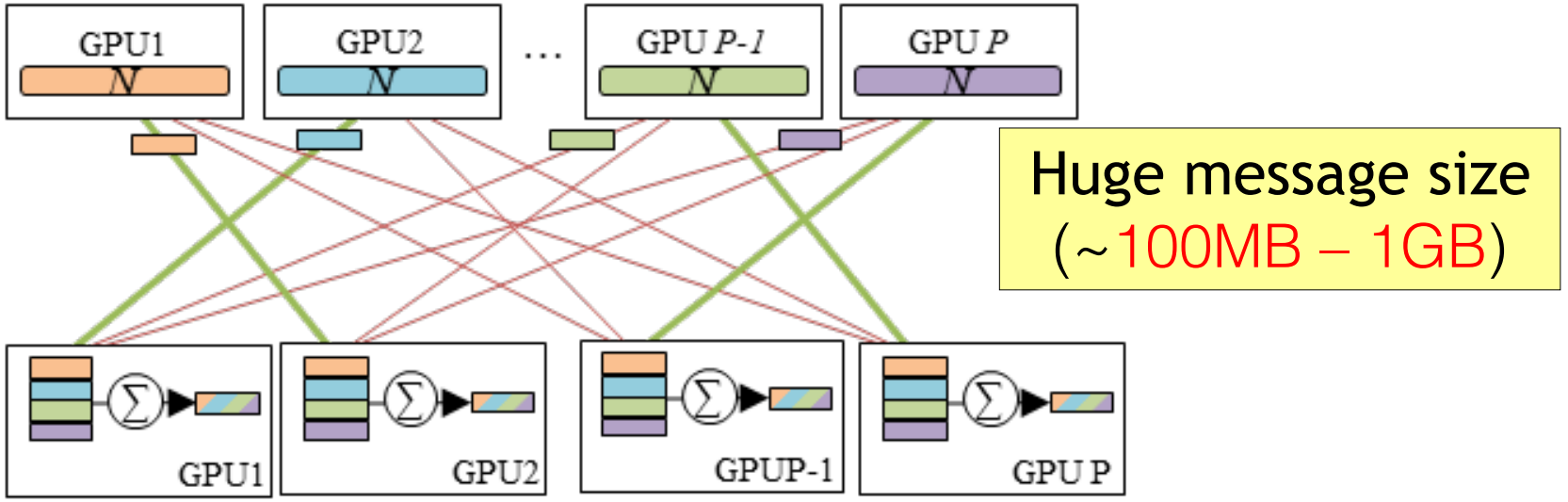
[1] P. Goyal, P. Doll´ar, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch SGD: training imagenet in 1 hour," arXiv preprint arXiv:1706.02677, 2017.
[2] Y. You, Z. Zhang, C. Hsieh, J. Demmel, and K. Keutzer, "Imagenet training in minutes," CoRR, abs/

Optimizing Collective Communication in DL Training (2 of 3)
(Challenges of Large Message Size)

Compute the gradient G_i of the weight on each GPUi

GPUs communication to compute the mean of the gradients (Allreduce operation)

$$G = \sum_{i=1}^{P} Gi$$



Huge message size (~100MB – 1GB)

Example of Image Classification, ImageNet data set

Model	AlexNet (2012)	GoogleNet (2015)	ResNet (2016)	DenseNet (2017)
# of gradients [1]	61M	5.5M	1.7 - 60.2M	15.3 - 30M
Message size	244 MB	22MB	240 MB	120 MB

[1] T. Ben-Nun and T. Hoefler, "Demystifying parallel and distributed deep learning: An in-depth concurrency analysis," arXiv preprint arXiv:1802.09941, 2018.

Optimizing Collective Communication in DL Training (3 of 3)

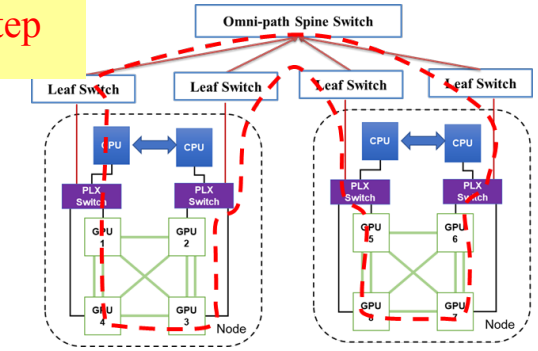
Proposal: Separate intra-node and inter-node comm. ➔ **multileader hierarchical algorithm**

- ➢ Phase 1: Intra-node reduce to the node leader
- ➢ Phase 2: Inter-node all-reduce between leaders
- ➢ Phase 3: Intra-node broadcast from the leaders

Key Results:

- ➢ Cut down the communication time up to **51%**
- ➢ Reduce the power consumption up to **32%**

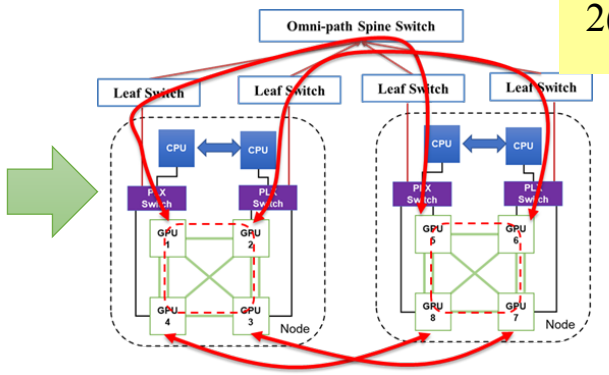
$2(P\text{-}1)$ steps, send $\frac{N}{P}$ per step



Ring-based algorithm

- Good for large message size
- Worse with inter-node comm.

$2(\frac{P}{k}\text{ -1})$ steps, $\frac{N\,(p-k)}{P\,k}$ per step

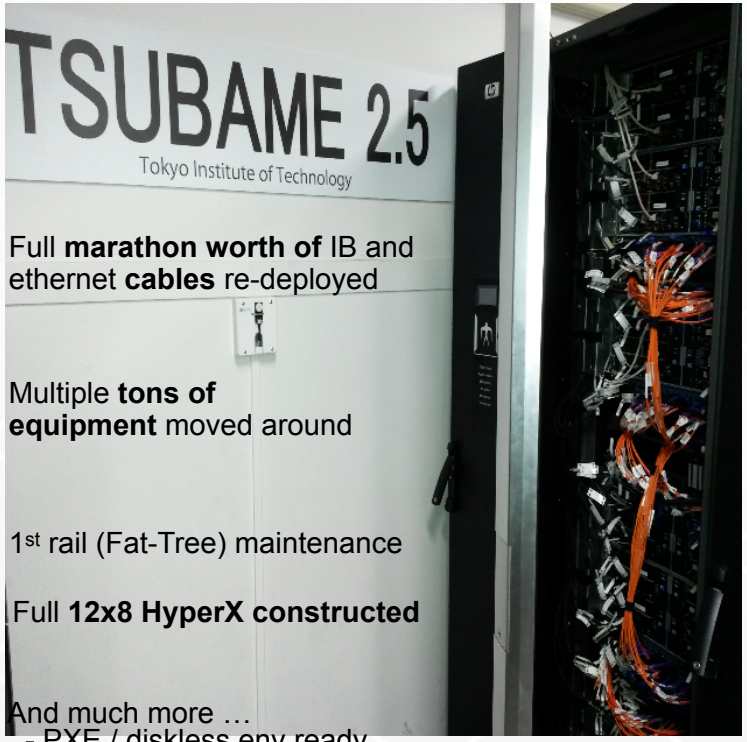


Multileader hierarchical algorithm

- Optimized for inter-node comm.

"Efficient MPI-Allreduce for Large-Scale Deep Learning on GPU-Clusters", Truong Thao Nguyen, Mohamed Wahib, Ryousei Takano, Journal of Concurrency and Computation: Practice and Experience (CCPE) , Accepted: to appear in 2019.10

1st large-scale Prototype – Motivation for HyperX



Full **marathon worth of** IB and ethernet **cables** re-deployed

Multiple **tons of equipment** moved around

1st rail (Fat-Tree) maintenance

Full **12x8 HyperX constructed**

And much more …
- PXE / diskless env ready
- Spare AOC under the floor
- BIOS batteries exchanged

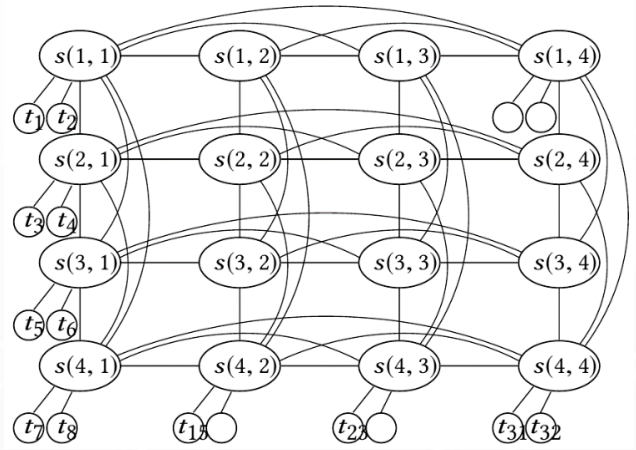


Fig.1: HyperX with n-dim. integer lattice ($d_1,\ldots,d_n$) base structure fully connected in each dim.

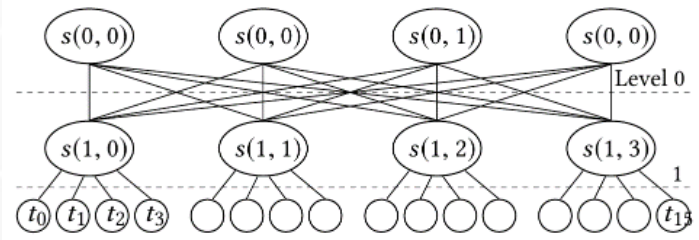
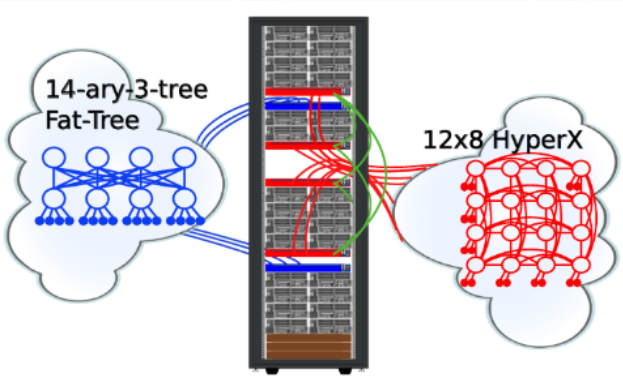


Fig.2: Indirect 2-level Fat-Tree

TokyTech's 2D HyperX:

- **24 racks** (of 42 T2 racks)
- 96 QDR switches (+ 1st rail) **without adaptive routing**
- 1536 IB cables (720 AOC)
- **672 compute nodes**
- **57% bisection bandwidth**



➔ **First large-scale 2.7 Pflop/s (DP) HyperX installation in the world!**

Theoretical Advantages (over Fat-Tree)

- **Reduced HW cost** (less AOC / SW)
- **Only needs 50% bisection BW**
- **Lower latency** (less hops)
- **Fits** rack-based **packaging**

Evaluating the HyperX and Summary

1:1 comparison (as fair as possible) of 672-node 3-level Fat-Tree and 12x8 2D HyperX

- NICs of 1st and 2nd rail even on same CPU socket
- Given our HW limitations (few "bad" links disabled)

Wide variety of benchmarks and configurations

- 3x Pure MPI benchmarks
- 9x HPC proxy-apps
- 3x Top500 benchmarks
- 4x routing algorithms (incl. PARX)
- 3x rank-2-node mappings
- 2x execution modes

Primary research questions

Q1: Will reduced bisection BW (57% for HX vs. ≥100% for FT) impede performance?

Q2: Two mitigation strategies against lack of AR? (→ e.g. placement vs. "smart" routing)

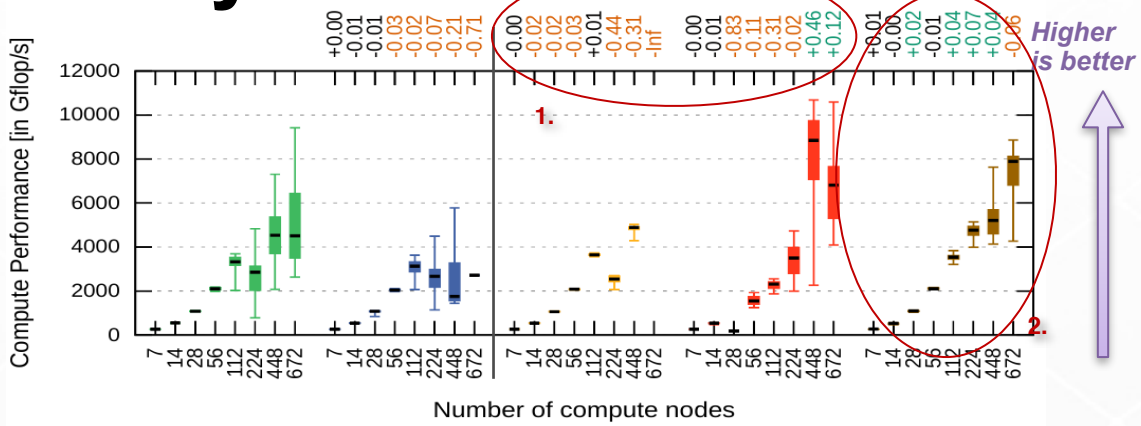


Fig.3: HPL (1GB pp, and 1ppn); scaled 7→ 672 cn

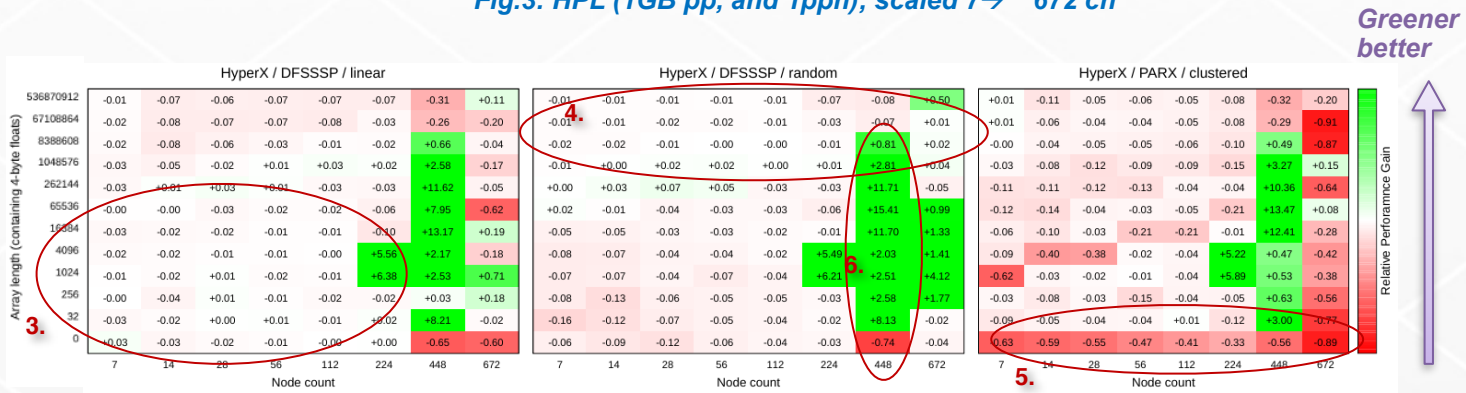


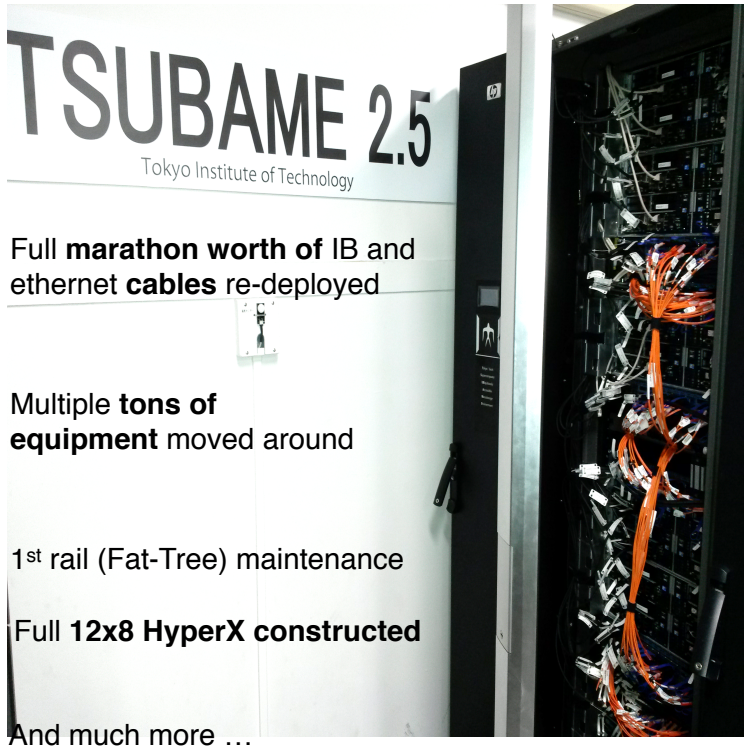
Fig.4: Baidu's (DeepBench) Allreduce (4-byte float) scaled 7→ 672 cn (vs. "Fat-tree / ftree / linear" baseline)

1. Placement mitigation can alleviate bottleneck
2. HyperX w/ PARX routing outperforms FT in HPL
3. *Linear* good for small node counts/msg. size
4. *Random* good for DL-relevant msg. size (± 1%)
5. "Smart" routing suffered SW stack issues
6. FT + ftree had bad 448-node corner case

Conclusion

HyperX topology is promising and cheaper alternative to Fat-Trees (even w/o adaptive R) !

Evaluating the HyperX Topology: A Compelling Alternative to Fat-Trees?[SC19]



Full **marathon worth of** IB and ethernet **cables** re-deployed

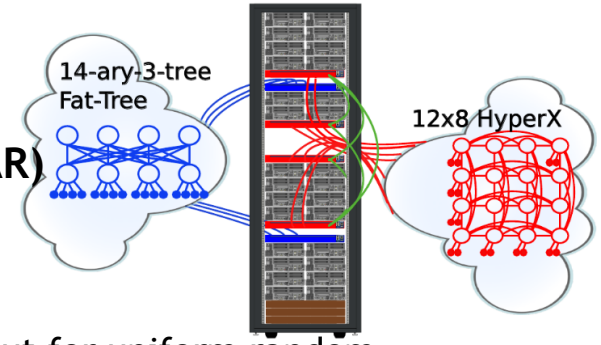
Multiple **tons of equipment** moved around

1st rail (Fat-Tree) maintenance

Full **12x8 HyperX constructed**

And much more …
- PXE / diskless env ready
- Spare AOC under the floor
- BIOS batteries exchanged

1:1 comparison (as fair as possible) of 672-node 3-level Fat-Tree and 12x8 2D HyperX
- NICs of 1st and 2nd rail even on same CPU socket
- Given our HW limitations (few "bad" links disabled)



Advantages (over FT) assuming **adaptive routing (AR)**
- **Reduced HW cost** (AOC/switches) → similar perf.
- **Lower latency** when scaling up (less hops)
- **Fits** rack-based **packaging** model for HPC/racks
- **Only needs 50% bisection BW** to provide 100% throughput for uniform random

Q1: Will reduced bisection BW (57% for HX vs. ≥100%) impede Allreduce perform
Q2: Mit
routing

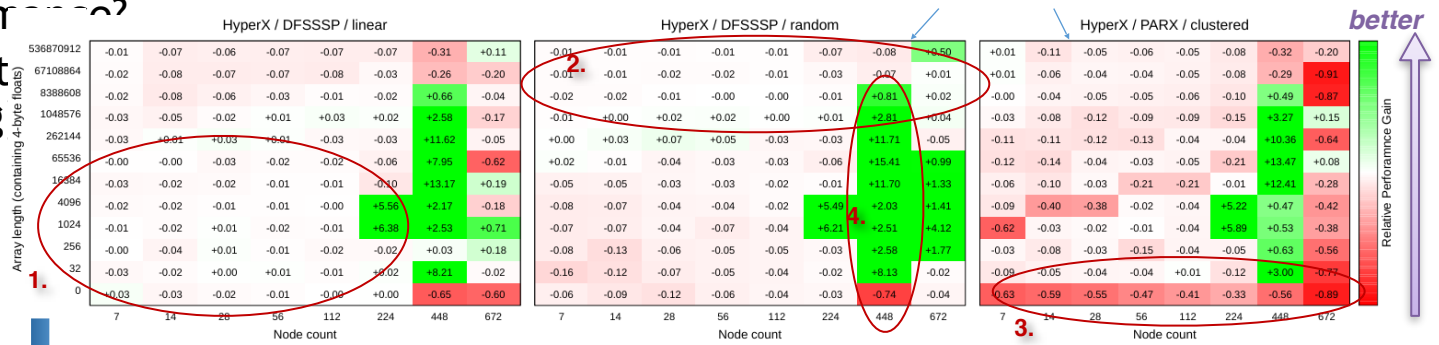


Fig.2: Baidu's (DeepBench) Allreduce (4-byte float) scaled 7→672 cn (vs. "Fat-tree / ftree / linear" baseline)

1. *Linear* good for small node counts/msg. size
2. *Random* good for DL-relevant msg. size (+ / —)
3. Smart routing suffered SW stack issues
4. FT + ftree had bad 448-node corner case

HyperX topology is promising and cheaper alternative to state-of-the-art Fat-Tree networks!

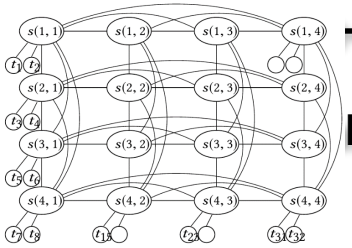


Fig.1: HyperX with n-dim. integer lattice ($d_1,\ldots,d_n$) base structure fully connected in each dim.

-scale 2.7 Pflop/s
llation in the

Our 2D HyperX:
- **24 racks** (of 42 T2 racks)
- 96 QDR switches (+ 1st rail)
- 1536 IB cables (720 AOC)
- **672 compute nodes**
- **57% bisection bandwidth**

Funded by and in collaboration with Hewlett Packard Enterprise, and supported by Fujitsu, JSPS KAKENHI, and JSP CREST

[1] Domke et al. "HyperX Topology: First at-scale Implementation and Comparison to the Fat-Tree" to be presented at SC'19 and HOTI'19

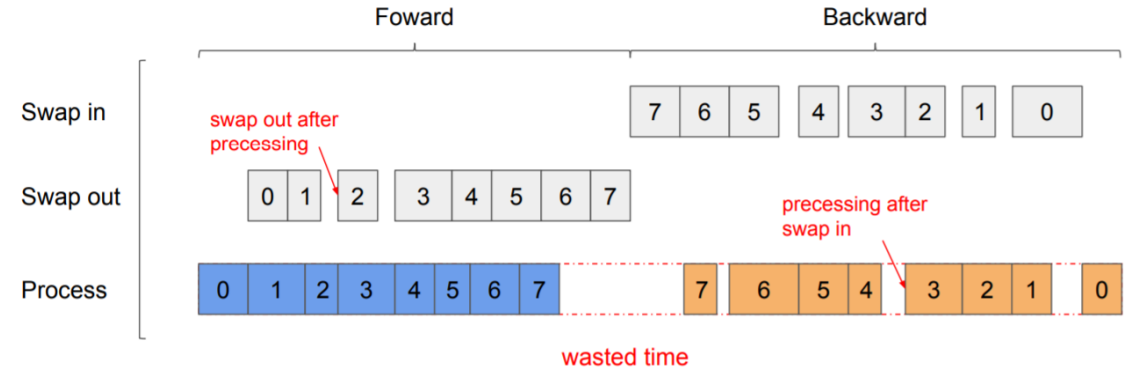
Breaking the limitation of GPU memory for Deep Learning

Haoyu Zhang,Wahib Mohamed, Lingqi Zhang,Yohei Tsuji, Satoshi Matsuoka

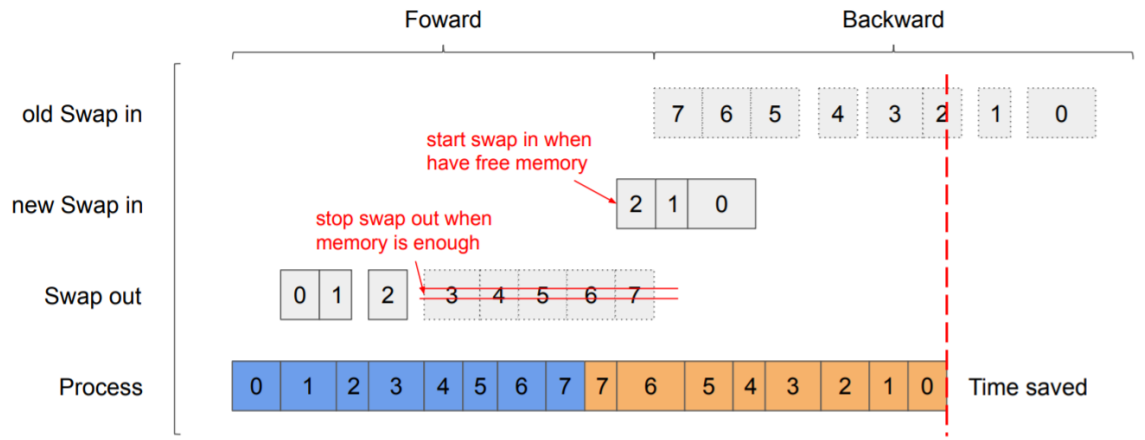
Motivation: GPU memory is relatively small in comparison to recent DL work load

Analysis:

$$
\begin{aligned}
Util &= T_{Proc}/T \\
&= \frac{Num\_buffer(l)/Proc_{Th}(l)}{max(Num\_buffer(l)/Proc_{Th}(l), Num\_buffer(l-1)/Swap\_in_{Th})}
\end{aligned}
$$



$$
Util = \begin{cases} \dfrac{Num\_buffer(l)/Proc_{Th}(l)}{max(Num\_buffer(l)/Proc_{Th}(l), Num\_buffer(l-1)/Swap\_in_{Th})}, \\ 1, \qquad T < \dfrac{\sum_{i=1}^{Catch}(Proc_{th}(i) \times T_{proc}(i))}{Swap\_in_{Th}} \end{cases}
$$

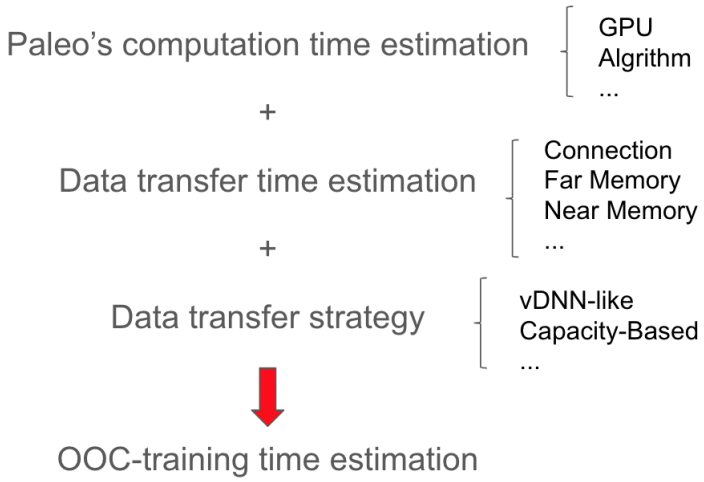


Breaking the limitation of GPU memory for Deep Learning

Haoyu Zhang,Wahib Mohamed, Lingqi Zhang,Yohei Tsuji, Satoshi Matsuoka

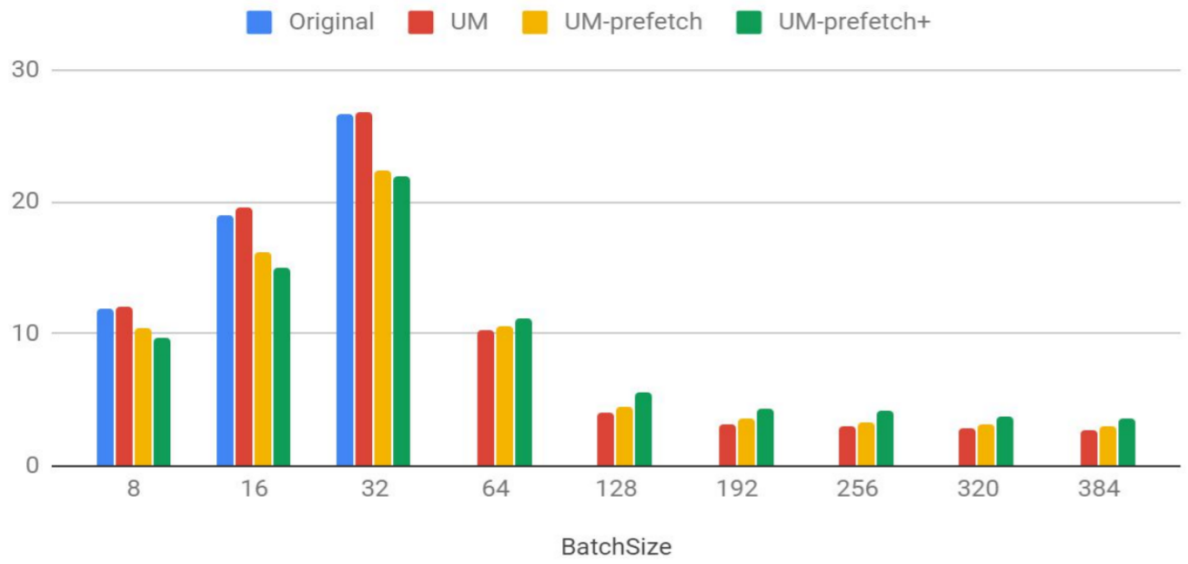
Proposal：

OOC-Paleo



UM-Chainer

prefetch()->explicit swap-in
no explicit swap-out



Case Study & Discussion:

Memory Capacity:

- Not so important as latency and throughput

Latency:

- Higher Bandwidth make no sense when buffer is too small
- Latency is decided by physical law

Bandwidth:

- Higher connection bandwidth
- Lower Memory bandwidth

Processor:

- Slower processor is acceptable

Breaking the limitation of GPU memory for Deep Learning

Haoyu Zhang,Wahib Mohamed, Lingqi Zhang,Yohei Tsuji, Satoshi Matsuoka

Assuming we have higher Bandwidth...

Resnet50,Batch-size=128

16GB/s->64GB/s:
Training time can be half

64GB/s->128GB/s:
Only a little time reduced

>128GB/s:
Most of the layers can not make full use of the bandwidth

>512GB/s:
Time almost do not decrease

Bandwidth	Time	percentage (bandwidth not full)	percentage (computation can not overlap)
16	967.8595572	0.409	0.733
32	569.9550342	0.466	0.642
64	407.2978908	0.574	0.472
128	371.9318064	0.688	0.438
256	362.5661138	0.835	0.398
512	359.7637498	0.915	0.398
1024	359.3012901	0.983	0.386
∞	359.3012901	1.000	0.386
Original version	306.9286403	N/A	N/A

Toward Training a Large 3D Cosmological CNN with Hybrid Parallelization

[1] Tokyo Institute of Technology, [2] Lawrence Livermore National Laboratory, [3] University of Illinois at Urbana-Champaign, [4] Lawrence Berkeley National Laboratory, [5] RIKEN Center for Computational Science, * oyama.y.aa@m.titech.ac.jp

August 5, 2019

The 1st Workshop on Parallel and Distributed Machine Learning 2019 (PDML'19) – Kyoto, Japan

Yosuke Oyama [1, 2]*, Naoya Maruyama [2], Nikoli Dryden [3, 2], Peter Harrington [4], Jan Balewski [4], Satoshi Matsuoka [5, 1], Marc Snir [3], Peter Nugent [4], and Brian Van Essen [2]

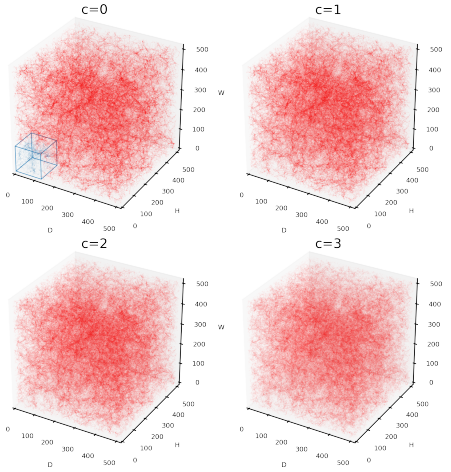
LLNL-PRES-XXXXXX

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Background

- **CosmoFlow** [1] is a project to estimate cosmological parameters from 3-dimensional universe data by using a 3D CNN

Input
(4×512×512×512 voxels)



Input ► **CNN** ► Predict

53 GiB

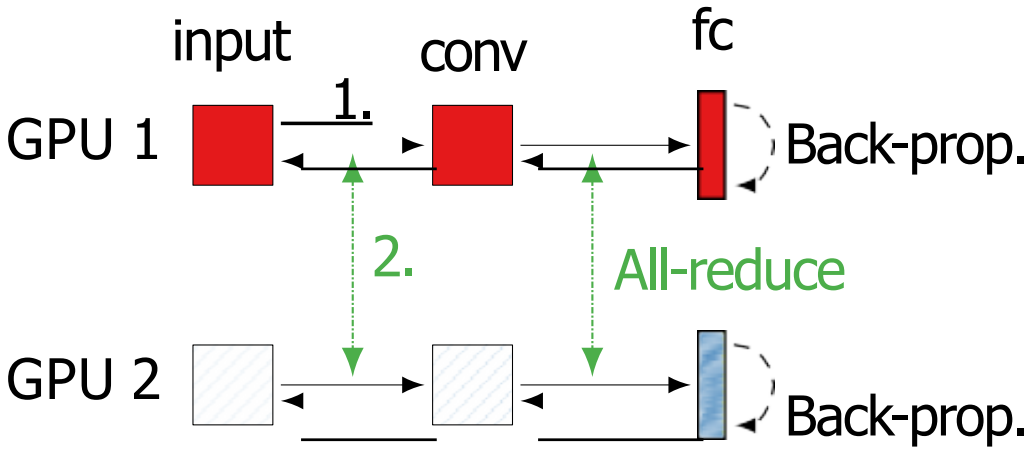
Output
(A vector of length 4)

$$\begin{bmatrix} \Omega_m \\ \sigma_8 \\ n_s \end{bmatrix} = \begin{bmatrix} -0.242 \\ 0.145 \\ -0.489 \end{bmatrix}$$

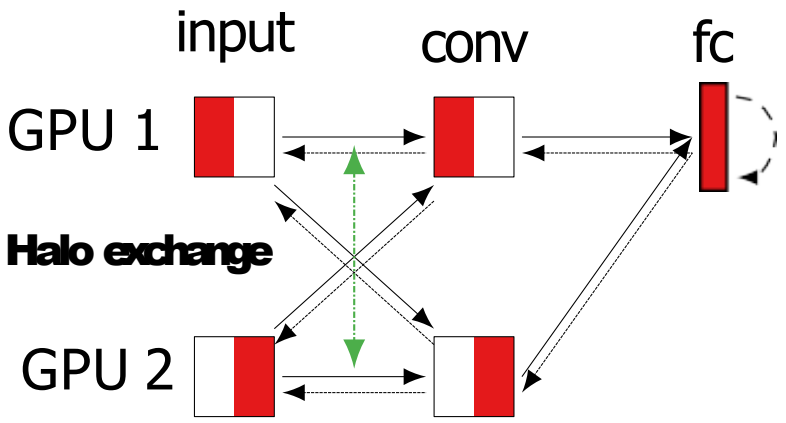
- **Problem**: GPU memory is too small to process high-resolution universe data
  → Another way to parallelize the model efficiently?

Background

- **Data-parallel** training distributes data samples among GPUs
  - ✓ Good weak scalability (O(1000) GPUs)



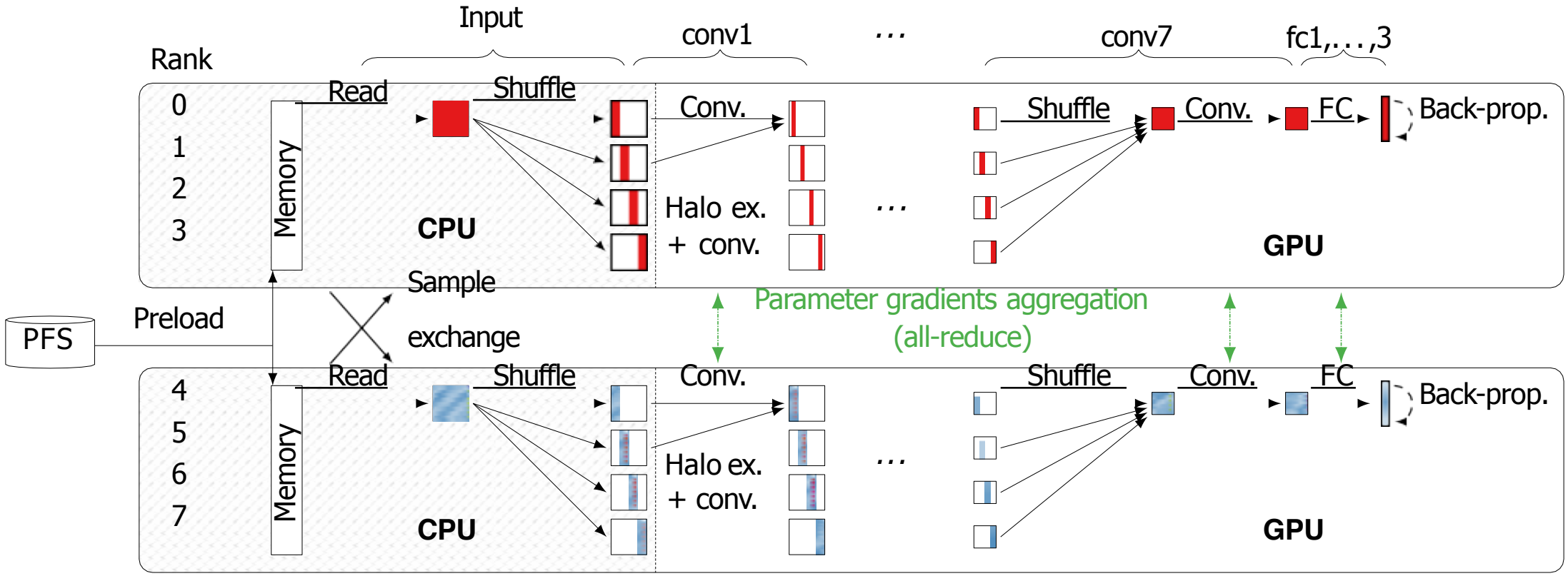
- **Model-parallel** training distributes the computation of a single sample (model) among GPUs
  - ✓ Can use more GPUs per sample
  - ✓ Can train larger models



- Data-parallelism + model-parallelism = **Hybrid-parallelism**

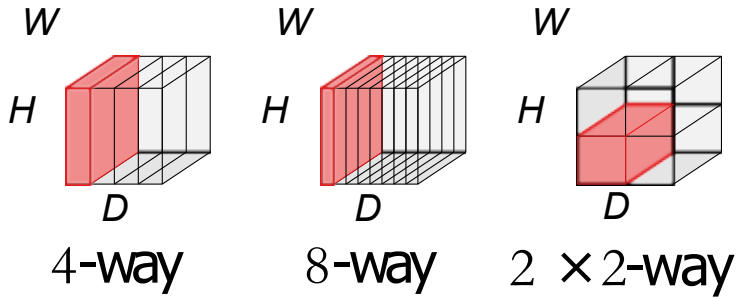
Proposal: Extending Distconv for 3D CNNs

- **LBANN** + **Distconv** [2]: A parallelized stencil computation-like hybrid-parallel CNN kernel library



Evaluation: Weak scaling

- Achieved **111x** of speedup over 1 node by exploiting hybrid-parallelism, even if layer-wise communication is introduced
- The 8-way partitioning is **1.19x of 4-way partitioning** with a mini-batch size of 64



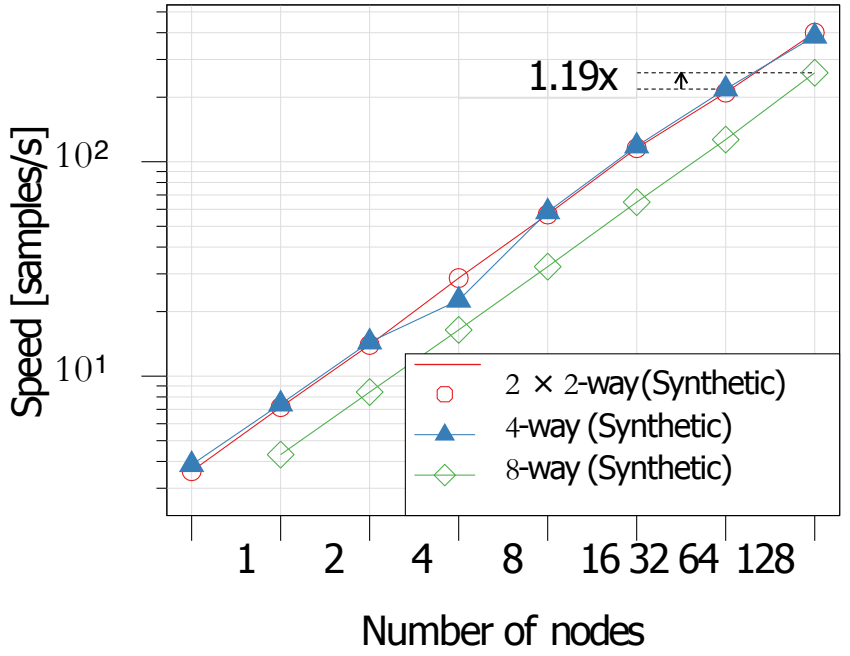


Figure: Weak scaling of the CosmoFlow network.

Evaluation: Strong scaling

- Achieved **2.28x** of speedup on 4 nodes (16 GPUs) compared to one node when $N = 1$
- The scalability limit here is 8 GPUs, and the main bottleneck is input data loading

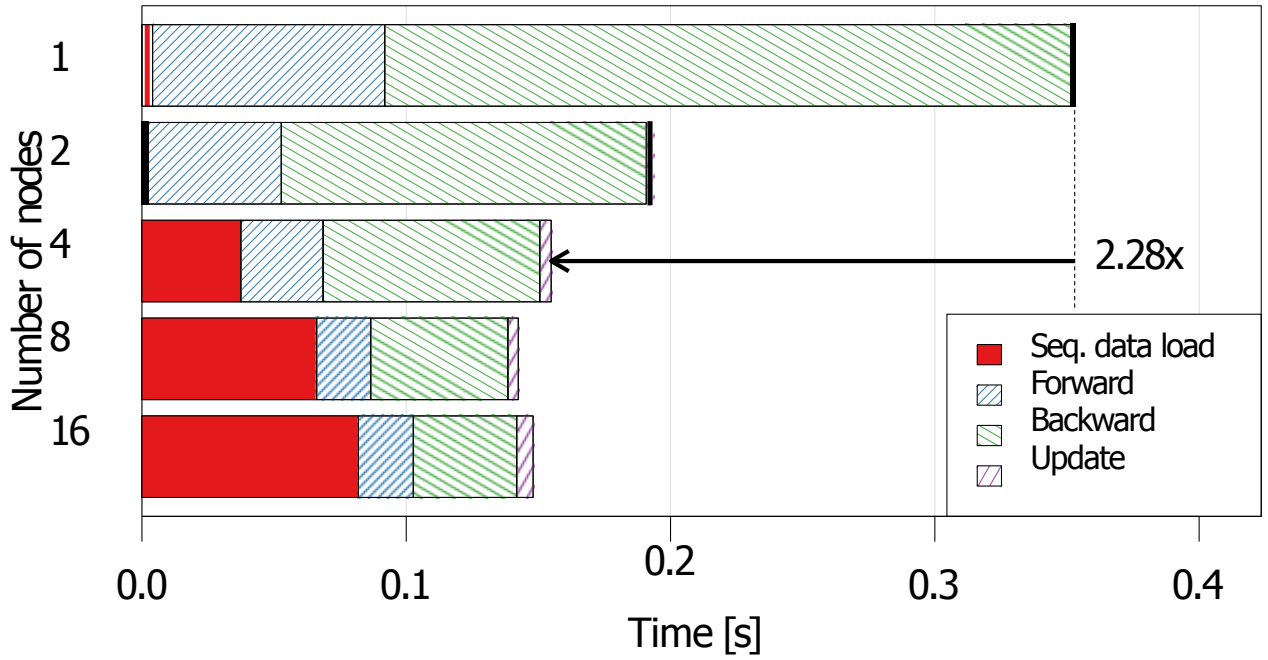


Figure: Breakdown of the strong scaling experiment when $N = 1$.

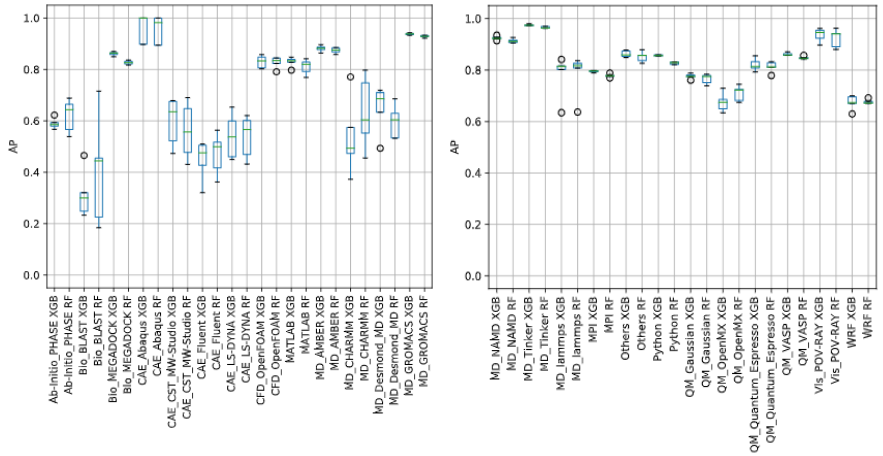
Machine Learning Models for Predicting Job Run Time-Underestimation in HPC system [SCAsia 19]

Motivation & Negative effects

1. When submitting a job, users need to estimate their job runtime
2. If job runtime is underestimated by the users
3. Job will be terminated by HPC system upon reaching its time limit

- Increasing time and financial cost for HPC users
- Wasting time and system resources.
- Hindering the productivity of HPC users and machines

- Evaluating by Average Precision(AP)

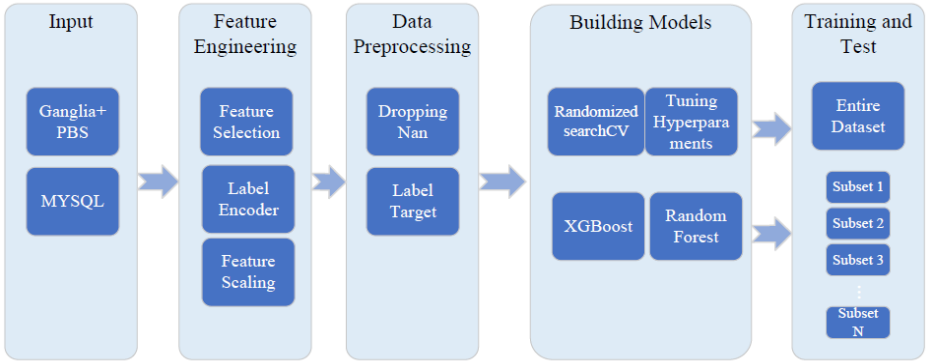


Evaluating by Simulation with Saved-Lost Rate (SLR)

$$SLR_C = \frac{Saved}{Lost + Punishment} = \frac{\sum_{tp=1}^{TP} (j.used\_walltime - C)_{tp}}{\sum_{p=1}^{P} j.used\_walltime_p + \sum_{fp=1}^{FP} C_{fp}}$$

Method

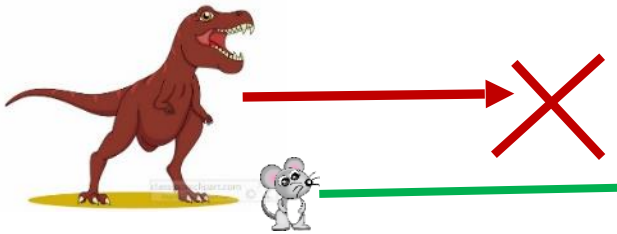
- Apply machine learning to train models for predicting whether the user has underestimated the job run-time
- Using data produced by TSUBAME 2.5



- Runtime-underestimated jobs can be predicted with different accuracy and SLR at different checkpoint times
- Summing up the "Saved" time of all the applications at best SLRs checkpoints, 24962 hours can be saved in total with existing TSUBAME 2.5 data
- Helping HPC users to reduce time and financial loss
- Helping HPC system administrators free up computing resources

Guo, Jian, et al. "Machine Learning Predictions for Underestimation of Job Runtime on HPC System." Asian Conference on Supercomputing Frontiers. Springer, 2018

Many Core Era



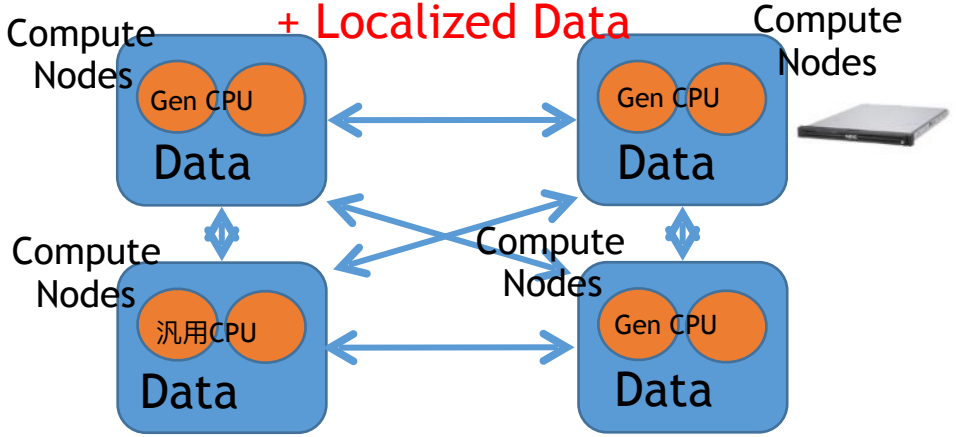
Post Moore
Cambrian Era

Flops-Centric Monolithic Algorithms and Apps

Flops-Centric Monolithic System Software

Hardware/Software System APIs
Flops-Centric Massively Parallel Architecture

Homogeneous General Purpose Nodes
+ Localized Data



Loosely Coupled with Electronic Interconnect

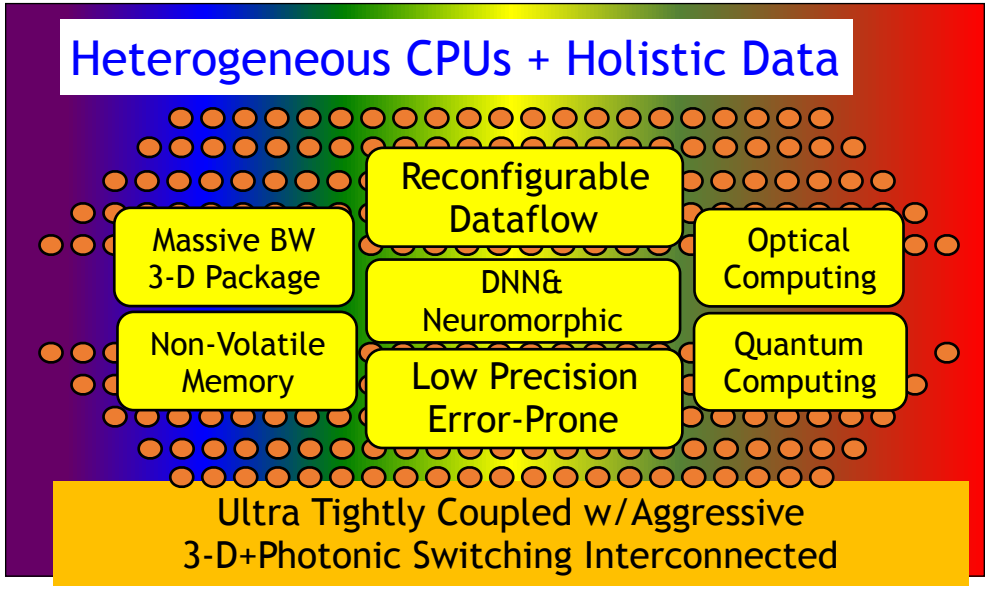
Transistor Lithography Scaling
(CMOS Logic Circuits, DRAM/SRAM)

~2025
M-P Extinction Event

Cambrian Heterogeneous Algorithms and Apps

Cambrian Heterogeneous System Software

Hardware/Software System APIs
"Cambrian" Heterogeneous Architecture



Novel Devices + CMOS (Dark Silicon)
(Nanophotonics, Non-Volatile Devices etc.)

R-CCS Strategies Towards Post-Moore Era

- **Basic Research on Post-Moore**
  - Funded 2017: DEEP-AI CREST (Matsuoka)
  - Funded 2018: NEDO 100x 2028 Processor Architecture (Matsuoka, Sano, Kondo, SatoK)
  - Funded 2019: Kiban-S Post-Moore Algorithms (NakajimaK etc.)
  - Submitted: Neuromorphic Architecture (Sano etc. w/Riken AIP, Riken CBS (Center for Brain Science))
  - In preparation: Cambrian Computing (w/HPCI Centers)
- **Author a Post-Moore Whitepaper towards Fugaku-next**
  - All-hands BoF last week at annual SWoPP workshop
  - Towards official “Feasibility Study” towards Fugaku-next
  - Similar efforts as K => Fugaku started in 2012

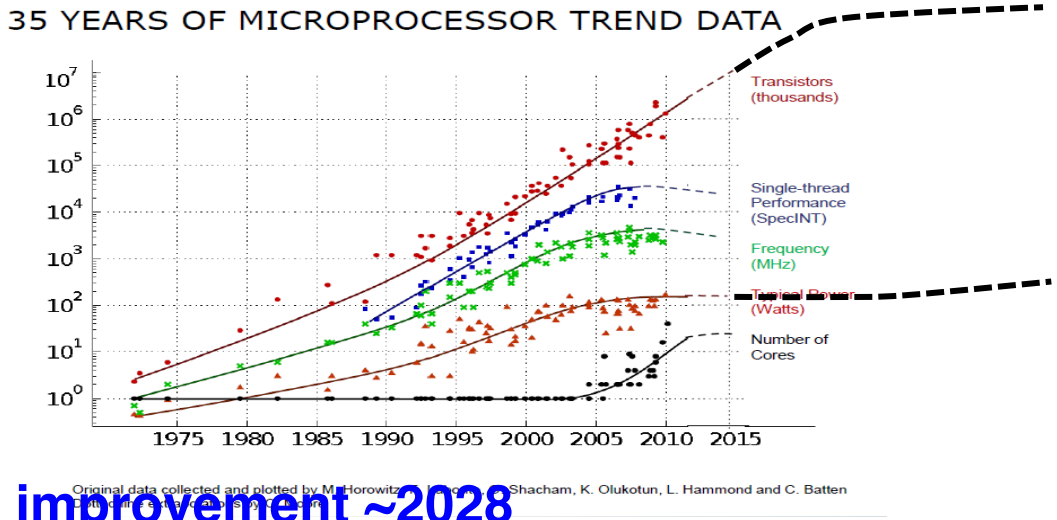
Basic Research #1: NEDO 100x Processor

2028: Post-Moore Era

～2015 ~25 Years Post-Dennard,
Many-core Scaling era

2016～Moore's Law Slowing Down

2025～Post-Moore Era, end of transistor
lithography (FLOPS) improvement



Research: Architectural investigation of perf. improvement ~2028

- 100x in 2028 c.f. mainstream high-end CPUs circa 2018 across applications

Key to performance improvement: from FLOPS to Bytes – data movement architectural optimization

- CGRA – Coarse-Grained Reconfigurable Vector Dataflow
- Deep & Wide memory architecture w/advanced 3D packaging & novel memory devices
- All-Photonic DWM interconnect w/high BW, low energy injection
- Kernel-specific HW optimization w/low # of transistors & associated system software, programming, and algorithms

NEDO 100x Processor

Towards 100x processor in 2028

- Various combinations of CPU architectures, new memory devices and 3-D technologies
- Perf. measurement/characterization/models for high-BW intra-chip data movement
- Cost models and algorithms for horizontal & hierarchical data movement
- Programming models and heterogeneous resource management