Research projects and collaborations

HIGPH PERFORMANCE
BIG DATA
RESEARCH TEAM

- Fundamental R&D in HPC
  - Design space exploration for the next-gen supercomputers (Jens Domke, Matsuoka team, AIST)
  - Benchmarking and Performance analysis of big data applications on NVDIMM (Andres Rubio Proano, FSU)
  - Reproducibility in MPI/OpenMP applications by record-and-replay techniques
  - Auto-detection of checkpoint variables (Nanchang Hangkong University, PNNL)
  - ABFT for tensor operations in deep learning framework (Nanchang Hangkong University, PNNL)
  - Failure analysis on Fugaku (Shoji, Yamamoto, Northeastern Univ.)
  - I/O optimization for 2D/3D sub-tiling of MPI-IO on a near-node local storage architecture (KTH)
- HPC for AI/BD
  - Data platform for Fugaku and RSC facilities, SPring-8 and SACLA (Matsuda, Kaneyama, Harada, Shoji +RSC)
  - DL4Fugaku: Deep learning framework tuning on Fugaku (Matsuoka team, Imamura team, Fujitsu)
  - Storage performance analysis and storage design exploration for deep learning (Takaaki Fukai)
- AI/BD for HPC
  - Big data compression with AI techniques (FSU)

Pacific Northwest NATIONAL LABORATORY

放射光科学総合研究センター
RIKEN SPring-8 Center

FUJITSU

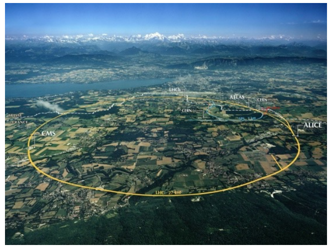
KTH VETENSKAP OCH KONST

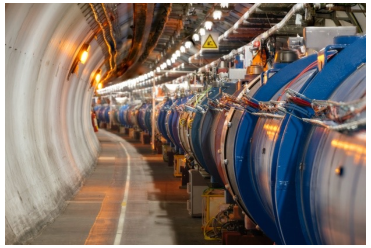
AIST

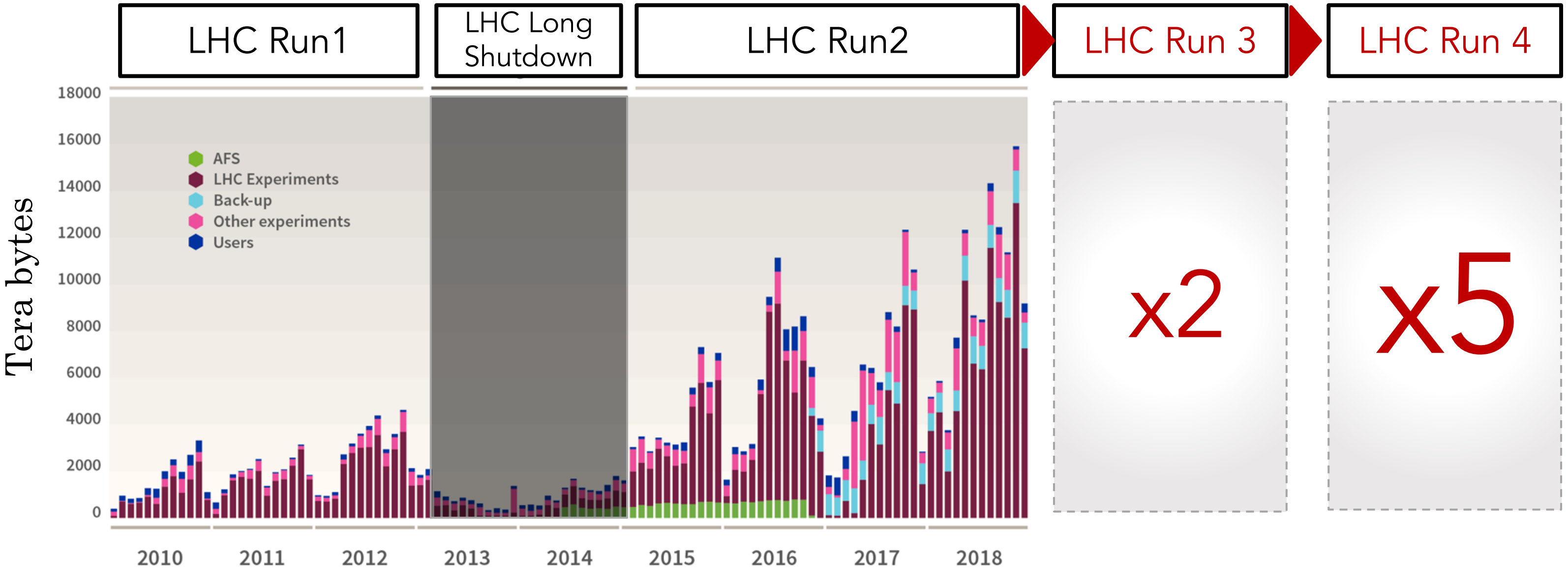
Big Data Generation and Transfer

Generation: Scientific big data is generated every day all over the world

- LHC (Large Hadron Collider) in CERN generated about 88PB of data in 2018 [1]
  - *"Data archival is expected to be two-times higher during Run 3 and five-times higher or more during Run 4 (foreseen for 2026 to 2029). "*







[1] Esra Ozcesmeci, "LHC: pushing computing to the limits", https://home.cern/news/news/computing/lhc-pushing-computing-limits March 1st, 2019

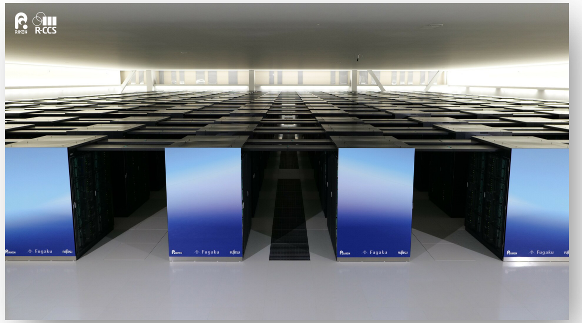
Big Data Generation and Transfer (Cont'd)

Transfer: Data transfer is an essential part of data analytics

- Generated data from sensors must be transferred to internal computers for the analysis
- In some case, the facilities needs to transfer the data to external collaborators via WAN
  – e.g.) In LHC, 830 PB of data and 1.1 billion files were transferred all over the world [1]

Sensors

Big data transfer



Internal/External Computers

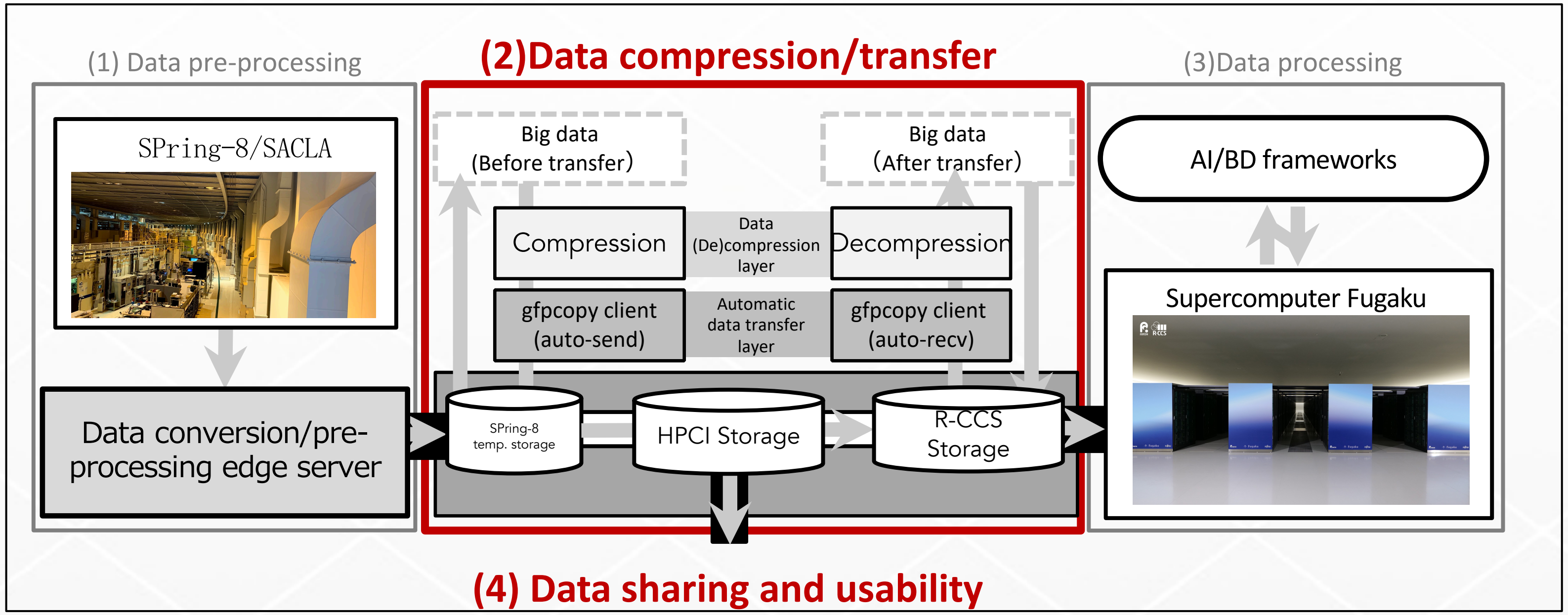
Efficient data transfer and its management is important in big data analysis

[1] Esra Ozcesmeci, "LHC: pushing computing to the limits", https://home.cern/news/news/computing/lhc-pushing-computing-limits March 1st, 2019

Research and development of an infrastructure for collecting, analyzing and utilizing big data in large-scale research facilities (Fugaku/SPring-8/SACLA)

Project Leader: Kento Sato

[Overview]



Project team members

Members
Kento Sato, R-CCS
Fumiyoshi Shoji, R-CCS
Motohiko Matsuda, R-CCS
Kaneyama Hidetomo, R-CCS
Hiroshi Harada, R-CCS
Jorji Nonaka, R-CCS
Kentaro Sano, R-CCS
Masaaki Kondo, R-CCS
Tomohiro Ueno, R-CCS
Takaki Hatsui, RSC
Yasumasa Joti, RSC

[Objective]

- The Objective of this project is to establish a "big data infrastructure" that enables data collection, analysis, and utilization between SPring-8/SACLA and Fugaku. We are working on following sub-proejcts:
  - (1) Data pre-processing infrastructure: To efficiently store experimental data obtained from sensors, we perform data conversion and pre-processing at the hardware level using FPGA
  - (2) Data compression and transfer infrastructure: We develop data compression and transfer infrastructure
  - (3) Data analysis infrastructure: We will build an infrastructure (workflow tools and deep learning framework) to efficiently analyze the data in HPC systems
  - (4) Data utilization infrastructure: We will build a data utilization infrastructure to make use of the collected primary data and analysis results (e.g., Single sign-on authentication, GakuNin RDM etc.)

HPCIシステム概要

- 高速ネットワーク
  - 「富岳」と全国の大学や研究機関(図1)の計算資源やストレージを高速ネットワーク(SINET5、図2)で結んだシステム

- シームレスな認証環境
  - 全国のHPC資源を効率よく利用する仕組みとして、HPCIは公開鍵暗号方式を用いたシングルサインオン (SSO)と呼ばれる機能を提供
  - 利用者は任意のHPCIリソースへログインすれば、利用権のある全てのHPCIリソースをシームレスに利用可能。
    - （例）「富岳」へログインして計算を行い結果をHPCI共用ストレージに転送したい場合に、再度 HPCI共用ストレージへログインする必要はありません。

図1：HPCIに共用計算資源を提供する機関 (平成30年度)

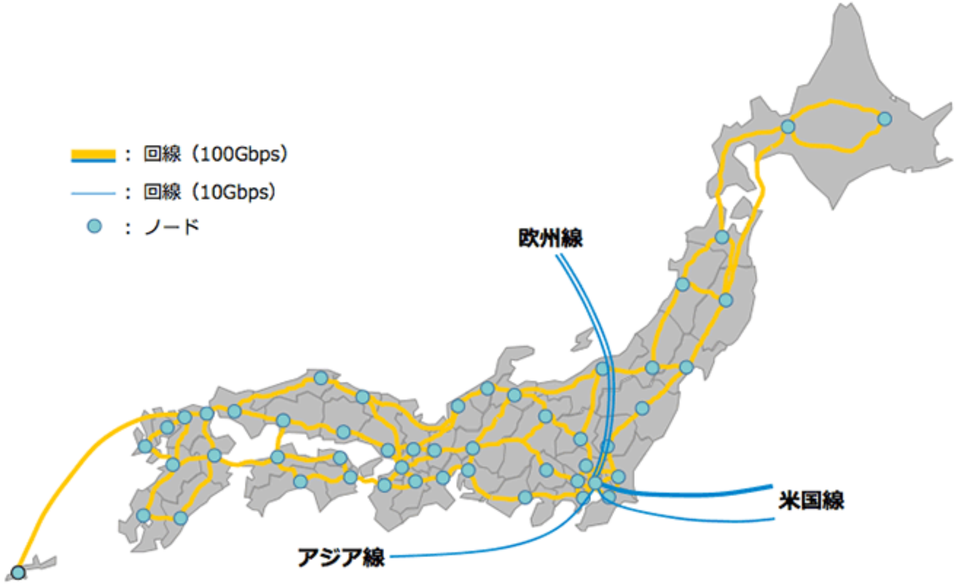


図2：SINET5回線構成図

HPCI共用計算資源

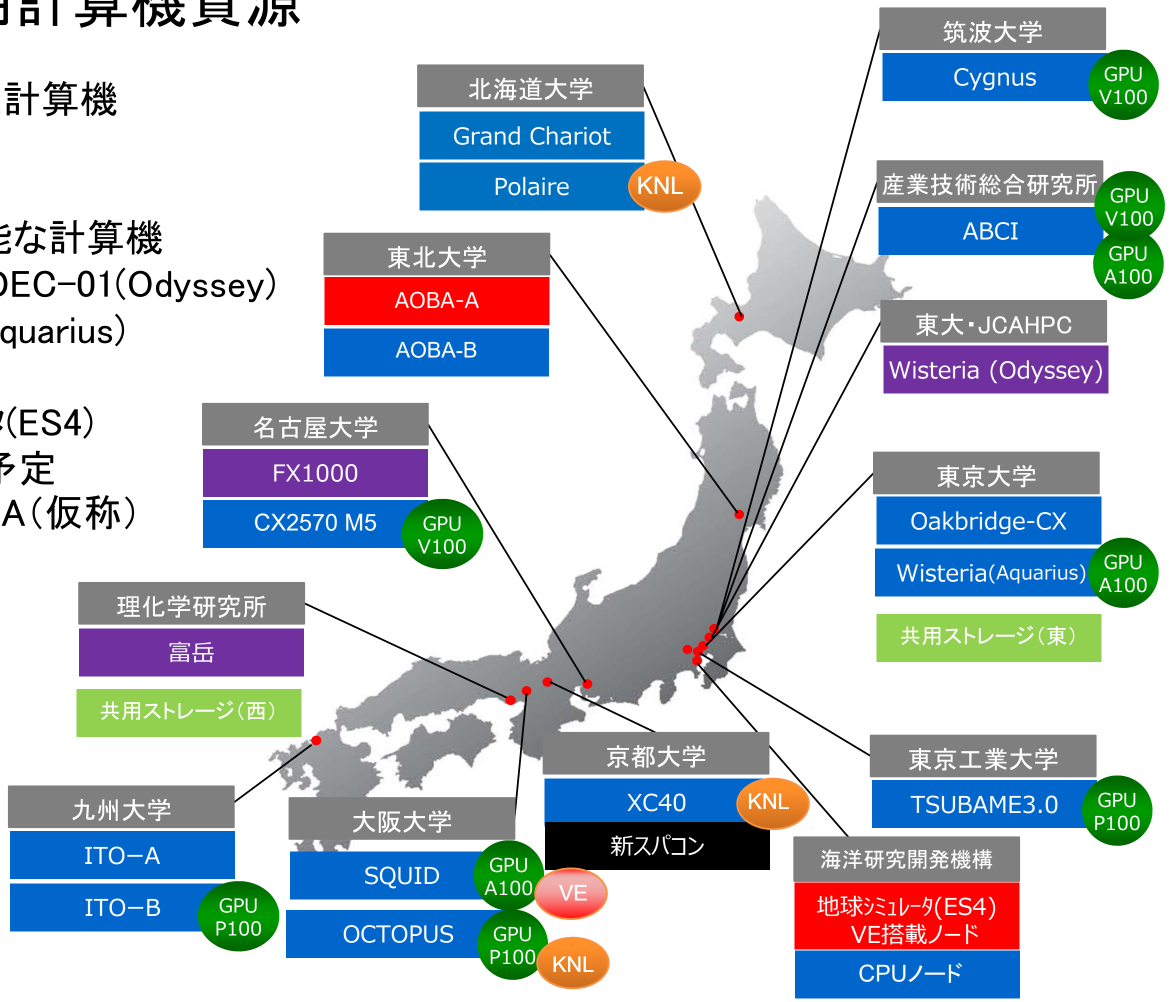
HPCI：High Performance Computing Infrastructure

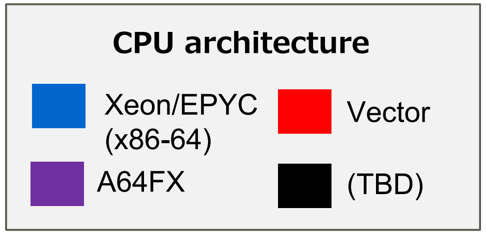
◻ 2022年度HPCI共用計算機資源

■ 2021年度にて提供終了した計算機
- ・JCAHPC　Oakforest-PACS
- ・東大　Reedbush-H / L

■ 2022年度から年間利用可能な計算機
- ・東大/JCAHPC　Wisteria/BDEC-01(Odyssey)
- ・東大　Wisteria/BDEC-01 (Aquarius)
- ・阪大　SQUID
- ・JAMSTEC　地球シミュレータ(ES4)

■ 2022年度中に計算機更新予定
- ・京大　　XC40 → 新スパコンA（仮称）





ネットワーク基盤：学術情報ネットワーク（SINET5）を利用

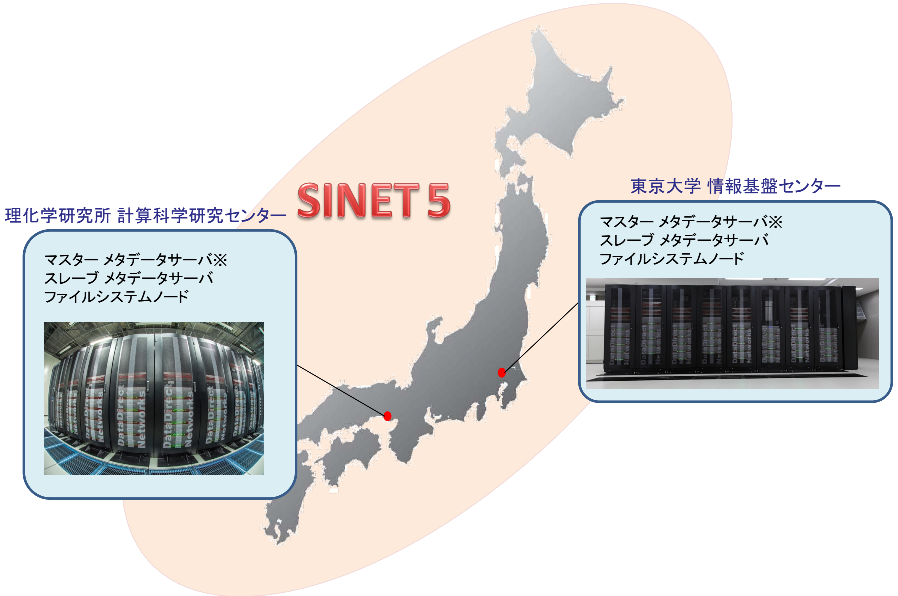
3

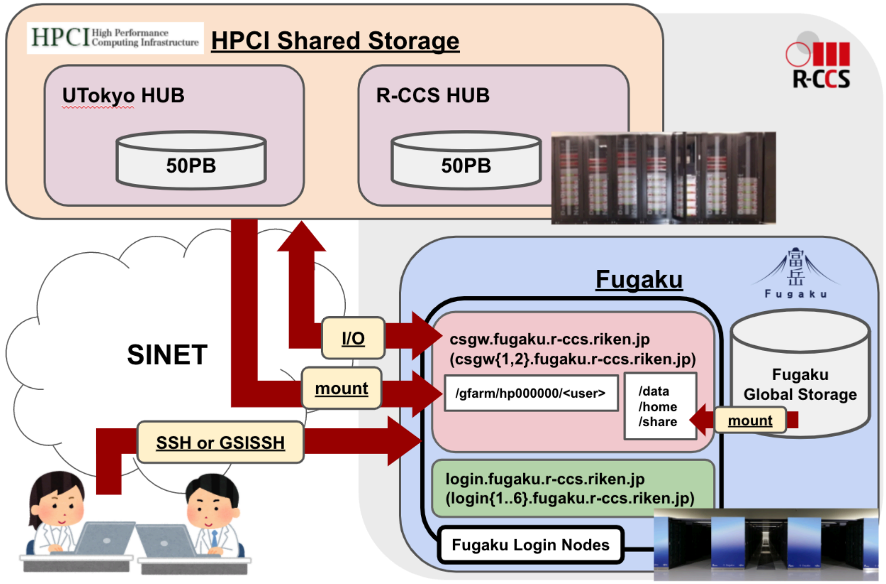
文部科学省委託事業「HPCIの運営」代表機関 一般財団法人高度情報科学技術研究機構 神戸センター “HPCI共用計算資源について”, 令和4年度HPCIシステム利用研究課題募集説明会 (2021年 9月28日版) ,
https://www.hpci-office.jp/materials/r04a_boshu_setsumeikai_hpci.pdf?a

HPCI共用ストレージ

- 概要
  - HPCI 計算資源利用時における大容量のデータの格納や HPCI 計算資源間(or センター間)あるいは HPCI コミュニティ(or 各研究分野)でのデータ共有を目的とした、広域に分散した大規模ストレージ

- ファイルシステム
  - Gfarmファイルシステムとよばれる広域分散ファイルシステムで実現
  - ２拠点のストレージを集約し、全国のHPCI計算機資源から、高速・大容量の単一ファイルシステムとしてデータ共有が可能
  - 富岳ゲートウェイを含む HPCI 計算機資源のログインノードでは、HPCI 課題参加者が共用ストレージを利用できるよう、共用ストレージのクライアント環境がインストールされています。
  - 共用ストレージのクライアント環境をお手元の機材にインストールすることで、お手元の機材から直接共用 ストレーズへアクセスが可能です





HPCI 共用ストレージ 利用マニュアル, 2021/04/16, 文書番号:HPCI-ST01-004-03
https://www.hpci-office.jp/materials/hpci-st01-004.pdf?0416

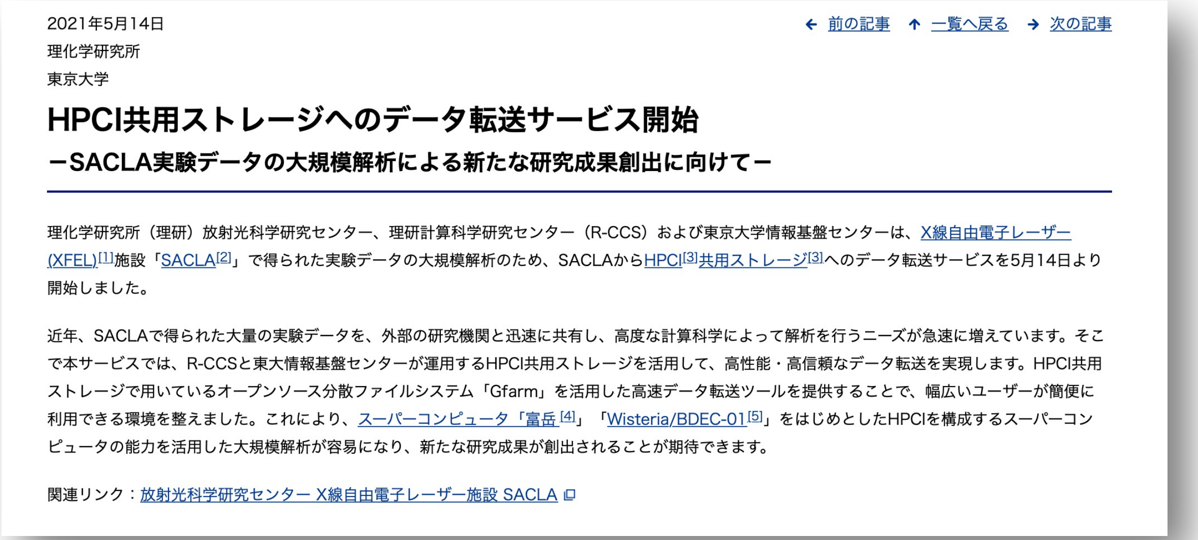
Data transfer service in SACLA

- We started data transfer service from SACLA to HPCI shared storage
  - To facilitates data analitycs in HPCI systems inlucidng Fugaku
- We are planing to expand the service to SPring-8 synchortron radiation facility and enhance the usability (Common authentication scheme, GakuNin RDM etc.)

SACLA HPC:
Data Transfer Service to HPCI Shared Storage

Source (May 14, 2021): http://xfel.riken.jp/users/bml09-1.html

Data Transfer Service to HPCI Shared Storage
Toward the creation of innovative achievement through SACLA


2021年5月14日
理化学研究所
東京大学

← 前の記事 ↑ 一覧へ戻る → 次の記事

HPCI共用ストレージへのデータ転送サービス開始

－SACLA実験データの大規模解析による新たな研究成果創出に向けて－

理化学研究所（理研）放射光科学研究センター、理研計算科学研究センター（R-CCS）および東京大学情報基盤センターは、X線自由電子レーザー(XFEL)[1]施設「SACLA[2]」で得られた実験データの大規模解析のため、SACLAからHPCI[3]共用ストレージ[3]へのデータ転送サービスを5月14日より開始しました。

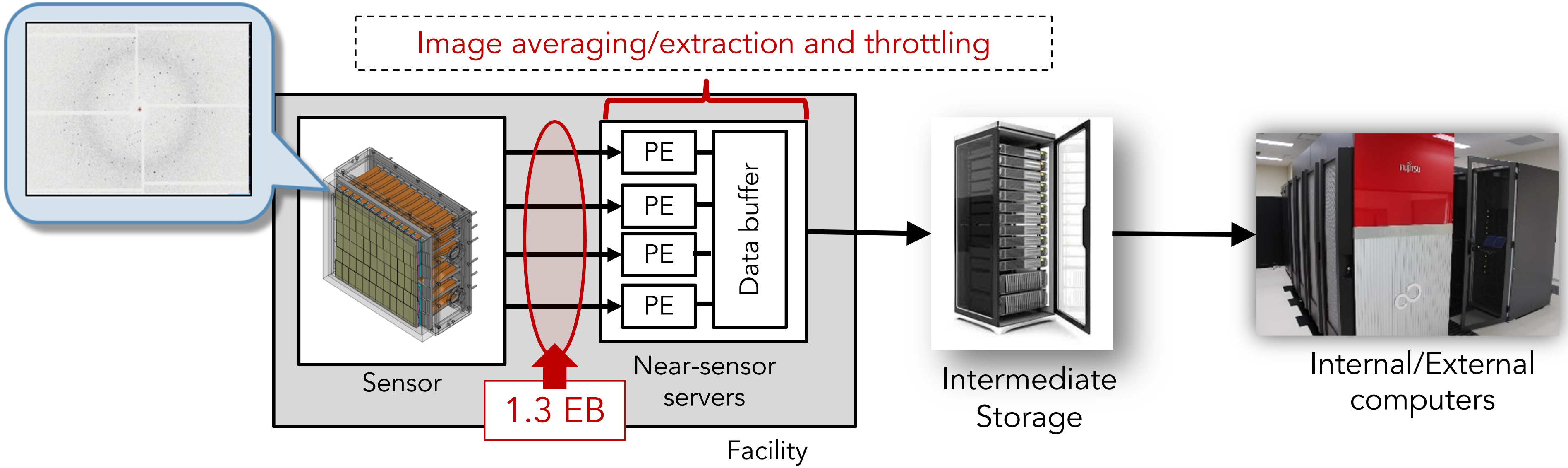
近年、SACLAで得られた大量の実験データを、外部の研究機関と迅速に共有し、高度な計算科学によって解析を行うニーズが急速に増えています。そこで本サービスでは、R-CCSと東大情報基盤センターが運用するHPCI共用ストレージを活用して、高性能・高信頼なデータ転送を実現します。HPCI共用ストレージで用いているオープンソース分散ファイルシステム「Gfarm」を活用した高速データ転送ツールを提供することで、幅広いユーザーが簡便に利用できる環境を整えました。これにより、スーパーコンピュータ「富岳[4]」「Wisteria/BDEC-01[5]」をはじめとしたHPCIを構成するスーパーコンピュータの能力を活用した大規模解析が容易になり、新たな研究成果が創出されることが期待できます。

関連リンク：放射光科学研究センター X線自由電子レーザー施設 SACLA

Source (May 14, 2021): https://www.riken.jp/pr/news/2021/20210514_1/

Big data transfer in SPring-8

- SPring-8 public beamlines (26 BLs) generated 0.32 PB/year in 2017
- With the next generation detector (CITIUS), it is projected that the facility will generate 1.3 ExaB of raw data per year in 2025
  - Actual transfer size can be reduced to 100-400 PB by
    - Image averaging/extraction
    - Reducing duty ratio to throttle data generation rate



We are trying to further compress this big data
to accelerate data transfer from sensors to HPC systems

Prediction is one of keys for good compression

Compression

Original data

1.1	1.5	1.8	2.1
1.0	1.4	2.3	2.7
1.3	1.8	2.5	3.1
1.9	2.1	2.6	3.3

diff (–)

Predicted data

1.1	1.5	1.8	2.1
1.0	1.3	2.3	2.7
1.3	1.8	2.5	3.0
1.9	1.9	2.5	3.3

=

Delta

0	0	0	0
0	0.1	0	0
0	0	0	0.1
0	0.2	0.1	0

Sequence of zeros is likely to appear many times

gzip

(LZ77 & Huffman encoding)

Compressed data

gunzip

Decompression

Original data

1.1	1.5	1.8	2.1
1.0	1.4	2.3	2.7
1.3	1.8	2.5	3.1
1.9	2.1	2.6	3.3

=

Predicted data

1.1	1.5	1.8	2.1
1.0	1.3	2.3	2.7
1.3	1.8	2.5	3.0
1.9	1.9	2.5	3.3

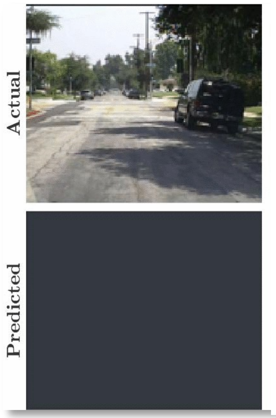
diff (+)

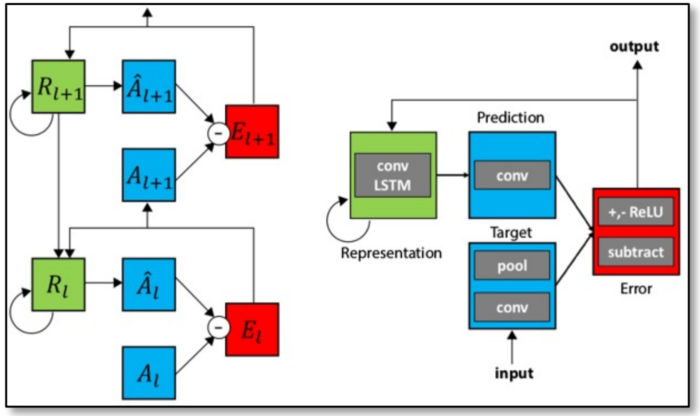
Delta

0	0	0	0
0	0.1	0	0
0	0	0	0.1
0	02.	0.1	0

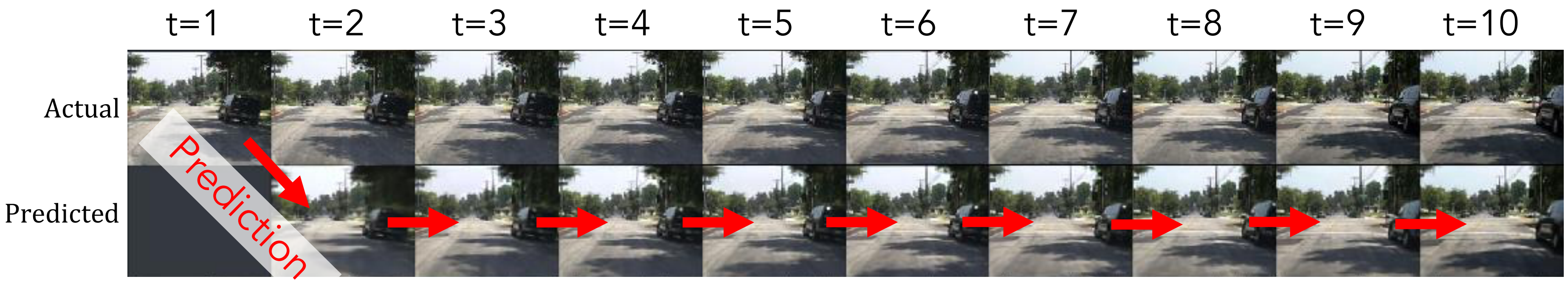
We use deep neural network (PredNet) for prediction

- PredNet [1]
  - Deep recurrent convolutional neural network
  - Given a frame of pictures/video, this NN can predict multiple future frames





https://coxlab.github.io/prednet/

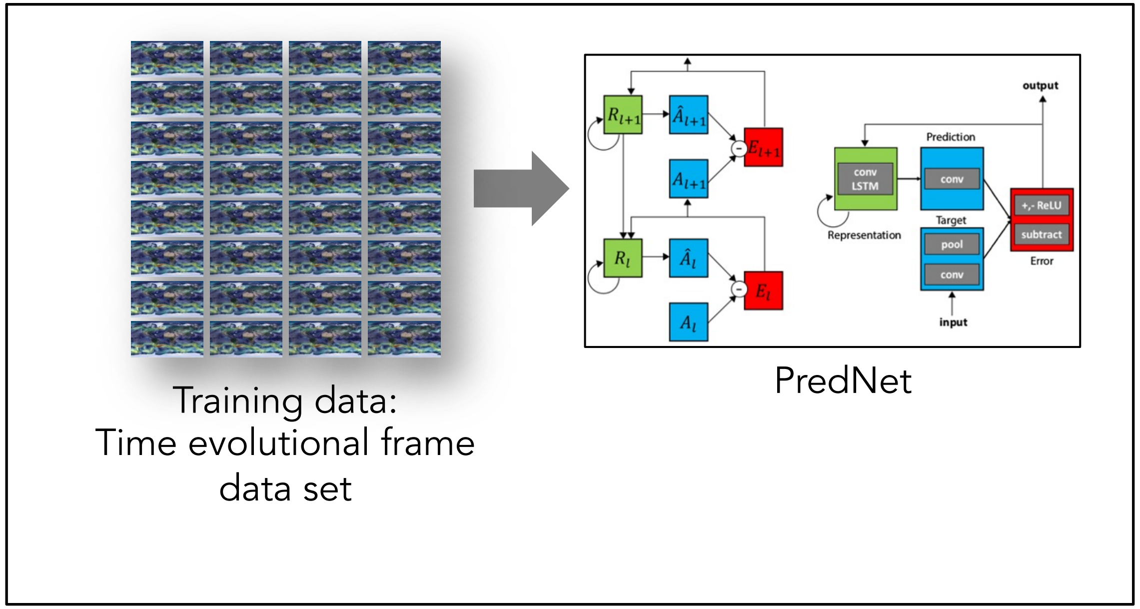


[1] Lotter, W., Kreiman, G., Cox, D.: Deep predictive coding networks for video prediction and unsupervised learning. arXiv preprint arXiv:1605.08104 (2016)

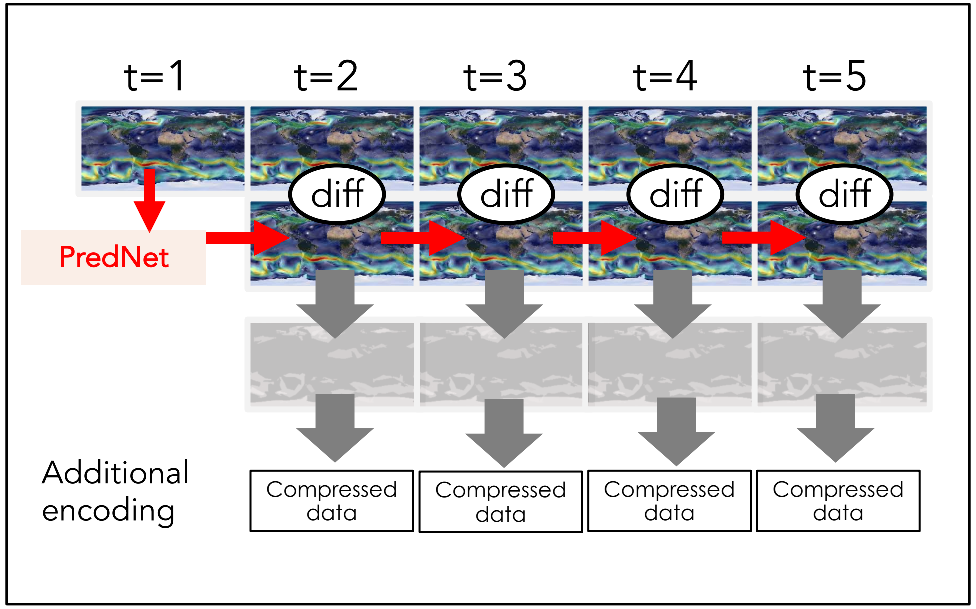
Compression: Predict future frames and encode

- We train PredNet to learn how pixels move and how fast
  - i.e.) Giving a number of time evolutional frames to PredNet
- When compressing frames from t=1 to t=5, we predict future frames from original data (t=1)
- We compute diff, apply series of encoding
- We only store (1) base frame data (t=1) and (2) compressed data

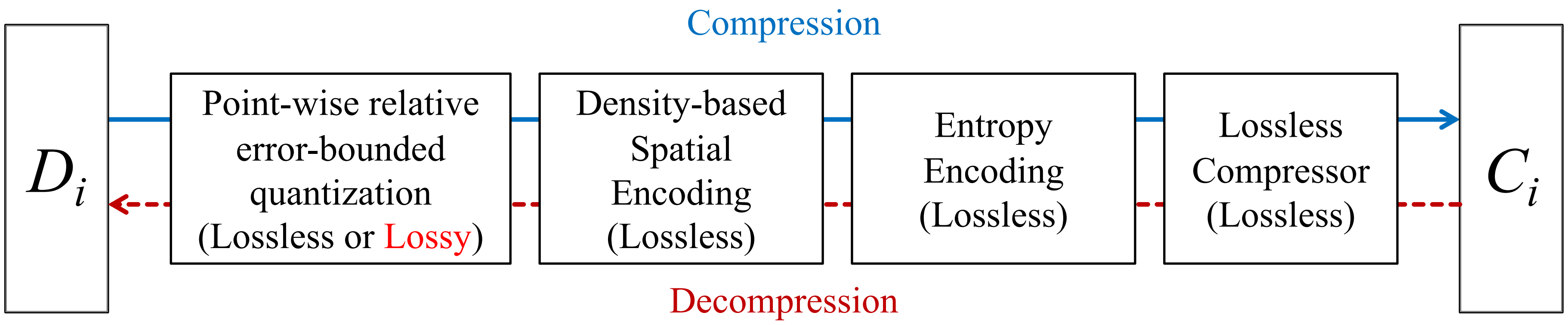
Training



Inference + Data compression



Encoding workflow



- Quantization is a key data conversion to give good compression rate
  - This data conversion tries to maximize data compression rate while bounding certain-level errors

- Point-wise relative error bound
  - All the individual values are kept below a specified error bound threshold (ε)
  - Formulation
    - Give original data: $D = \{d_0, d_1, \ldots,\}$ and quantized data: $D' = \{d'_0, d'_1, \ldots,\}$
    - The following inequality holds for each data point:

$$\max_{d_i \in D,\, d'_i \in D'} \left| \frac{d_i - d'_i}{d_i} \right| \leq \varepsilon$$

TEZIP achieves high compression rate with comparable compression time

- TEZIP achieves an improvement up to 3.2× in terms of compression ratio.
- On average, lossless TEZIP delivers 2.1× better compression ratio compared to the second-best lossless compressor x265
- “Baseline” computes delta values from the previous frame

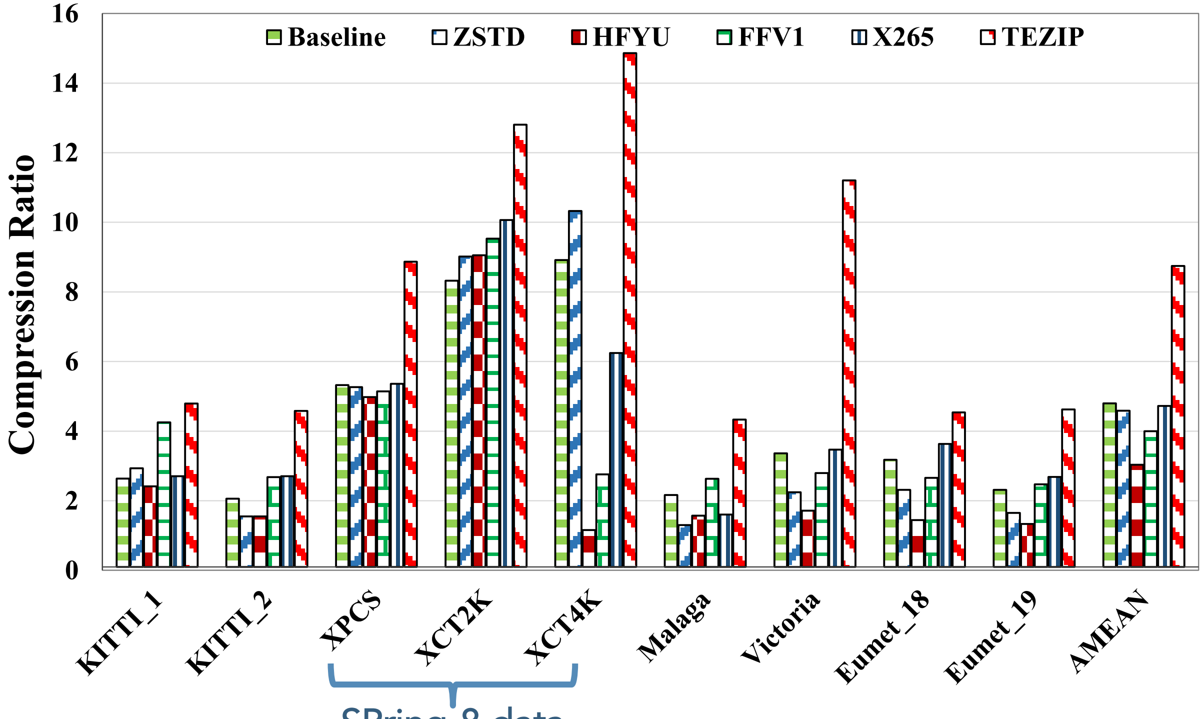


Fig. 8. Compression ratio with lossless compressors.

Lossy compression mode in TEZIP futher improves compression ratio under the same error-bound

- TEZIP's lossy compression mode can set point-wise relative error-bound at quantization
  - The error-bound is set for SZ and TEZIP (configurable) based on errors in ZFP (unconfigurable)
- Results
  - TEZIP achieves an improvement up to 3.3x than the second best (SZ) in terms of compression ratio

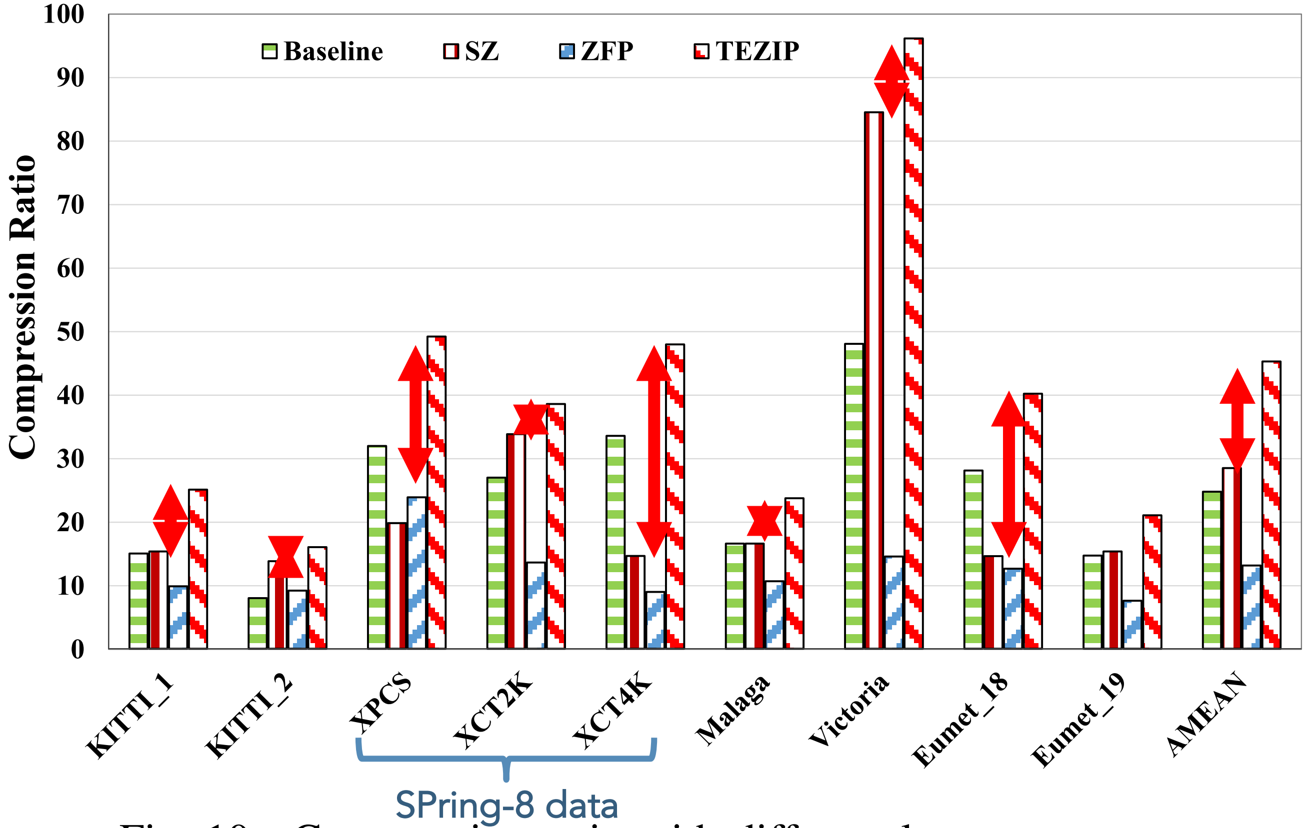
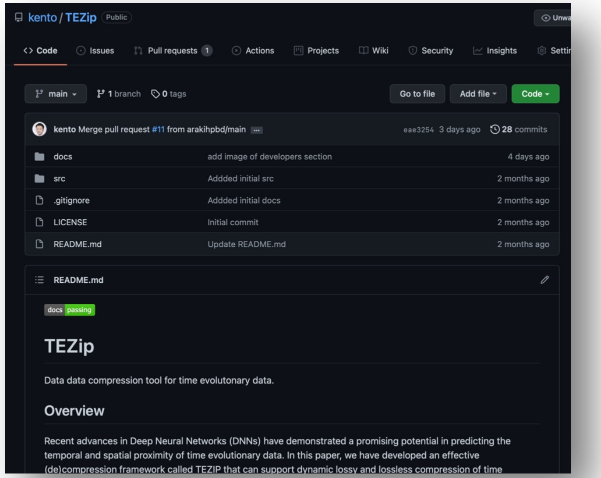


Fig. 10. Compression ratio with different lossy compressors

TEZIP is open-source software and future works

- We open-sourced TEZIP and released documents
- Future works
  - Improvement in quantization
  - Improvement in Prediction
    - TEZIP relies on a generic predictor (PreNet)
    - The compression ratio will be futhrer imporved with domain-specific predictors

Github:
https://github.com/kento/TEZip



Generic predictor (PredNet)

Domain-specific predictor

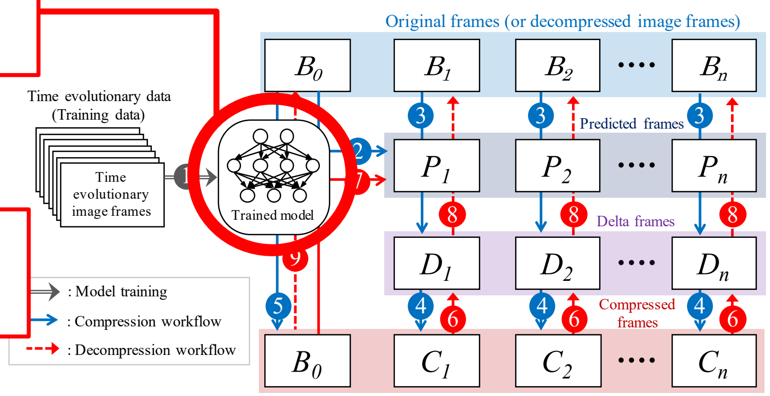
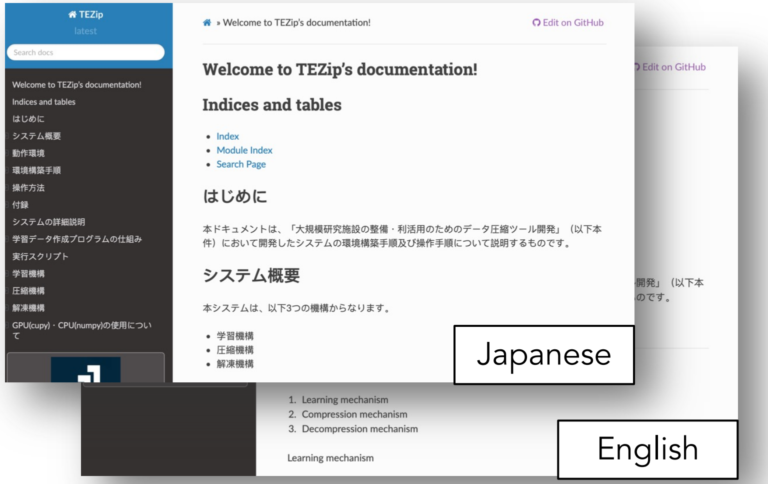


Fig. 1. Workflows of TEZIP (de)compression

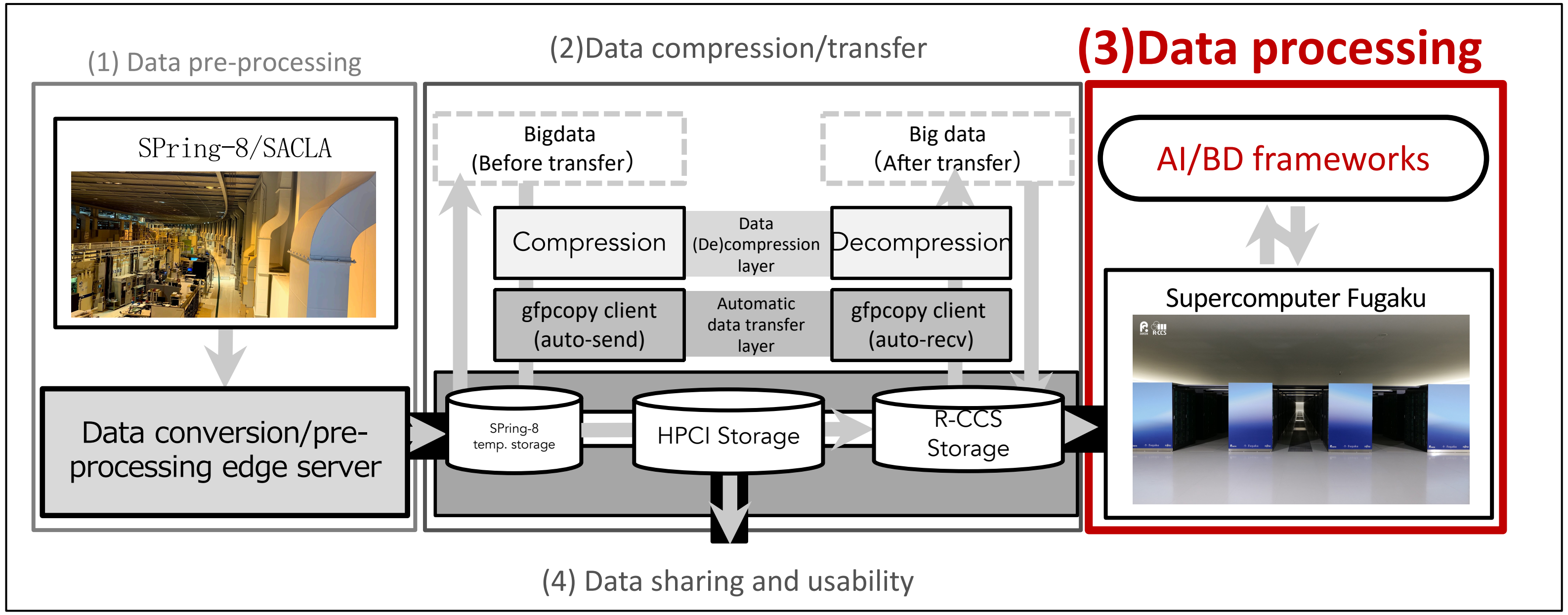
Readthedocs:
https://tezip.readthedocs.io/en/latest/?badge=latest

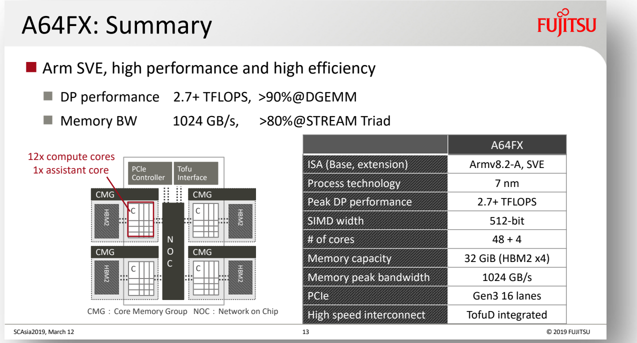


This project is seeking for a Postdoc or a Researcher !
https://www.hpbd.r-ccs.riken.jp/recruiting/

Supercomputer Fugaku & Deep learning

- Once we move data to computers, the users will analyze the data and can use AI for the feature detection, Image recognition, segmentation etc.
- We must provide fast and scalable AI training environments on Fugaku
- GPU has become a popular platform for executing DL, but we revisit the idea of running DL on CPUs in Fugaku





A64FX: Summary

FUJITSU

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

12x compute cores
1x assistant core

PCIe Controller, Tofu Interface, CMG, HBM2, NOC

CMG : Core Memory Group NOC : Network on Chip

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

SCAsia2019, March 12 13 © 2019 FUJITSU

Toshiyuki Shimizu, "Post-K Supercomputer with Fujitsu's Original CPU, A64FX Powered by Arm ISA", Nov. 15th, 2018

→ High perf. FP16/INT8
→ High bw mem (1024 GB/sec)
→ Scalable TofuD net.

To make use of Fugaku/A64FX performance, tuning AI software stack is indispensable

DL4Fugaku: Deep learning for Fugaku

- **Objective: Fast and scalable deep learning on Fugaku/A64FX**
  - Conduct porting, performance analysis and tuning
  - Deploy large-scale deep learning environment
  - Enhance the usability for production use in Fugaku

- **MOU for RIKEN/Fujitsu collaboration on AI framework development in Fugaku (Nov. 25, 2019)**

- **RIKEN R-CCS internal teams are working together**
  - Under collaboration with Industry & academia
  - Porting, tracing DL, performance analysis, tuning, merge to upstream

RIKEN R-CCS

Operation team
- ~~Application tuning development unit~~
  → Software Development Technology Unit

Research teams
- High performance AI system research team
- High performance big data research team
- Large-scale parallel numerical computing technology research team

Collaborators

- AIST
- Fujitsu limited
- ARM
- Linaro
- Cybozu
- Tokyo Tech
- ~~Fujitsu Laboratories~~

(alphabetical order)

https://www.hpbd.r-ccs.riken.jp/hpbd/en/dl4fugaku-project/

※ Some of software introduced in the rest of DL4Fugaku project slides is under development.
Experimental results will be changed in future in the course of tuning

DL4Fugaku Project Menbers

FUJITSU

Framework & oneDNN porting & tuning

Naoki Shinjo, Akira Asato, Atsushi Ike, Koutarou Okazaki, **Yoshihiko Oguchi**, Masahiro Doteguchi, **Jin Takahashi**, Kazutoshi Akao, Masaya Kato, Takashi Sawada, **Naoto Fukumoto**, Kentaro Kawakami, Naoki Sueyasu, Kouji Kurihara, Masafumi Yamazaki, Takumi Honda

Fugaku AI project

RIKEN

Tuning for Fugaku

Satoshi Matsuoka, High Performance Artificial Intelligence Systems Research Team Leader
Kento Sato, High Performance Big Data Research Team Leader
Kazuo Minami, Application Tuning Development Unit Leader
Akiyoshi Kuroda, Application Tuning Development Unit

arm Linaro

Cybozu Labs

Technical support

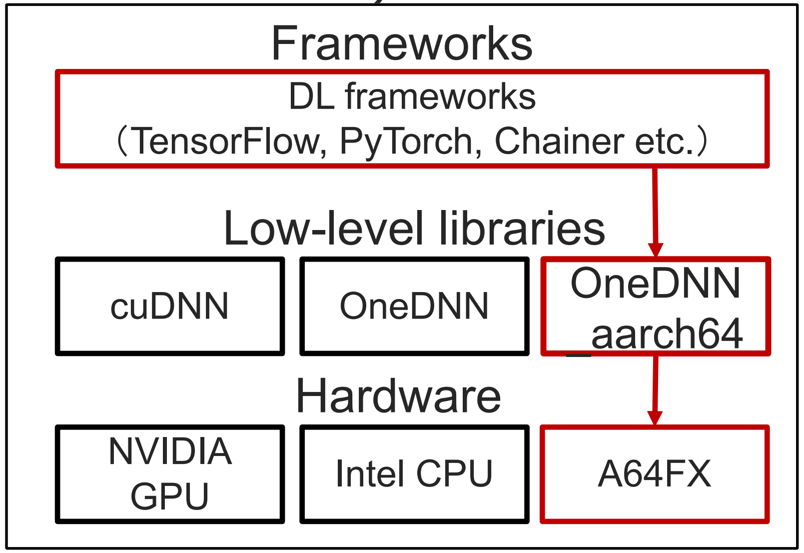
Shigeo Mitsunari (Xbyak)

Porting and Tuning approach

- **Deep learning software stack**
  - Deep learning frameworks reply on low-level numerical libraries optimized for specific hardware
    - cuDNN for NVIDIA GPU, OneDNN for Intel CPU, ??? for A64FX

- **Approach**
  - We decided to tune OneDNN for Fugaku's A64FX CPUs (OneDNN_aarch64) instead of full scratch development

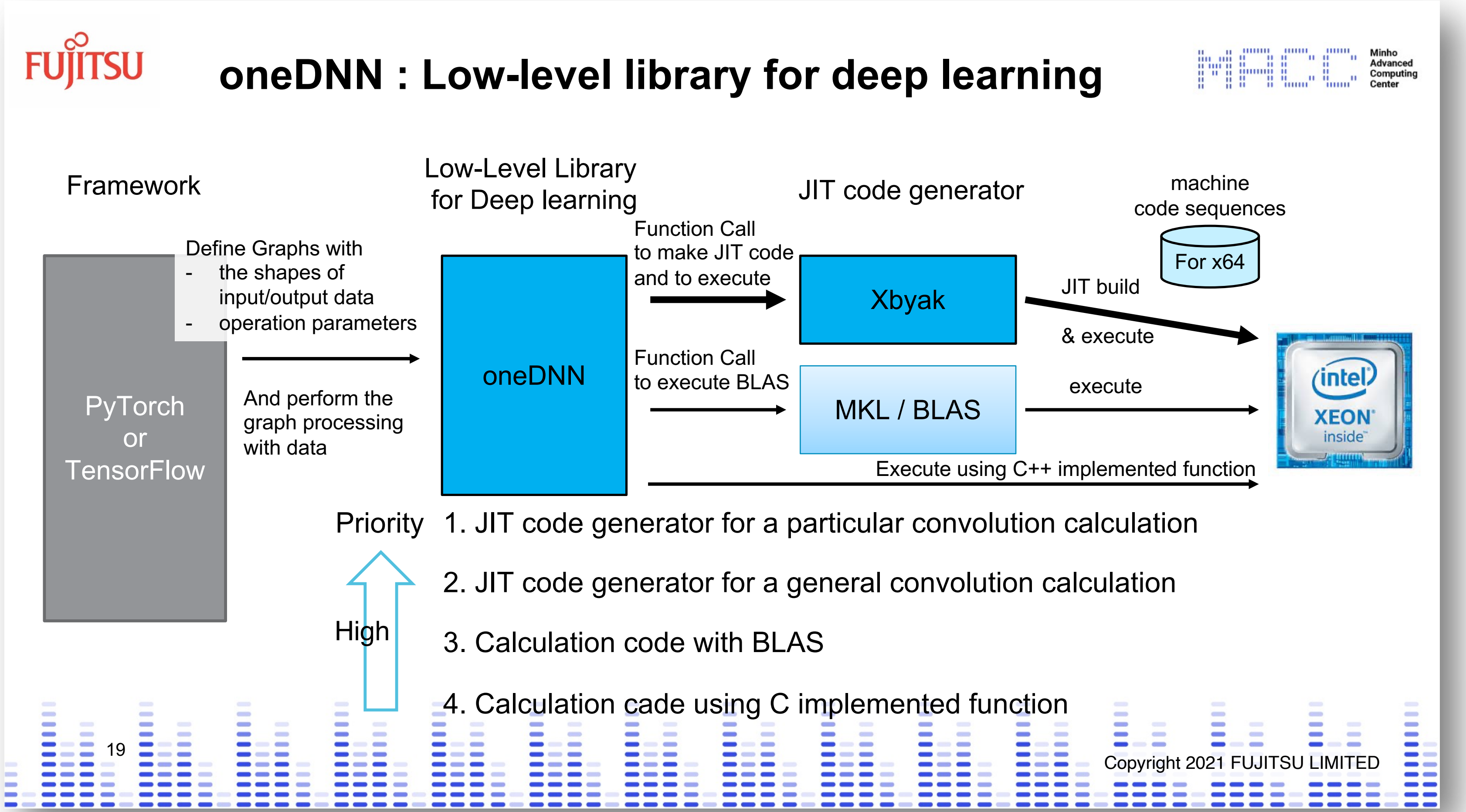
- **Current status**
  - The source codes are in a github repository
    - https://github.com/fujitsu/dnnl_aarch64
  - We also contribute to upstream of OneDNN repo



Intel Math Kernel Library for Deep Neural Networks (Intel MKL-DNN)
→ Deep Neural Network Library (DNNL)
→ oneAPI Deep Neural Network Library (oneDNN)

Slide courtesy of Jin Takahashi, Fujitsu laboratory ltd. with translation and modifications

Original oneDNN@Intel logic

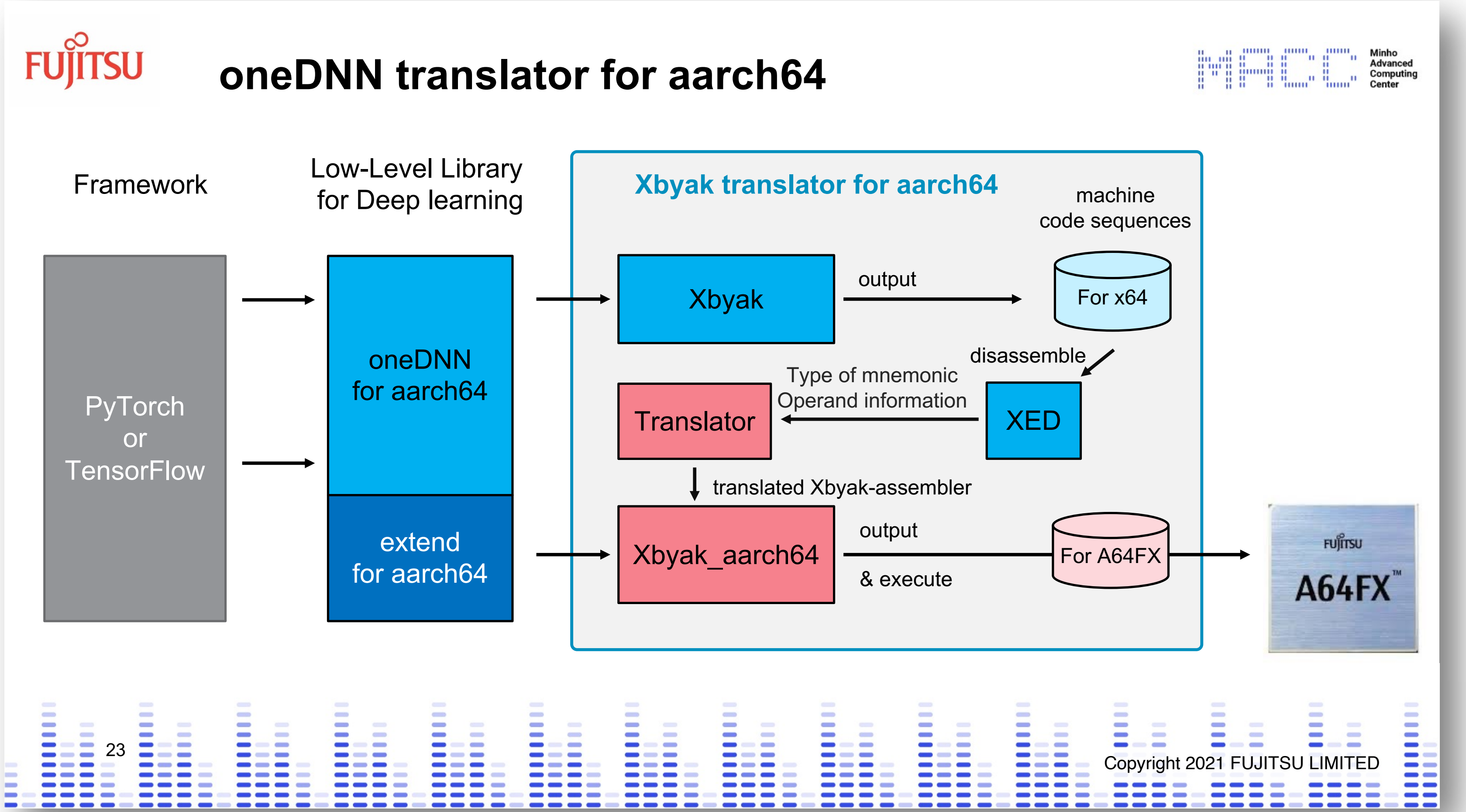


1. OneDNN gets information from a framework about (1) Shapes of input/output data; (2) Operation parameteers of each layer
2. OneDNN calls the fastest tensor routine based on the information
3. The priority is
   a) JIT-generated code
   b) BLAS
   c) C code implemented in OneDNN

- The generated code is cached and reused
- The same convolution kernels are called many time in deep learning
- The JIT-generation overhead is negligble for deep learning workloads

Slides: Masafumi Yamazaki (Fujitsu Ltd), “Deep learning on Fugaku”, MUG: MACC User Group Workshop, June 2021

Sustanable Porting Workflows



Slides: Masafumi Yamazaki (Fujitsu Ltd), "Deep learning on Fugaku", MUG: MACC User Group Workshop, June 2021

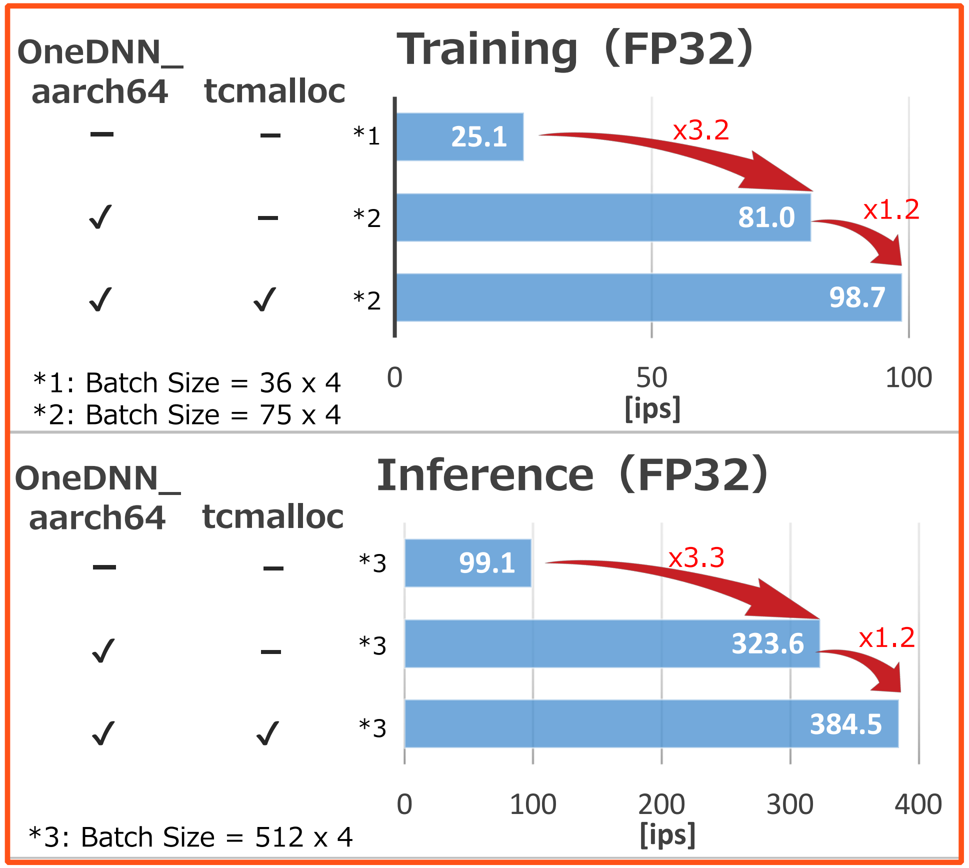
- By using the Xbyak, XED-Translator cascade, when the instruction set is extended, Xbyak and XED are replaced with the updated ones, and we only need to modify the mapping table between intel and Arm instructions in the Translator.

Perfomrance Evaluation: ResNet-50 on A64FX (A single node)

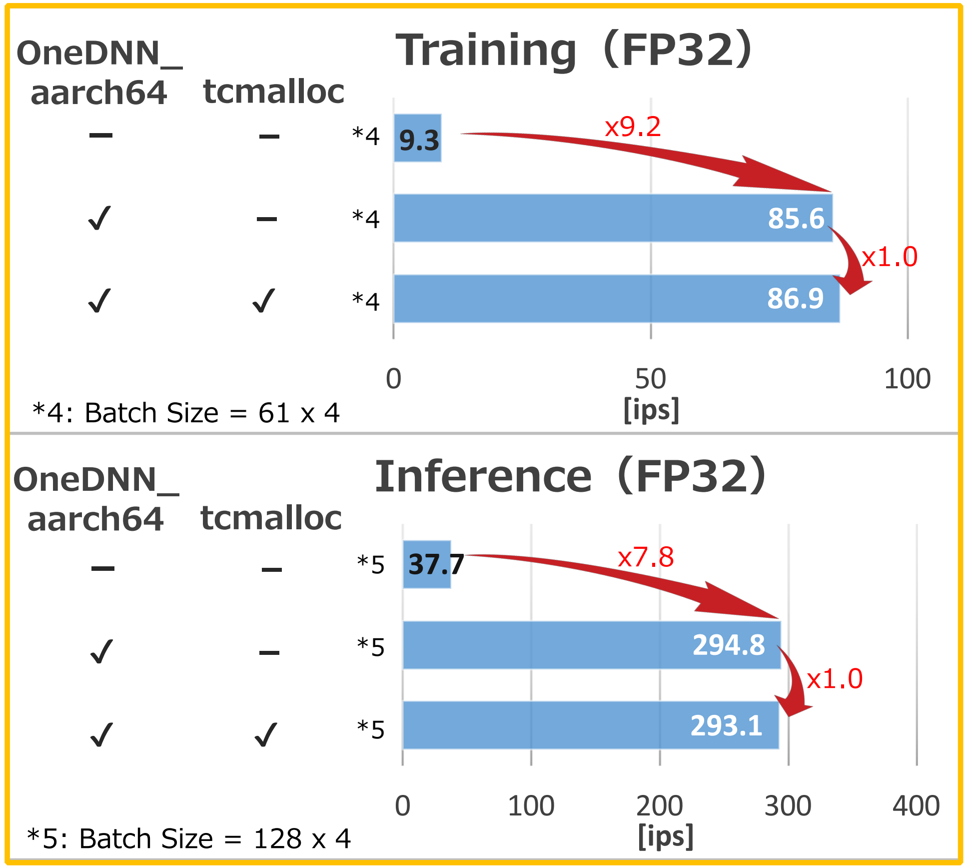
- **Environment**
  - HW: A64FX (2.2GHz, 48 cores, HBM2 32GB)
  - SW: Fujitsu compier (fcc), Fujitsu numerical libraries (SSL-II)

Ref.) NVIDIA GPU V100: 905 ips [1]
PyTorch/ResNet-50(training)/ImageNet2012

PyTorch v1.5.0



TensorFlow v2.1.0

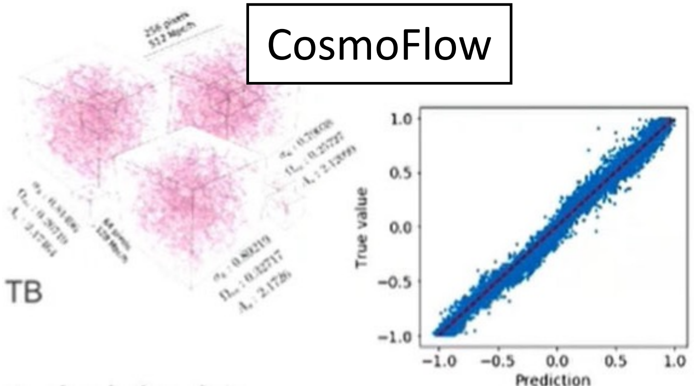


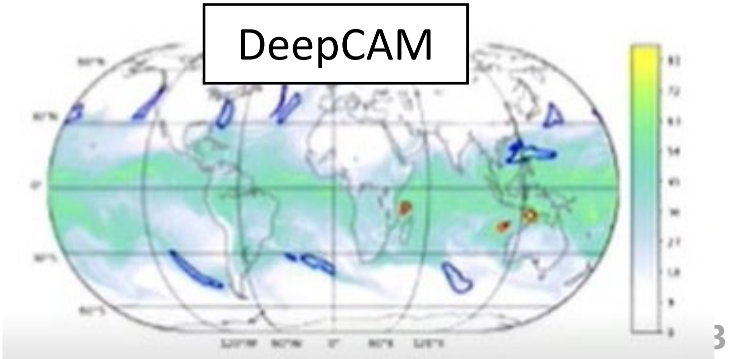
[1] NVIDIA Data Center Deep Learning Product Performance, https://developer.nvidia.com/deep-learning-performance-training-inference

Slide courtesy of Jin Takahashi, Fujitsu laboratory ltd. with translation and modifications

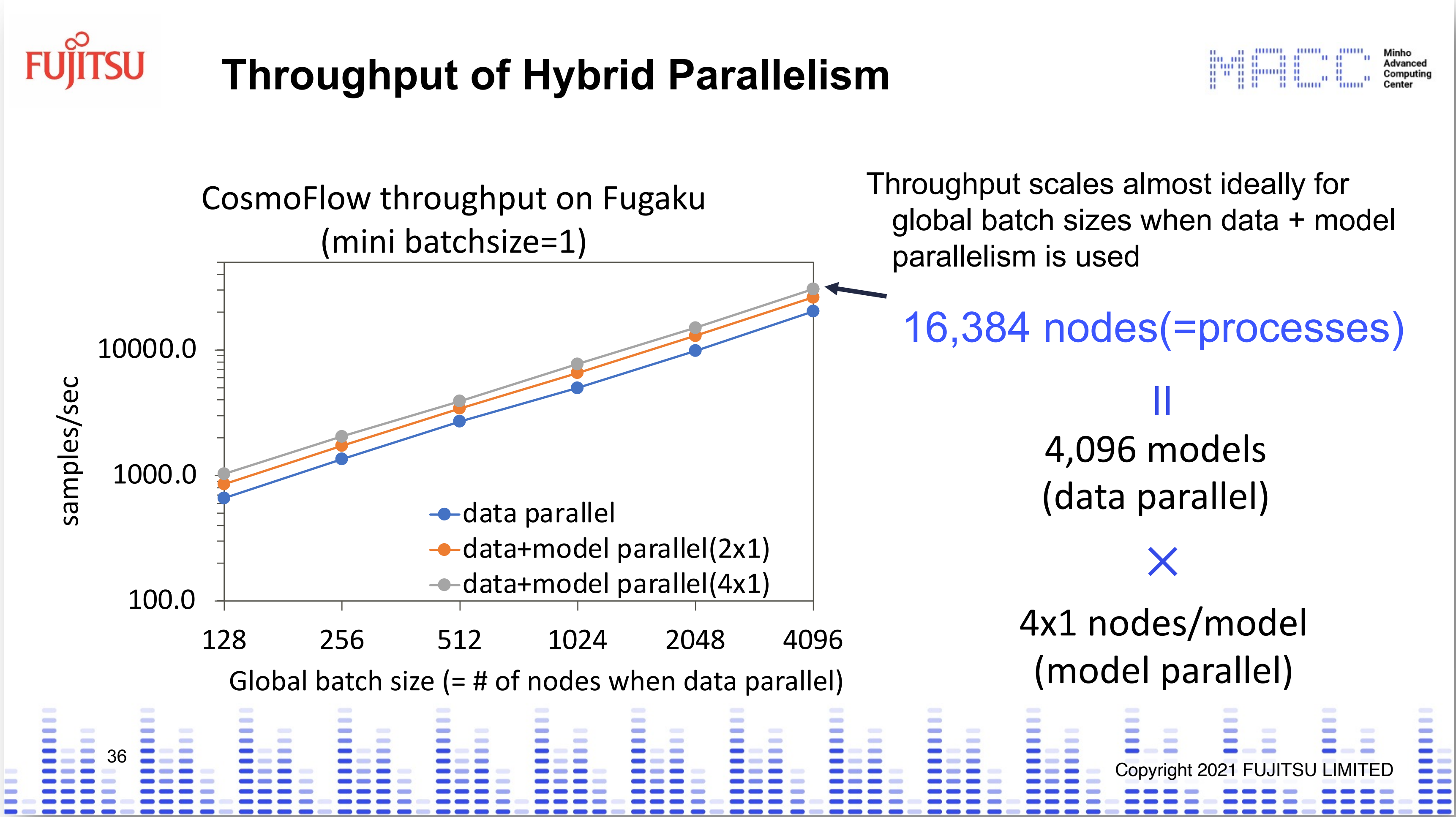
MLPerf HPC v0.7 (v1.0) Benchmark

- **MLPerf HPC (v0.7) Benchmark**
  - One of deep learning benchmarks in MLPerf HPC
  - Repository: https://github.com/mlcommons/hpc
  - Benchmarks
    - CosmoFlow（宇宙科学）
      - Predict cosmological parameters from N-body cosmo simulation data
      - 3D CNN for regression of 4 parameters
      - Training data shape is（128, 128, 128, 4)
      - Training data size is 5.1TB
    - DeepCAM（気候・気象）
      - Indentify extreme weather phenomena in climate simulation data
      - 2D semantic segmentation with DeepLabV3+ model which predicts 3 classes per pixel (atomaspheric river, tropical cyclon or background)
      - Training data shape is (768, 1152, 16) and labeled with 3 per-pixel classes
      - Training data size is 8.8 TB
    - Catalyst (v1.0)





Our process topology optimization enables scalable training



Slides: Masafumi Yamazaki (Fujitsu Ltd), "Deep learning on Fugaku", MUG: MACC User Group Workshop, June 2021

We achieved good scalability with a hybrid using of data&model parallel training

MLPerf HPC (v0.7) ranking: CosmoFlow

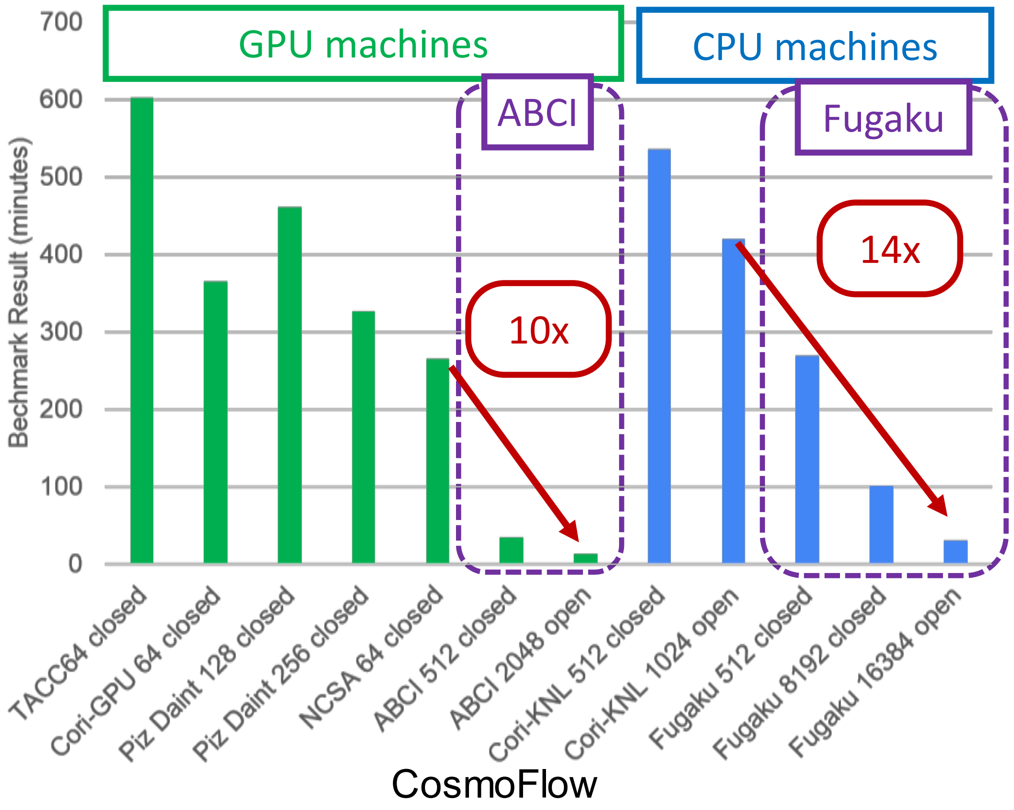
- **Fugaku was ranked at No 2. in MLPerf HPC ranking (Nov., 2020) even with "1/10 of Fugaku nodes"**
  - Fujitsu, AIST and RIKEN Achieve Unparalleled Speed on MLPerf HPC Machine Learning Processing Benchmark
    - https://www.hpcwire.com/off-the-wire/fujitsu-aist-and-riken-achieve-unparalleled-speed-on-mlperf-hpc-machine-learning-processing-benchmark/

Submitter	System	Processor	#	Accelerator	#	Software	Benchmark results (minutes, smaller is better) CosmoFlow	DeepCAM
e on-premises								
CSCS	daint_gpu_n128_tf2.2.0	Intel® Xeon® Processor E5-2690 v3 @2.60	128	NVIDIA P100-PCIE-16G	128	TensorFlow 2.2.0	461.01	
CSCS	daint_gpu_n256_tf2.2.0	Intel® Xeon® Processor E5-2690 v3 @2.60	256	NVIDIA P100-PCIE-16G	256	TensorFlow 2.2.0	327.01	
Fujitsu	ABCI PRIMERGY CX2570 M4	Intel® Xeon® Gold 6148 Processor @2.40	512	NVIDIA V100	1024	PyTorch 1.6.0		11.71
Fujitsu	ABCI PRIMERGY CX2570 M4	Intel® Xeon® Gold 6148 Processor @2.40	256	NVIDIA V100	512	TensorFlow 2.2.0	34.42	
Fujitsu/RIKEN	fugaku_512xA64FX_tensorflow_closed	FUJITSU Processor A64FX	512	N/A	0	TensorFlow 2.2.0 + Mesh TensorFlow	268.77	
Fujitsu	fugaku_8192xA64FX_tensorflow_closed	FUJITSU Processor A64FX	8192	N/A	0	TensorFlow 2.2.0 + Mesh TensorFlow	101.49	
LBNL	corigpu_n64_pt1.6.0	Intel® Xeon® Gold 6148 Processor @2.40	16	NVIDIA V100	64	PyTorch 1.6.0		139.29
LBNL	corigpu_n64_tf1.15.0	Intel® Xeon® Gold 6148 Processor @2.40	16	NVIDIA V100	64	TensorFlow 1.15.0	364.73	
LBNL	coriknl_n512_tf1.15.2	Intel® Xeon Phi™ Processor 7250 @1.40G	512	N/A	0	TensorFlow 1.15.2	536.06	
NCSA	hal_v100_n16_tf1.15.0	IBM POWER 9 model 2.2	32	NVIDIA V100	64	TensorFlow 1.15.0	265.59	
TACC	Frontera-RTX	Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10G	32	NVIDIA Quadro RTX 50	64	TensorFlow 1.15.2	602.23	

Division Times

Submitter	System	Processor	#	Accelerator	#	Software	Benchmark results (minutes, smaller is better) CosmoFlow	DeepCAM
e on-premises								
Fujitsu	ABCI PRIMERGY CX2570 M4	Intel® Xeon® Gold 6148 Processor @2.40	512	NVIDIA V100	1024	PyTorch 1.6.0		10.49
Fujitsu	ABCI PRIMERGY CX2570 M4	Intel® Xeon® Gold 6148 Processor @2.40	1024	NVIDIA V100	2048	TensorFlow 2.2.0	13.21	
Fujitsu	fugaku_16384xA64FX_tensorflow_open	FUJITSU Processor A64FX	16384	N/A	0	TensorFlow 2.2.0 + Mesh TensorFlow	30.07	
LBNL	coriknl_n1024_tf1.15.2	Intel® Xeon Phi™ Processor 7250 @1.40G	1024	N/A	0	Tensorflow 1.15.2	419.69	

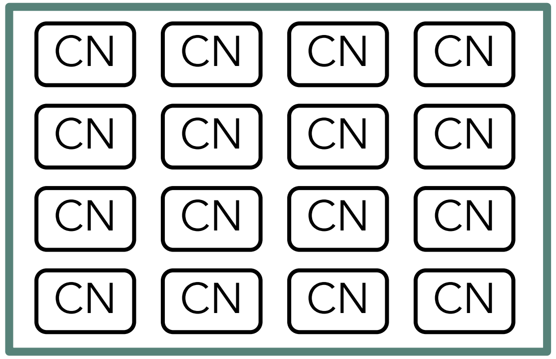
Submitter	System	Processor	#	Software	Time [min]
Fujitsu	ABCI	Xeon Gold 6148 Tesla V100 GPU	1024 2048	TensorFlow	13.21
Fujitsu / RIKEN	Fugaku	A64FX	16384	TensorFlow + Mesh TensorFlow	30.07



MLPerf HPC (v1.0) introduced scalability rules

- MLPerf HPC v0.7 in FY2020: Strong scaling metric
  - Strong scaling metric
    - Measures time to train one model on a system
  - Due to the large-batch problem, 1/10 of Fugaku nodes give the best performance
  - Benchmarks: CosmoFlow, DeepCAM

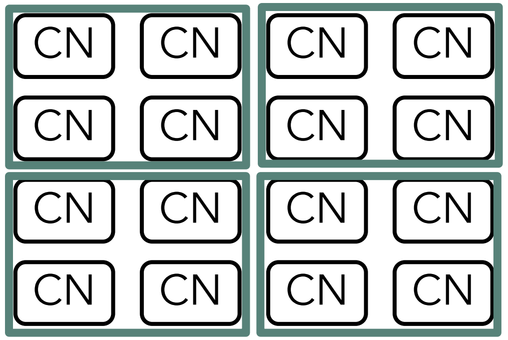
Training one model



- MLPerf HPC v1.0 in FY2021: Strong + Weak scaling metric
  - v1.0 introduces a new weak scaling metric (in addition to strong scale metric)
    - Time-to-train → Throghputs (models/second)
  - Weak scaling metric
    - Train multiple models on a system and measure # of trained models per sec
    - Models are independently trainined eath other ans it is scalable
    - We could use 1/2 of Fugaku nodes
  - Benchmarks: CosmoFlow, DeepCAM and Catalyst
  - Six metrics: {CosmoFlow, DeepCAM and Catalsyt} x {Strong, Week}

→ We targeted CosmoFlow & Week scaling metric

Training multiple models



MLPerf HPC v1.0 result (CosmoFlow & Weak scaling metric)

- Fugaku took first place in MLPerf HPC v1.0 (CosmoFlow, weak scaling metric)
  - 82,944 CPUs are used (128 CPUs per model instance)
  - Trained 637 models in 8.26 hours (495.6 mins)
  - → 1.285 modesl / min

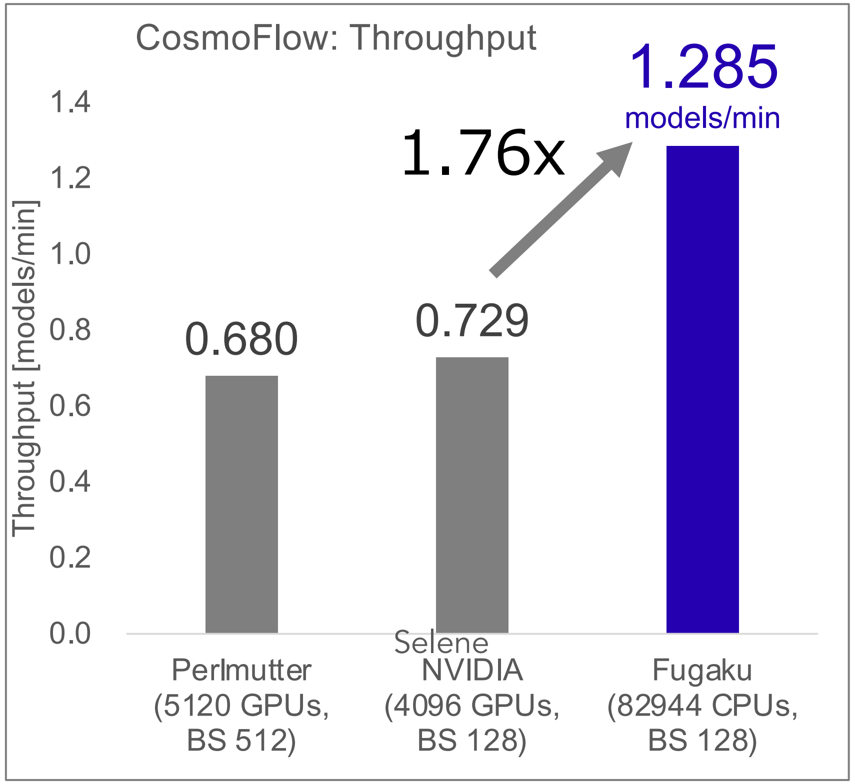
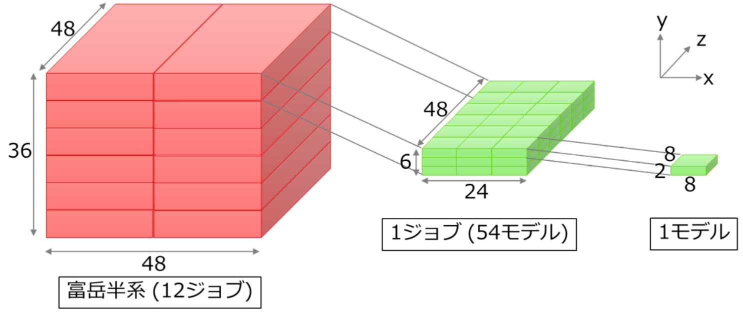
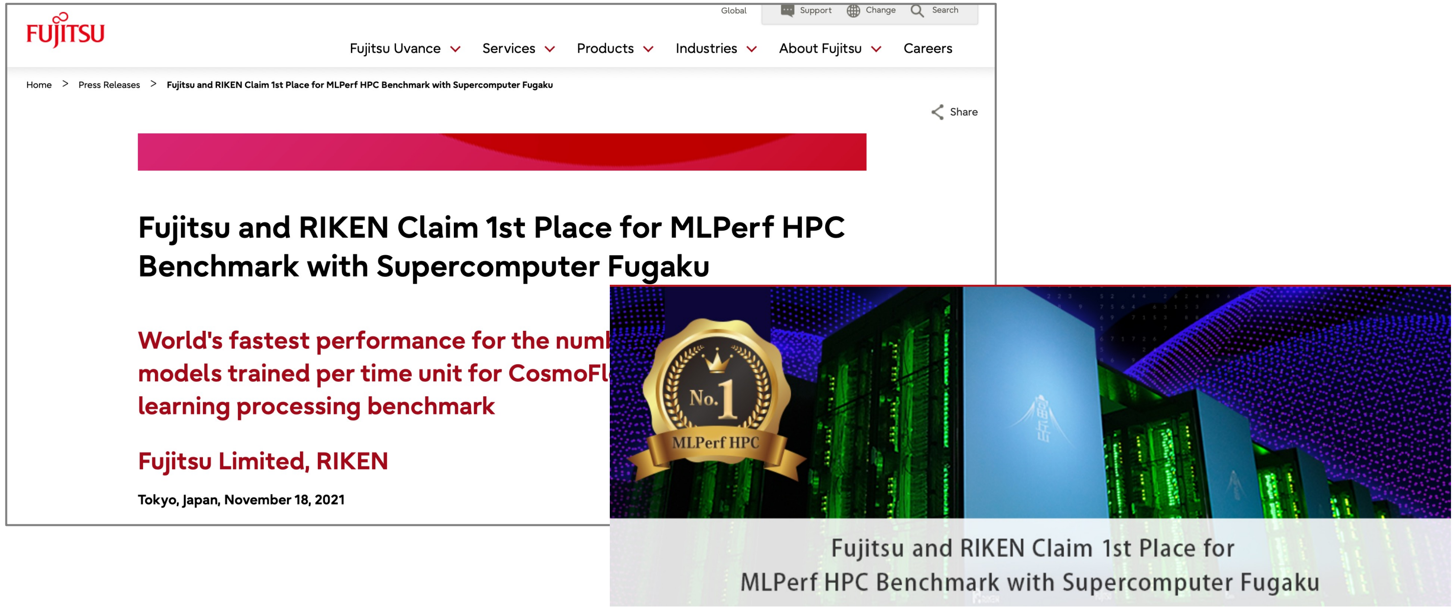


Figure from Koichi Shirahata (Fujitsu Ltd) presentation at SC21 BoF (MLPerf HPC)



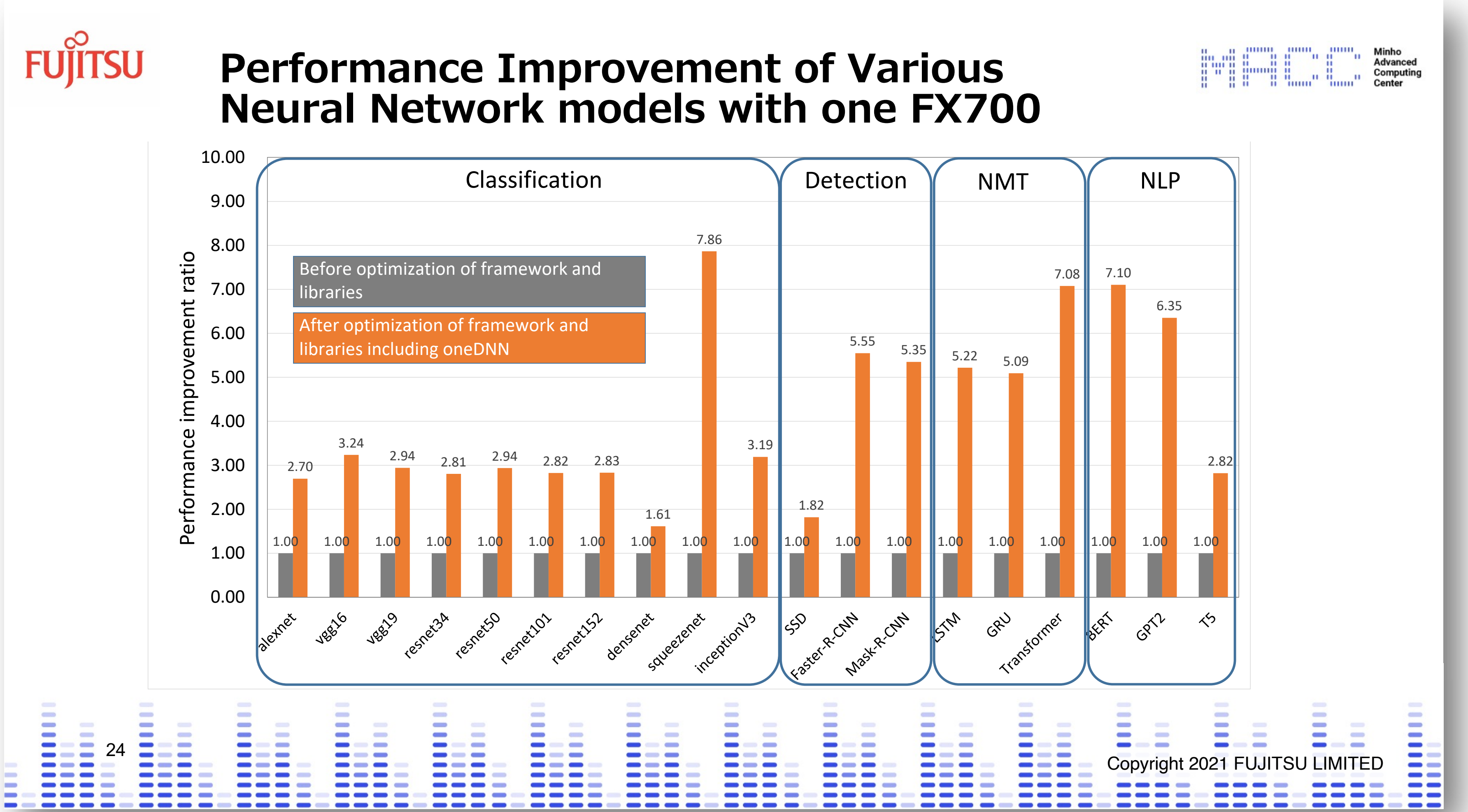
12 x 54 = 648 models



Source: https://www.fujitsu.com/global/about/resources/news/press-releases/2020/1119-02.html

Performance results on other neural networks

- With tuned oneDNN for A64FX, we achieve1.6x to 7.8x performance improvement



Slides: Masafumi Yamazaki (Fujitsu Ltd), "Deep learning on Fugaku", MUG: MACC User Group Workshop, June 2021

Supported TensorFlow and PyTorch versions on Fugaku

- Fugaku officially supports **TensorFlow-2.2.0**, **PyTorch-1.7.0**/1.6.0. These versions are linked to the Fujitsu's oneDNN library tuned for A64FX
- **Location**
  - FEFS storage : /home/apps/oss/...
- **Package versions**

環境			モデル対応					提供状況	
FW	OneDNN	Horovod	ResNet50	OpenNMT	ResNetX	BERT	Mask-RCNN	理研様提供	Fujitsu github 公開
PT v1.5.0	v0.21.0	v0.19.0	✓					✓	✓
PT v1.6.0	v1.6.0	v0.20.3	✓	✓	✓			✓	✓
PT v1.7.0	v2.1.0	v0.20.3	✓	✓	✓	✓		–	✓
PT v1.7.0	v2.1.0L01	v0.20.3	✓	✓	✓	✓	✓	–	✓
TF v2.1.0	v0.21.2	v0.19.5	✓					✓	✓
TF v2.2.0	v2.1.0	v0.19.5	✓	✓	✓	✓		–	✓
TF v2.2.0	v2.1.0L01	v0.19.5	✓	✓	✓	✓	✓	–	✓

  - Other: Python ver.3.8.2 + mpi4py ver.3.0.3, pandas ver.1.2.2, numpy ver.1.19.0, scipy ver.1.5.2, h5py ver.2.8.0, libtensorflow_cc.so ver.2.2.0, Batched BLAS ver.1.0, fapp ver.1.0.0 etc.

Summary

- Data pratform is important for data-drive science
  - We launched a project to build data pre-processing/compression/analysis/utilization platform for RCS facilities (SPring-8, SACLA) and Fugaku
  - For data compression, we introduced TEZip for fast data transfer
    - AI-driven data compression tool designed for time evolutionary data
    - Compression rates are up to 15 in the lossless mode and 50 in the lossy mode in the real SPring-8 data
- DL4 Fugaku Project
  - We extended OneDNN library for A64FX by developoing the Xbyak translator
  - In MLPerf HPC v1.0 (CosmoFlow), Fugaku recieved No. 1 in the weak scaling metric
  - We also tuned many other NNs such as data classification, detection, NMT and NLP
- Working with the operation team, we woulid like to enahcen the usability of Fugaku and other systems

Our team is seeking for researchers, postdocs and Ph.D. students.
If you are interested in joining our projects, pleasae feel free to contact me: kento.sato@riken.jp