DL4Fugaku: Deep Learning for Fugaku

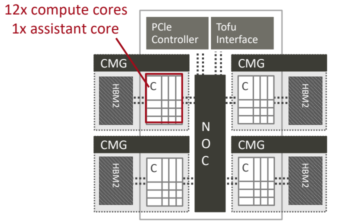
Deep learning for Fugaku

- 近年、大規模深層学習は、多くの技術的課題を解決するために不可欠な機械学習技法とし注目（画像認識[1]、音声認識[2] etc.）
- より多くの学習データをより短い時間でNNモデルを構築するために高速かつスケーラブルな大規模深層学習が重要
- AIプラットフォームとして「富岳」が期待

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



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

→ 高性能 FP16/INT8

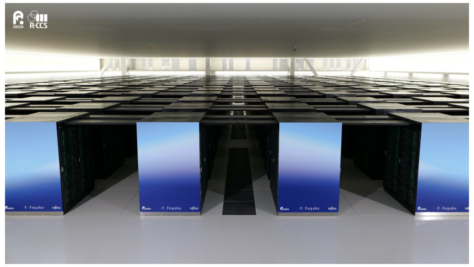
→ 高メモリバンド幅 (1024 GB/sec)

→ スケーラブルなTofuDインターコネクト

「富岳」のもつハードウェアスペックをフルに引き出すためには、大規模深層学習ソフトウェアのチューニングが不可欠

[1] R. Wu, S. Yan, Y. Shan, Q. Dang, and G. Sun. Deep image: Scaling up image recognition. *arXiv:1501.02876*, 2015.
[2] G. E. Dahl, D. Yu, L. Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Transactions on Audio, Speech, and Language Processing, 20(1), 2012.

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
  - (Now →) Enhance the usability for production use in Fugaku

- **MOU for RIKEN/Fujitsu collaboration on AI framework development in Fugaku**

- **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
- 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)

Nov. 25th, 2019

Right : Naoki Shinjo, Head of Unit. Platform Development Unit. Fujitsu Limited
Left : Satoshi Matsuoka, R-CCS Director

※ 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: Deep learning for Fugaku

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

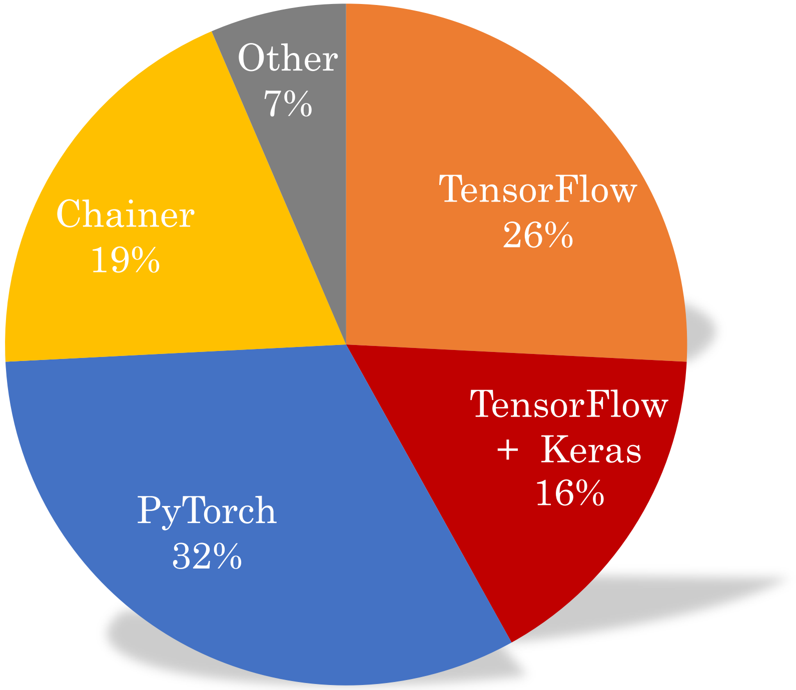
Technical support

Shigeo Mitsunari (Xbyak)

スーパーコンピュータ「富岳」に向けた 深層学習フレームワーク利用調査

- **実施期間**
  - 2019年10月から11月
- **利用調査機関/ユーザー**
  - RIKEN R-CCS
  - RIKEN AIP
  - HPCI 重点課題
  - ABCI ユーザー

→ 既に既存のシステム上でAI学習・推論計算を行なっているユーザが回答



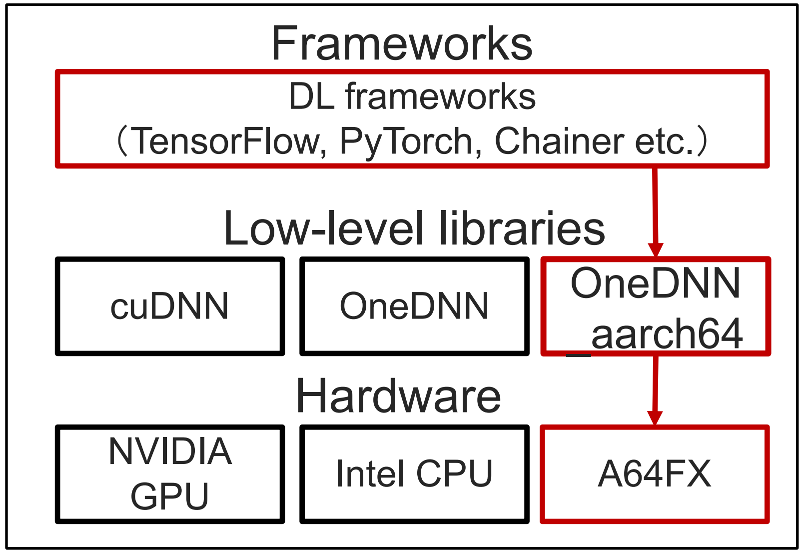
※ "Other" users develop and use their own DL frameworks

Porting and Tuning approach

- **Deep learning software stack**
  - Deep learning frameworks are relying 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

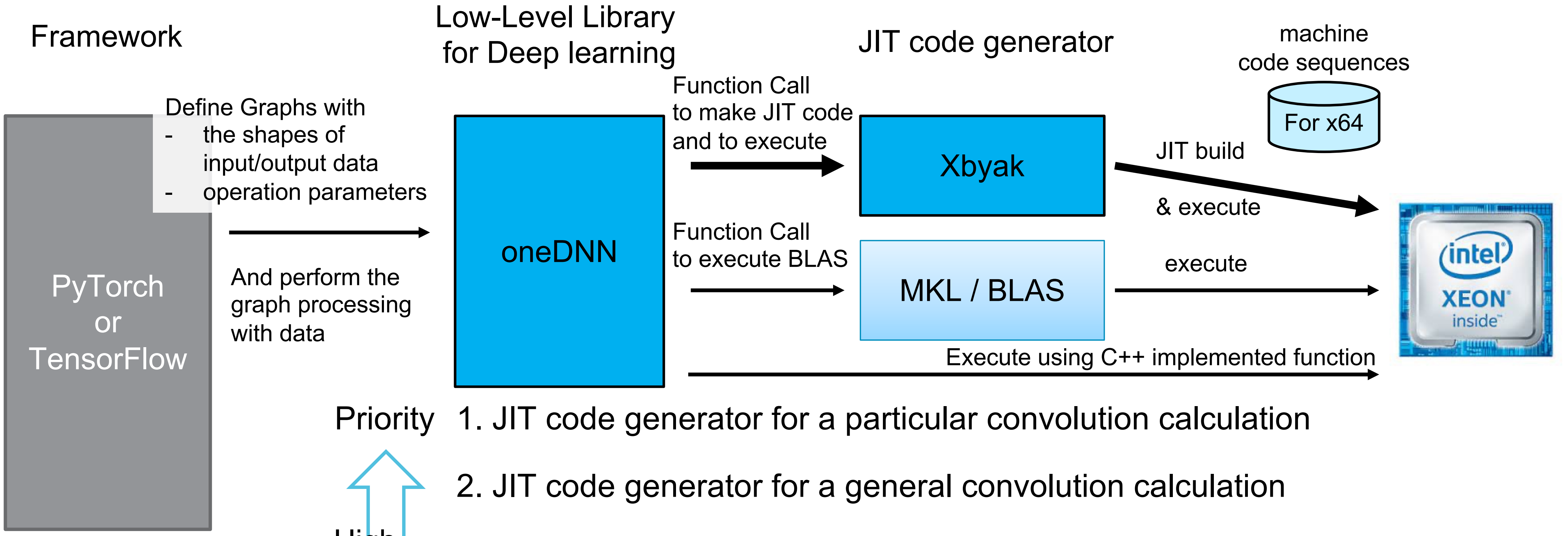
- **Current status**
  - Porting and tuning are finished
  - The source codes are in a github repository
    - https://github.com/fujitsu/dnnl_aarch64
  - OneDNN_aarch64 is in a maintenance phase



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

oneDNN : Low-level library for deep learning



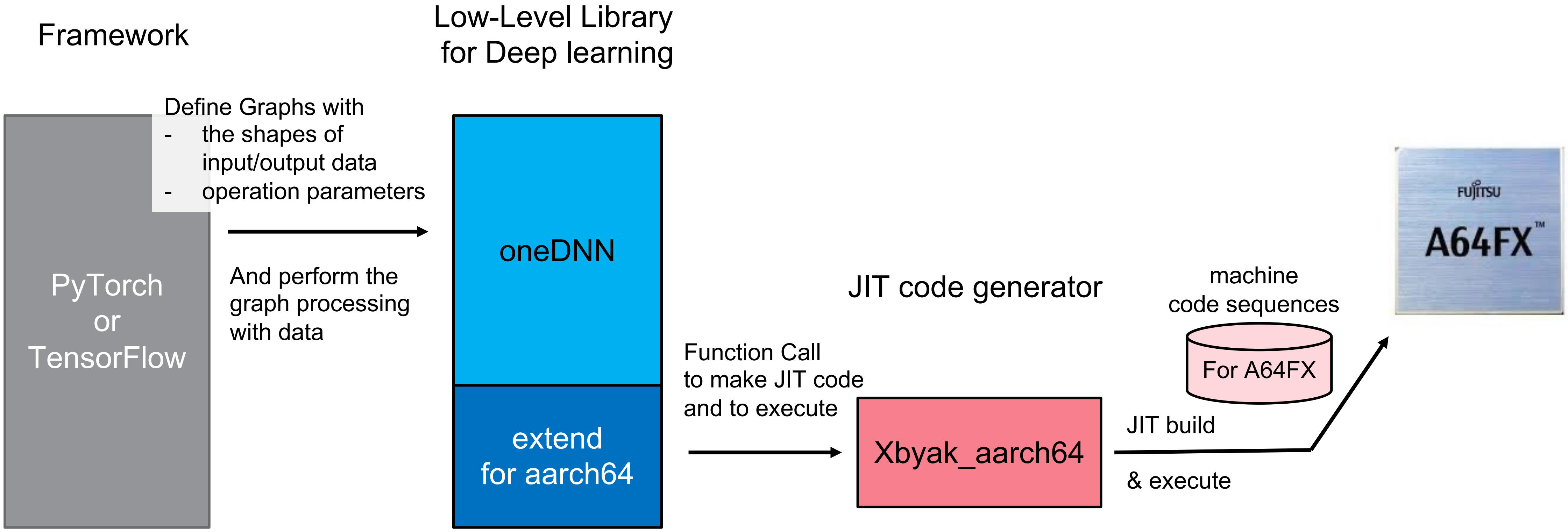
Priority

High

1. JIT code generator for a particular convolution calculation
2. JIT code generator for a general convolution calculation
3. Calculation code with BLAS
4. Calculation cade using C implemented function

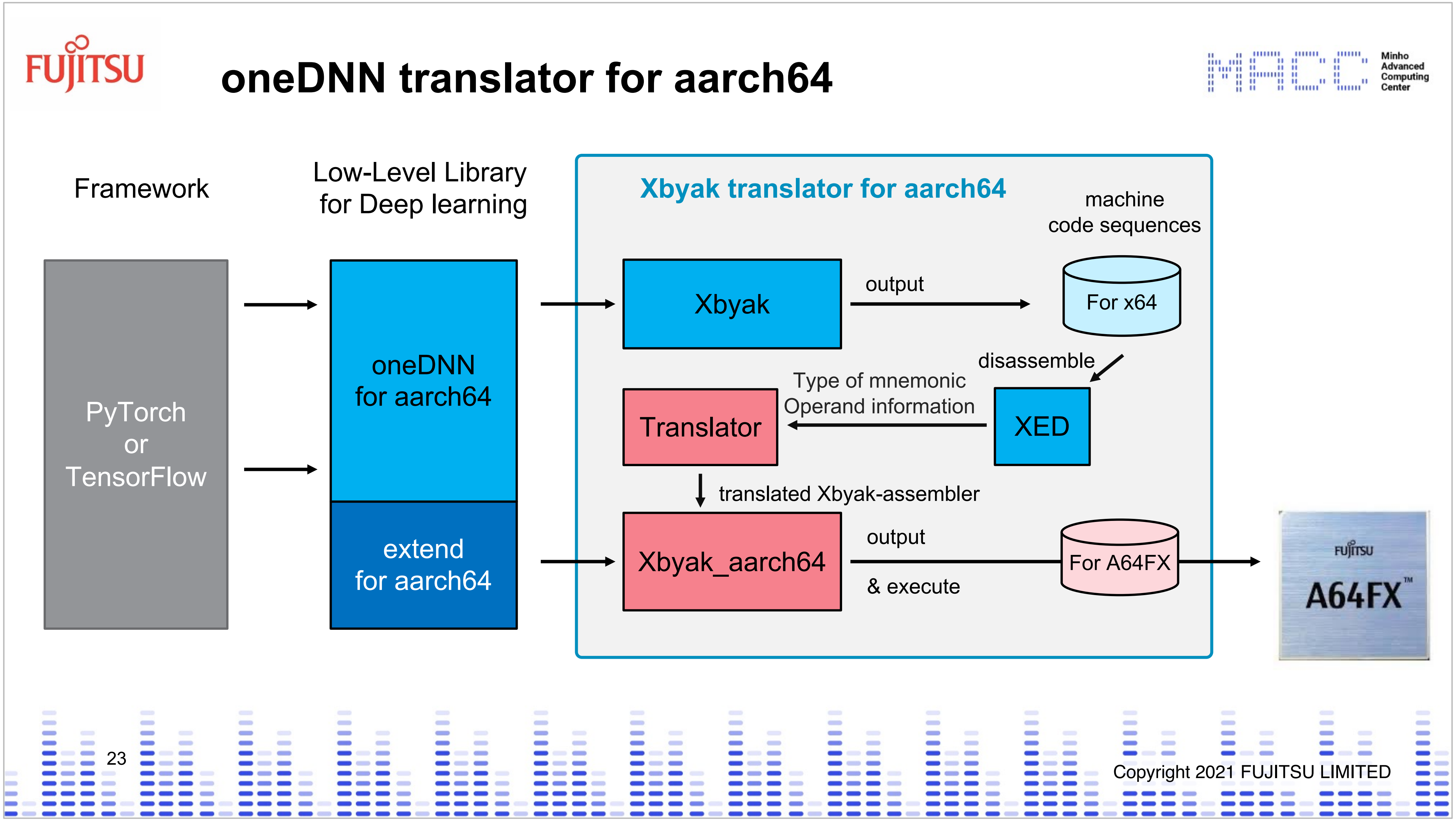
Slides: Masafumi Yamazaki (Fujitsu Ltd), “Deep learning on Fugaku”, MUG: MACC User Group Workshop, June 2021

oneDNN for aarch64



We extended oneDNN with JIT code for aarch64.

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.

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

FUJITSU

C++ code of oneDNN using Xbyak, Xbyak aarch64

Minho Advanced Computing Center

oneDNN JIT generation code for x86

```
…
for (int jj = jj_start; jj < jj_end; jj += stride_w)
    if (jcp.kernel_kind == expl_bcast)
        vfmadd231ps(vmm_out(jj, ii),
                vmm_inp(jj, nb_ic_block),
                vmm_wei);
    else
        vfmadd231ps(vmm_out(jj, ii),
                vmm_wei,
                EVEX_compress_addr(aux_reg_dst,
                    get_dst_offset(jj, oc, ki), true));
…
```

oneDNN JIT generation code for A64FX

```
…
for (int jj = jj_start; jj < jj_end; jj += stride_w) {
    if (stride_w == 1) {
        fmla(zreg_out_s(jj, ii), reg_p_all_ones,
                zreg_inp_s(jj, nb_ic_block),
                zreg_wei_s(wei_count));
    } else {
        int aux_output_offset = get_dst_offset(jj, oc, ki);
        prev_ofs = bcast_load_sw(jj, nb_ic_block,
                aux_output_offset, prev_ofs, jj_end);
        fmla(zreg_out_s(jj, ii), reg_p_all_ones,
                zreg_inp_s(jj % stride_w, nb_ic_block),
                zreg_wei_s(wei_count));
…
```

machine
code sequences
(disassembled)

```
184:    8592411e    ldr     z30, [x8, #144, mul vl]
188:    65bf0f00    fmla    z0.s, p3/m, z24.s, z31.s
18c:    65bf0f21    fmla    z1.s, p3/m, z25.s, z31.s
190:    65bf0f42    fmla    z2.s, p3/m, z26.s, z31.s
194:    65bf0f63    fmla    z3.s, p3/m, z27.s, z31.s
198:    65bf0f84    fmla    z4.s, p3/m, z28.s, z31.s
19c:    65bf0fa5    fmla    z5.s, p3/m, z29.s, z31.s
```

Ref. oneDNN/src/cpu/x64/jit_avx512_common_conv_kernel.cpp
oneDNN/src/cpu/aarch64/jit_sve512_conv_kernel.cpp:

22

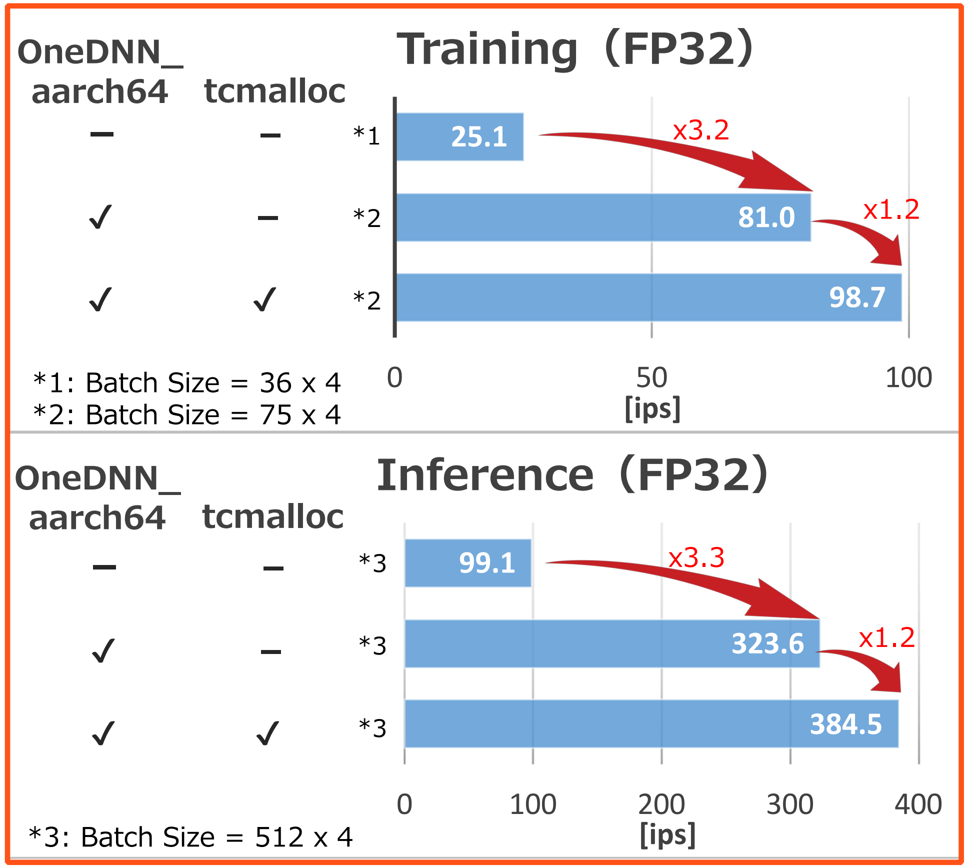
Slides: Masafumi Yamazaki (Fujitsu Ltd), “Deep learning on Fugaku”, MUG: MACC User Group Workshop, June 2021

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

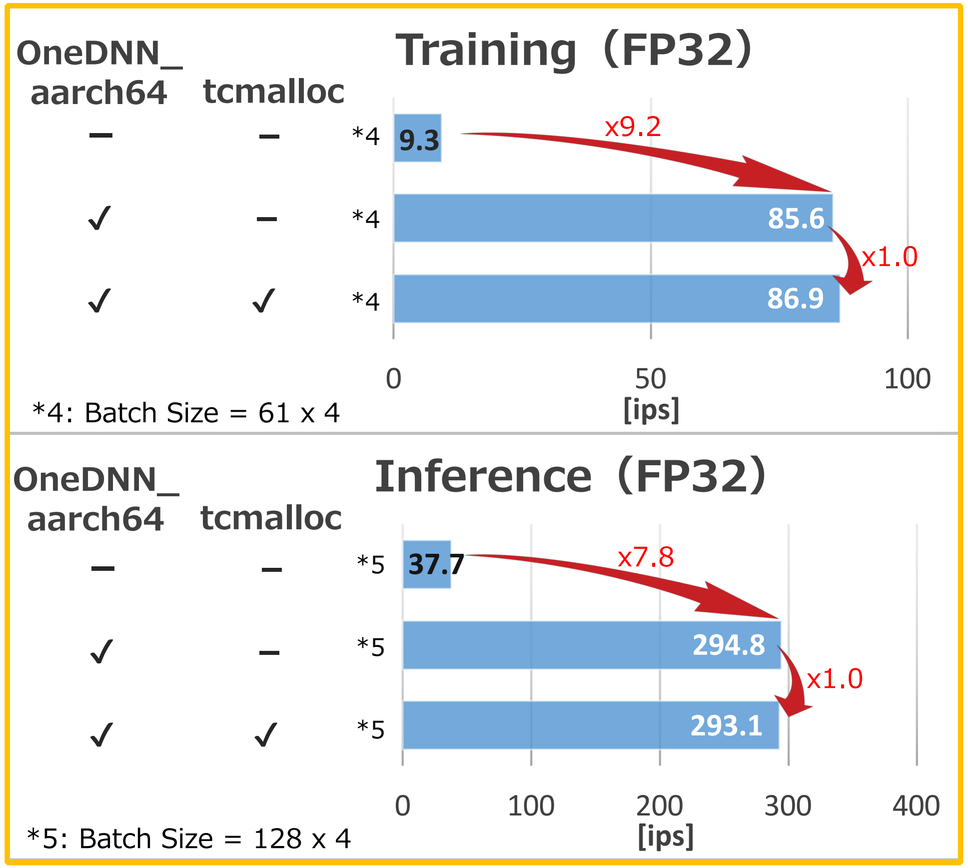
- **Environment**
  - HW: A64FX (2.2GHz, 48 cores, HBM2 32GB)
  - SW: Fujitsu compier (fcc), Fujitsu numerical libraries (SSL-II)

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

PyTorch v1.5.0



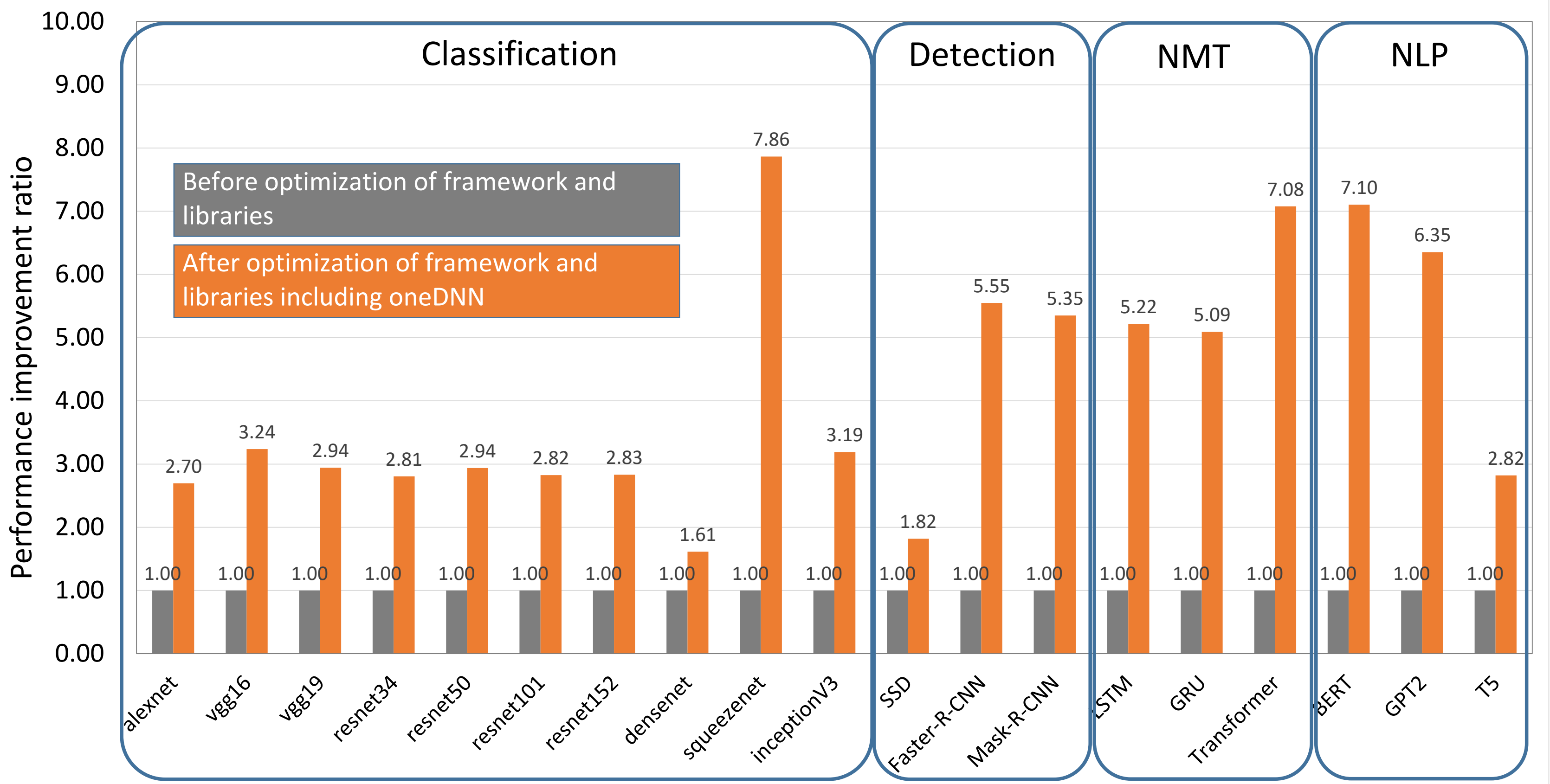
TensorFlow v2.1.0



[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

Performance Improvement of Various Neural Network models with one FX700



24

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

MLPerf HPC Benchmark

- **MLPerf HPC Benchmark**
  - One of deep learning benchmarks in MLPerf
  - The latest version is v0.7 (as of November 2020)
  - Repository: https://github.com/mlcommons/hpc

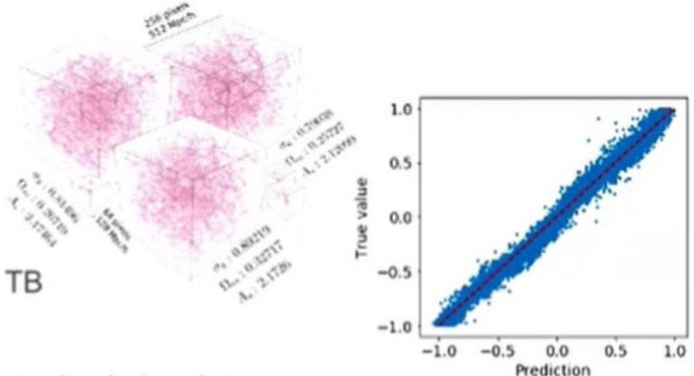
- **Measurement**
  - MLPerf HPC measures time to convergence to the certain inference accuracy

- **Fujitsu/RIKEN submitted to the MLPerf HPC Benchmark ranking (in Nov. 2020)**
  - Other submitters: TACC, NERCS(Cori), NCSA, Piz Daint etc.

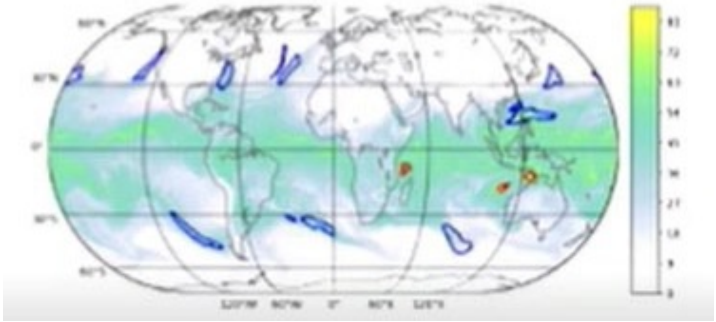
MLPerf HPC Benchmark (contn'd)

- **MLPerf HPC Benchmark has two applications**
  - 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 whic predicts 3 classes per pixel (atomaspheric river, tropical cyclon or background)
    - Training data shape is (768, 1152, 16), 3 per-pixel classes
    - Training data size is 8.8 TB

CosmoFlow



DeepCAM



MLPerf HPC ranking: CosmoFlow

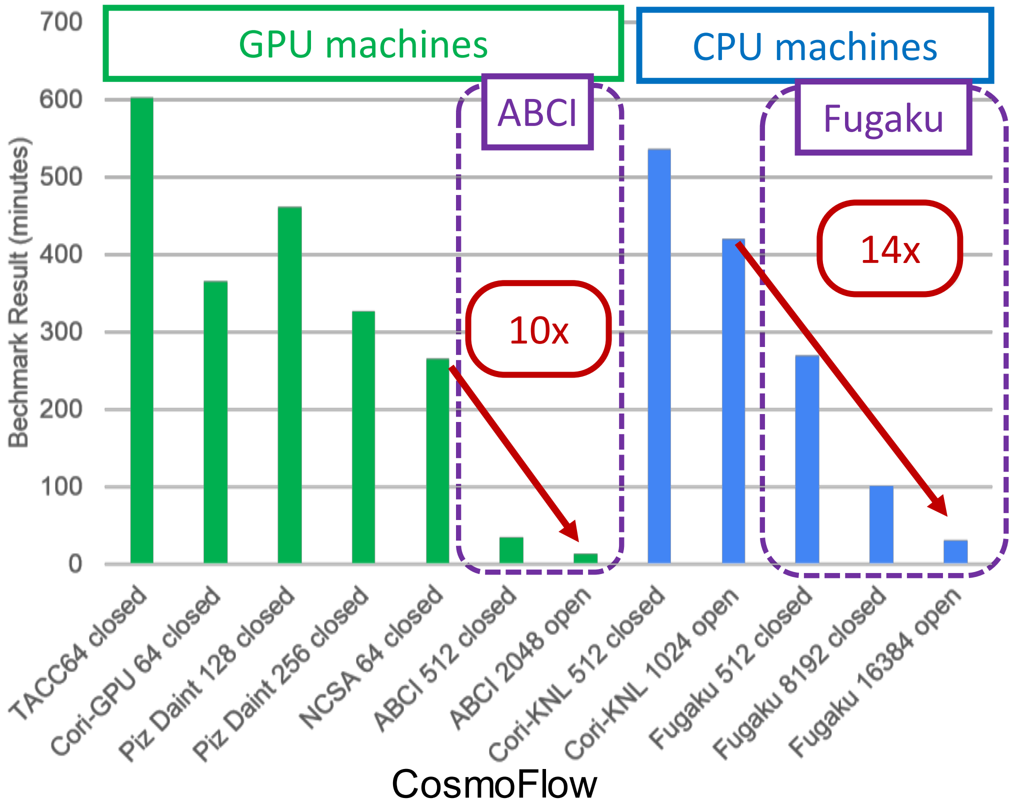
- **Fugaku was ranked at No 2. in MLPerf HPC ranking (Nov., 2020) even with "1/10 of Fugaku nodes"**
  - Fujitsu, AIST and RIKEN Achieve Unparalleled Speed on MLPerf HPC Machine Learning Processing Benchmark
    - https://www.hpcwire.com/off-the-wire/fujitsu-aist-and-riken-achieve-unparalleled-speed-on-mlperf-hpc-machine-learning-processing-benchmark/

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	

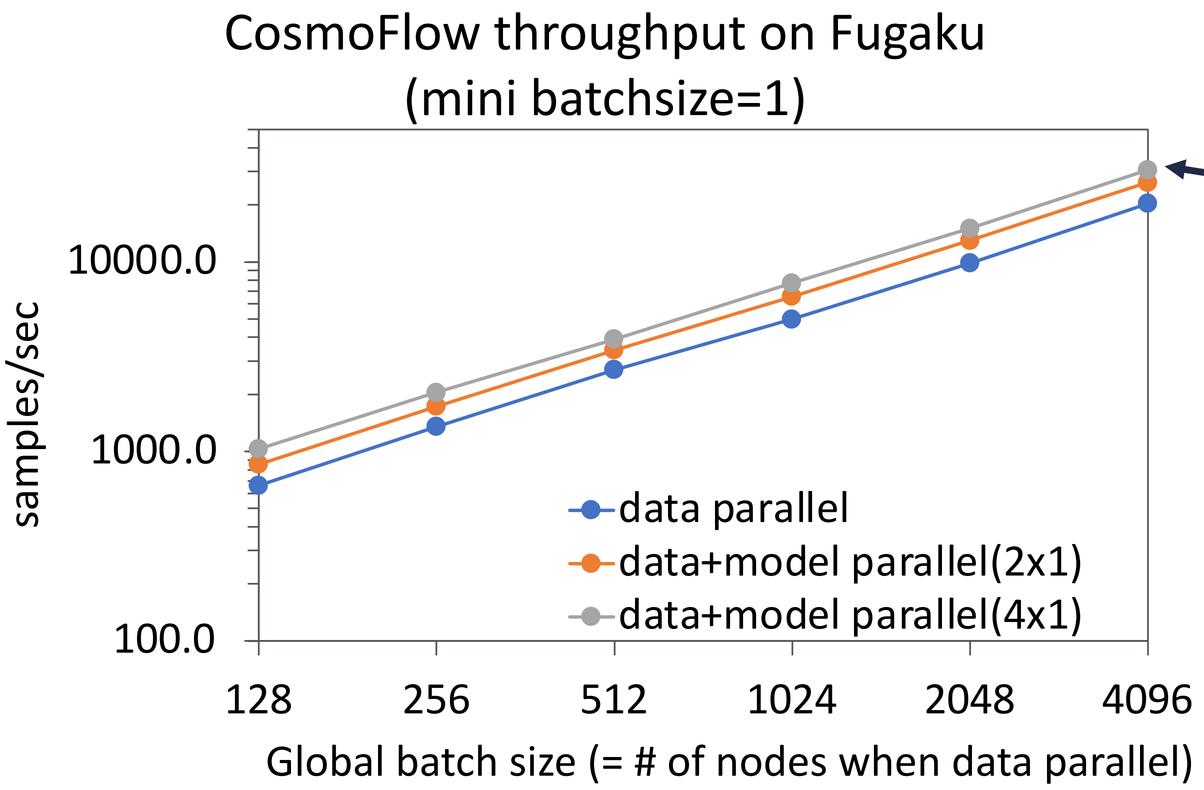
Division Times

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



Throughput of Hybrid Parallelism



Throughput scales almost ideally for global batch sizes when data + model parallelism is used

16,384 nodes(=processes)

||

Batch size 4,096

×

4x1 partitioning

全系富岳（N並列 and 1並列xN）実行における学習速度では世界最速のDNN実行環境であると期待

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

Use of AL frameworks on Fugaku

「富岳」で利用できるAIフレームワーク

「富岳」共用開始に併せてAIフレームワーク**TensorFlow-2.2.0**, **PyTorch-1.7.0**/1.6.0を正式に提供を開始しました． 富士通研究所が開発した「富岳」向け**高速化ライブラリoneDNN**を組み込んでいます

- **インストール場所**
  - 「富岳」第2階層ストレージファイルシステム上：/home/apps/oss/...
- **インストールパッケージバージョン**

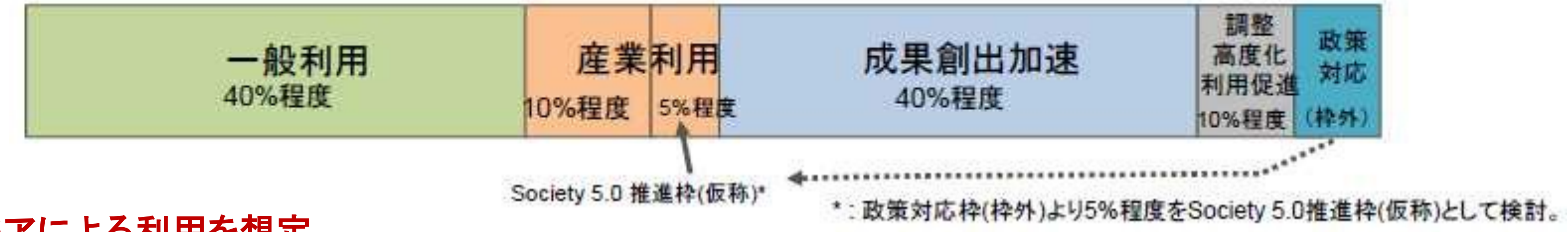
環境			モデル対応					提供状況	
FW	OneDNN	Horovod	ResNet50	OpenNMT	ResNetX	BERT	Mask-RCNN	理研様提供	Fujitsu github 公開
PT v1.5.0	v0.21.0	v0.19.0	✓					✓	✓
PT v1.6.0	v1.6.0	v0.20.3	✓	✓	✓			✓	✓
PT v1.7.0	v2.1.0	v0.20.3	✓	✓	✓	✓		–	✓
PT v1.7.0	v2.1.0L01	v0.20.3	✓	✓	✓	✓	✓	–	✓
TF v2.1.0	v0.21.2	v0.19.5	✓					✓	✓
TF v2.2.0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

スーパーコンピュータ「富岳」利活用の基本方針（2020年7月17日文科省通知）

□ 計算資源配分の考え方

第45回文科省HPCI計画推進委員会(2020年12月9日)資料より



■ 一般利用
- 主としてアカデミアによる利用を想定。
- 公募により、「富岳」の機能・性能を有効に活用する、幅広い研究課題を科学的見地から審査した上で、採択。

■ 産業利用
- 産業界による利用を想定。
- 公募により、「富岳」の機能・性能を有効に活用する、幅広い課題を科学的、社会経済的見地から審査した上で、採 択 。
- Society5.0 の実現に資する課題を実施する枠（Society5.0推進枠（仮称））を設ける。（ 例：産業界のコンソーシアム、産学連携 による利用などを想定）。

高度情報科学技術研究機構 （RIST）への利用申請・採択後利用可能に

■ 成果創出加速
- 「富岳」成果創出加速プログラムで採択された課題等、特に、科学的・社会的課題の解決に直結する成果の創出が早 期に見込める研究課題を実施。

■ 調整・高度化・利用拡大
- 「富岳」の運用機関であるR-CCSが中心となって、安定運用のためのシステム調整に必要な取組、幅広いユーザーの利用に資する高度化研究・利用支援、計算科学の先導的研究開発等を実施。

■ 政策対応
- 政策的に重要又は緊急と認められる課題（例：感染症対策、気象・防災分野、国が実施する他の研究開発プロジェクトでの利用、計算分野の国際連携に資する利用等）を柔軟に実施。

具体的な利用方法についてはHPCIのHPをご覧ください。
https://www.hpci-office.jp/