PCCC20企業発表資料

FUJITSU
shaping tomorrow with you

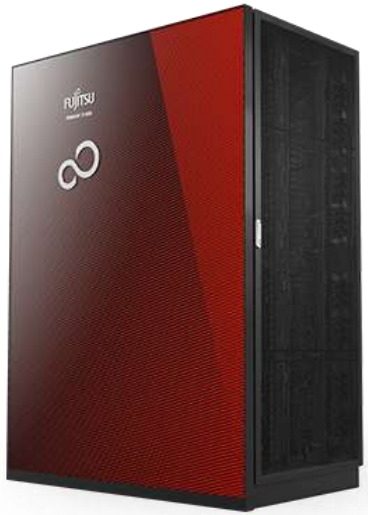
富士通のHPCクラスタへの取り組み
A64FXにおけるAIライブラリ開発状況と MLPerf HPC登録値について

2020/12/14

(株)富士通研究所
中島 耕太

富士通が提供するHPCシステム

- 超ハイエンドからミッドレンジまで幅広い用途に対応
- 代表的なハイエンドシステム
  - 理研「富岳」・「京」
  - 産総研「ABCI」
    - PRIMERGY CX (NVIDIA V100)
  - JAXA「SORA-TOKI」
    - FX1000 (Fujitsu A64FX)
  - JCAHPC「Oakforest-PACS」
    - PRIMERG CX (Intel Xeon Phi)



FX1000

FX700

PRIMERGY GX

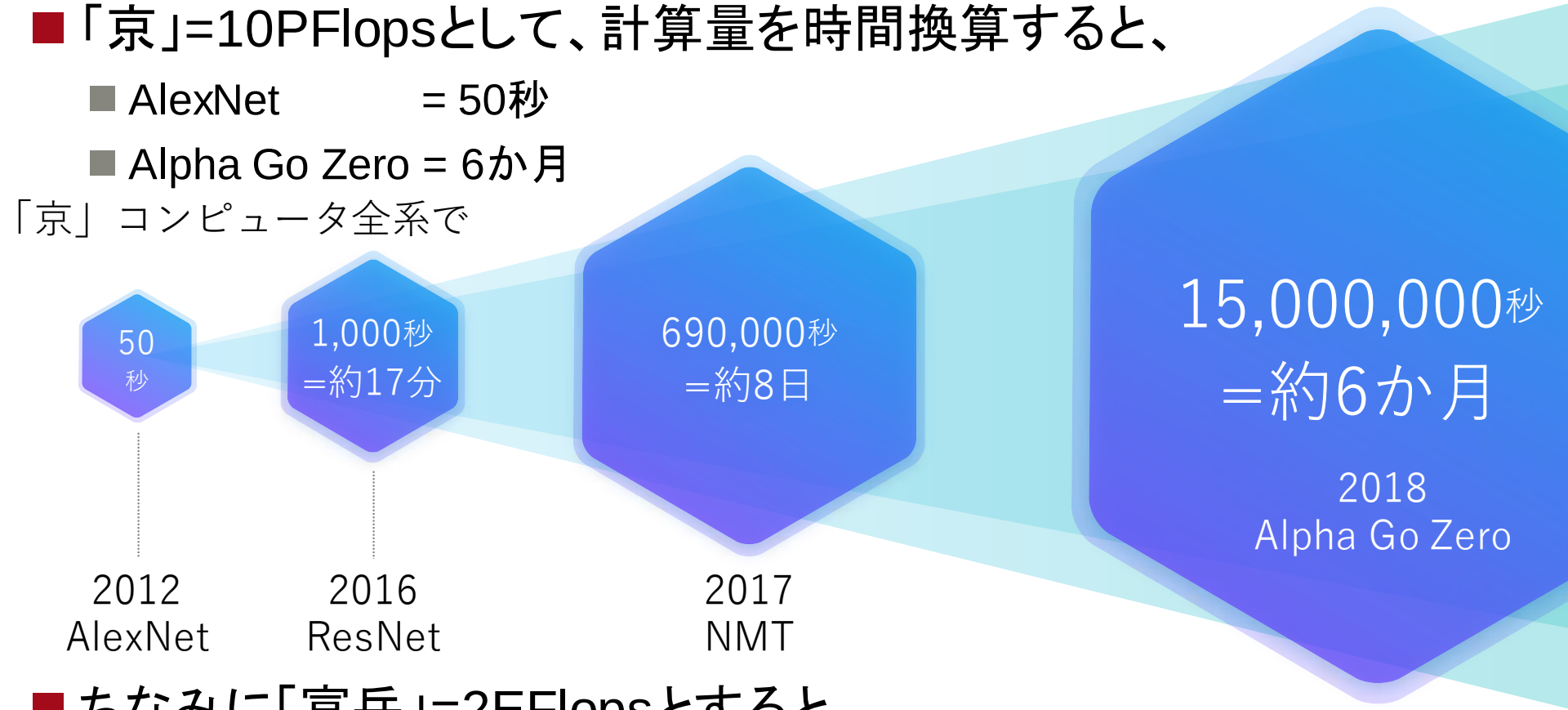
PRIMERGY CX

HPC/AIの様々な用途に合わせた素材を組み合わせてシステムを提供

深層学習が求める計算需要

FUJITSU

- 深層学習が求める計算量: 5年で30万倍
- 「京」=10PFlopsとして、計算量を時間換算すると、
  - AlexNet = 50秒
  - Alpha Go Zero = 6か月



- ちなみに「富岳」=2EFlopsとすると、
  - Alpha Go Zero = 20時間

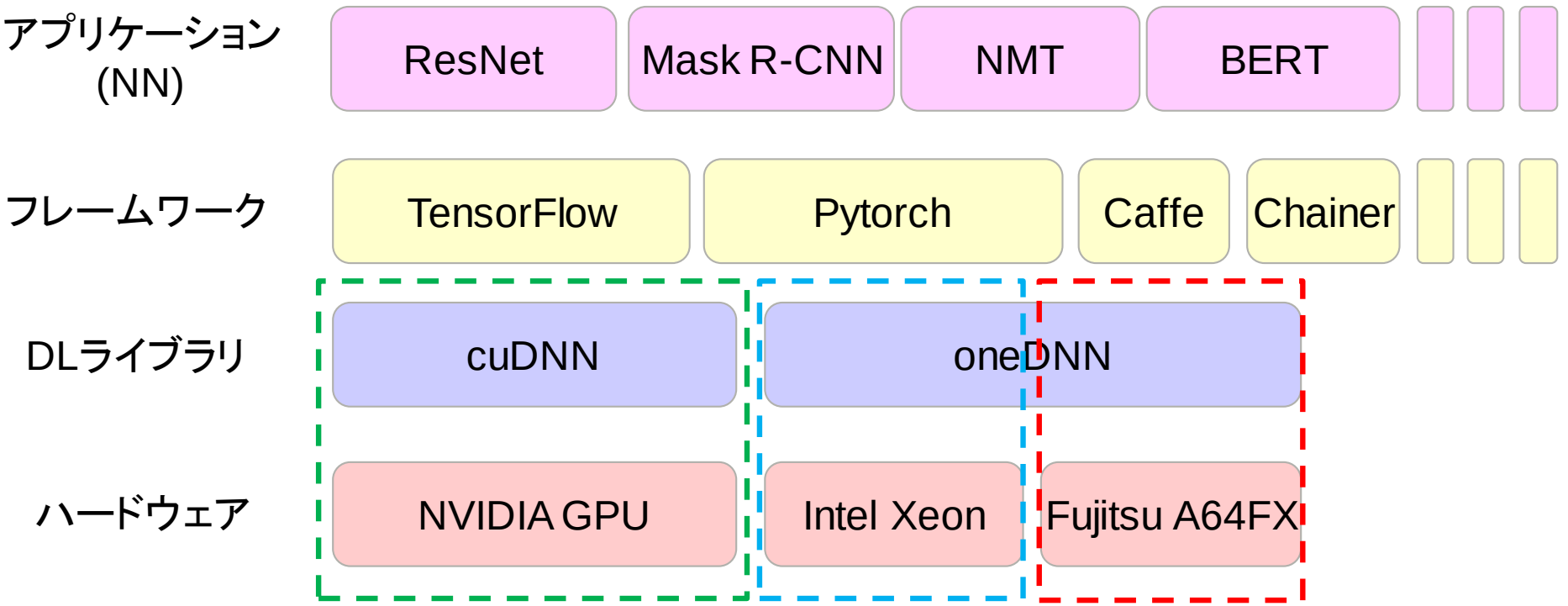
※ あくまで計算量を換算したもの。実際に並列計算できるかというと別。

深層学習が求める計算需要: スーパーコンピュータ級の計算量

深層学習のソフトウェアスタック

FUJITSU

- A64FXでの深層学習を実現するためにDLライブラリを開発
- Intelが開発を主導するOSS「oneDNN」の一部として実装
  - oneDNNがサポートするプラットフォームの一つとしてARM向けコードを実装
  - 本家にもすでに一部コードはマージ

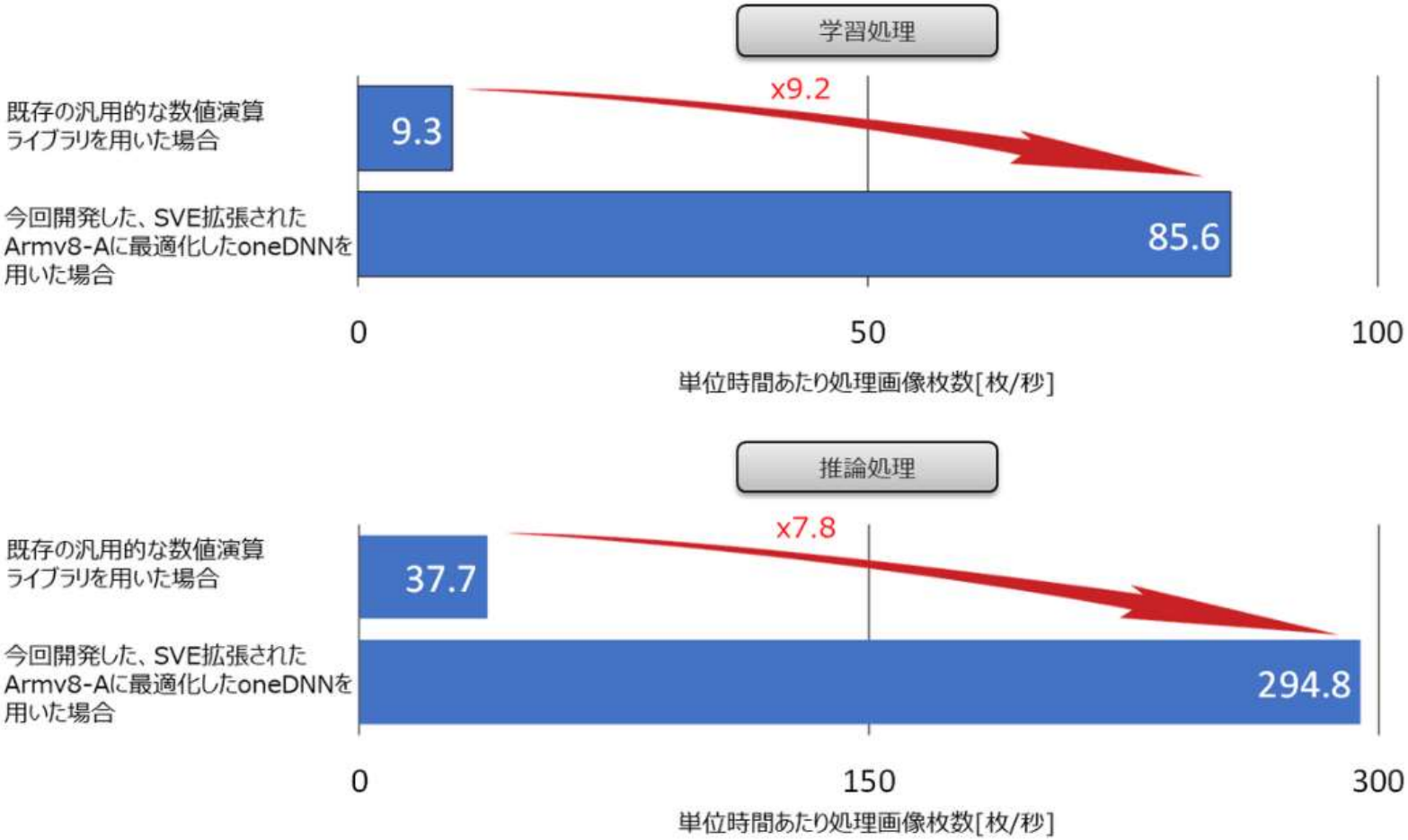


TensorFlowやPyTorchで利用可能に

oneDNNによる深層学習性能

FUJITSU

- SVE命令の適切な活用でA64FXが持つ性能を十分に引き出だす



ニューラルネットワーク: ResNet-50, フレームワーク: TensorFlow

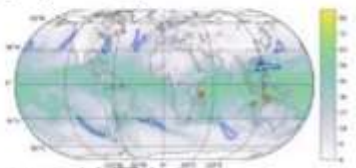
https://blog.fltech.dev/entry/2020/11/18/fugaku-onednn-deep-dive-ja

アプリケーションレベルでの性能評価

FUJITSU

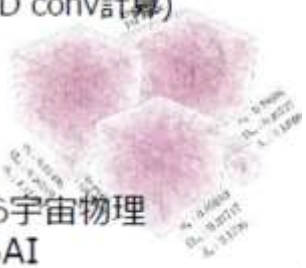
- MLPerf HPCベンチマーク
  - 深層学習のベンチマーク MLPerf trainingの一種
  - 2020年11月に最初のバージョンv0.7の結果を公開
- 題材となる学習処理にかかった時間を計測
  - MLPerf HPC v0.7では「CosmoFlow」と「DeepCam」が題材
  - 問題サイズはあらかじめ決められている
  - ある精度に達するまでの学習にかかった時間を計測

Deep CAM (441層)



気候シミュレーション結果から
異常気象を見つけるAI

CosmoFlow (3D conv計算)



暗黒物質の分布から宇宙物理
パラメータを推測するAI

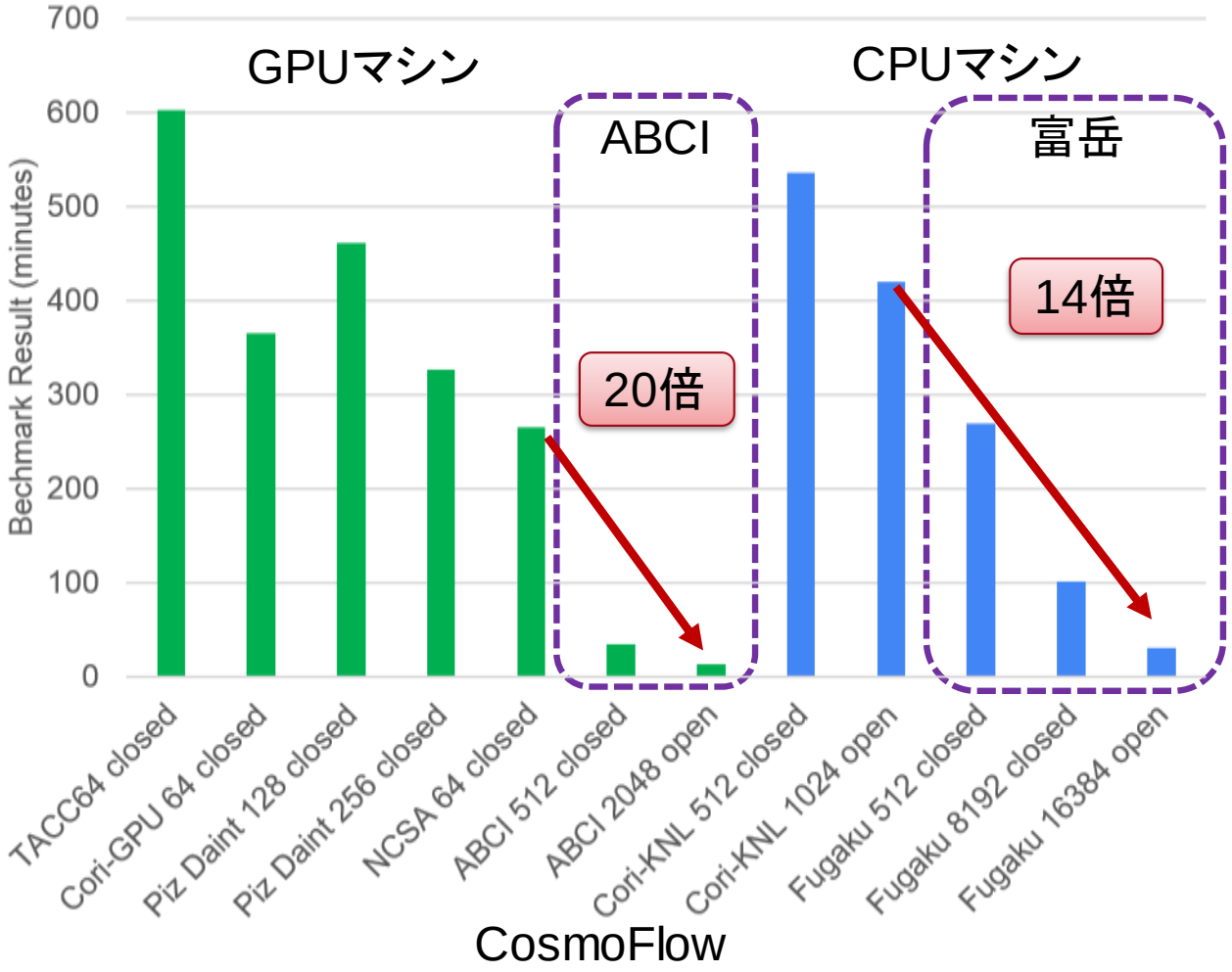
- 登録機関
  - 欧米からは、TACC、NERSC-Cori、NCSA、Piz Daintが登録
  - 日本からは、富岳(全系の1/10)、ABCI(全系の1/2)が登録

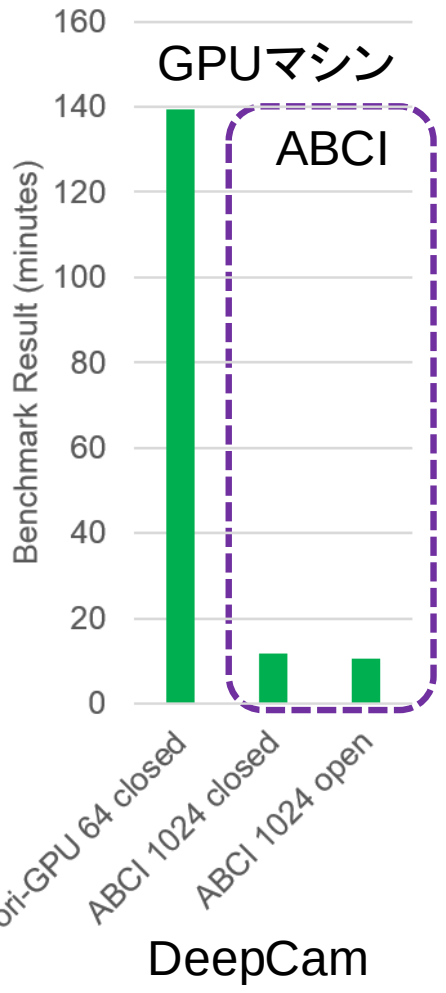
※富岳の計測は、理研・富士通の共同研究による
※ABCIの計測は、産総研・富士通研の共同研究によるもので、グランドチャレンジを活用

MLPerf HPC v0.7結果

FUJITSU

■ 他の機関と比較し、ABCI・富岳の結果は非常に高速





圧倒的な高速性を実現

何がすごいの？

FUJITSU

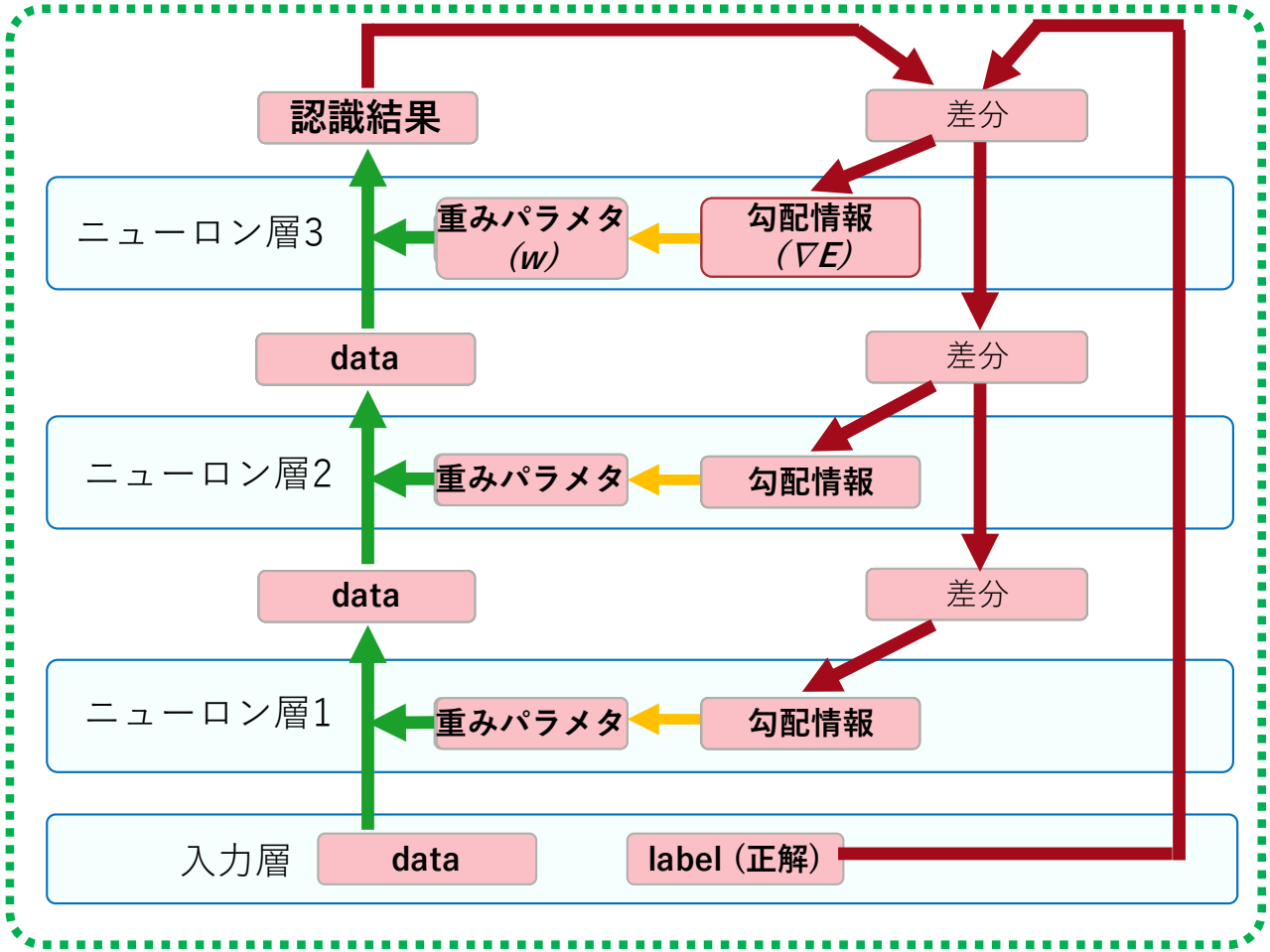
- 大規模なら高速化できるのは当たり前？
  - No。大規模化そのものが難しい
  - 並列数に事実上の上限がある

大規模化のチューニング力が求められる

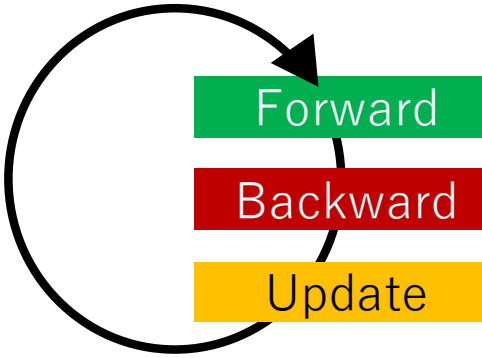
- 技術的チャレンジ
  - データ並列の大規模化
  - モデル並列への挑戦

深層学習のサイクル

- Forward/Backward計算で勾配情報を求める
- 何回かのForward/Backwardをまとめてupdateする



学習処理サイクル



大規模AI学習の課題(1): データ並列

FUJITSU

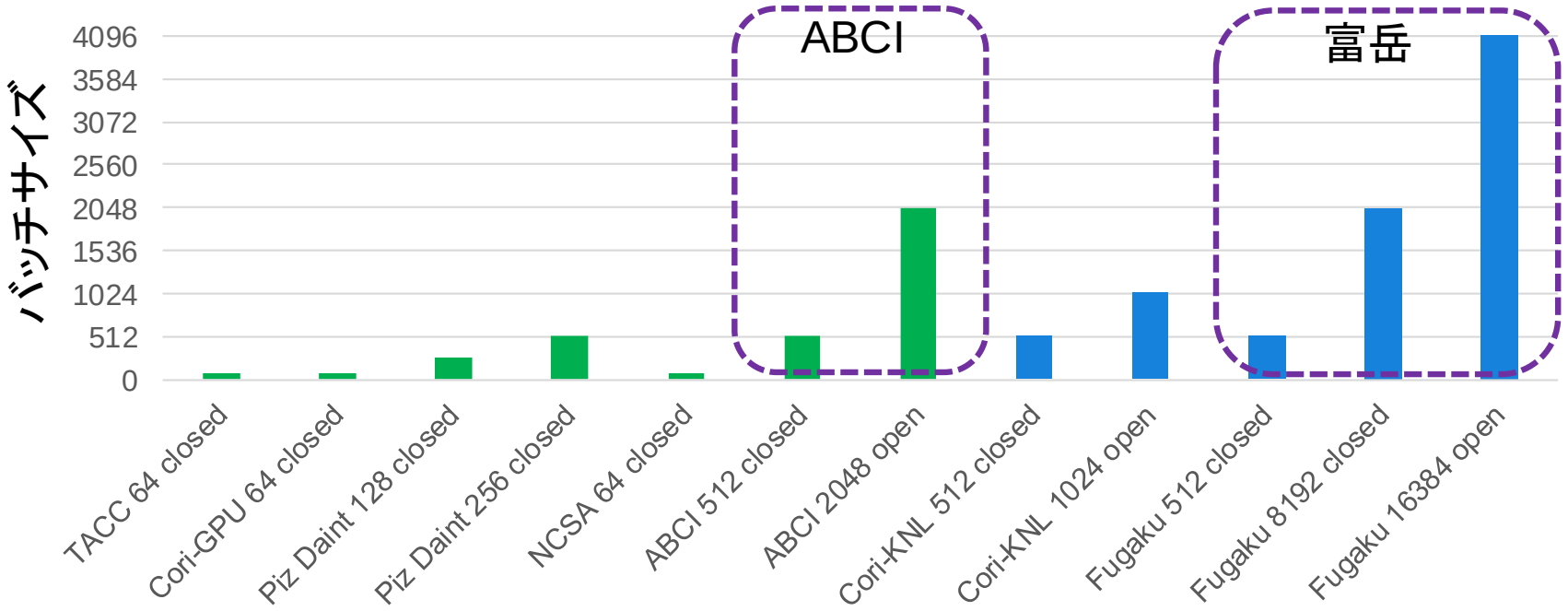
- データ並列では、勾配情報の集約を行ってアップデートを実施
- バッチサイズ(一括アップデートの単位)が大きすぎると精度低下

バッチサイズの上限が並列数上限を決める

ML Perf HPCにおけるバッチサイズ

- バッチサイズ増大に伴い学習精度が低下
  - 並列数を増やすにはバッチサイズを増やす必要がある
  - バッチサイズを増やすとEpoch数が増加 → 実行時間増加

Epoch数増加を抑制しつつバッチサイズを増やす
ハイパーパラメータチューニングを追求

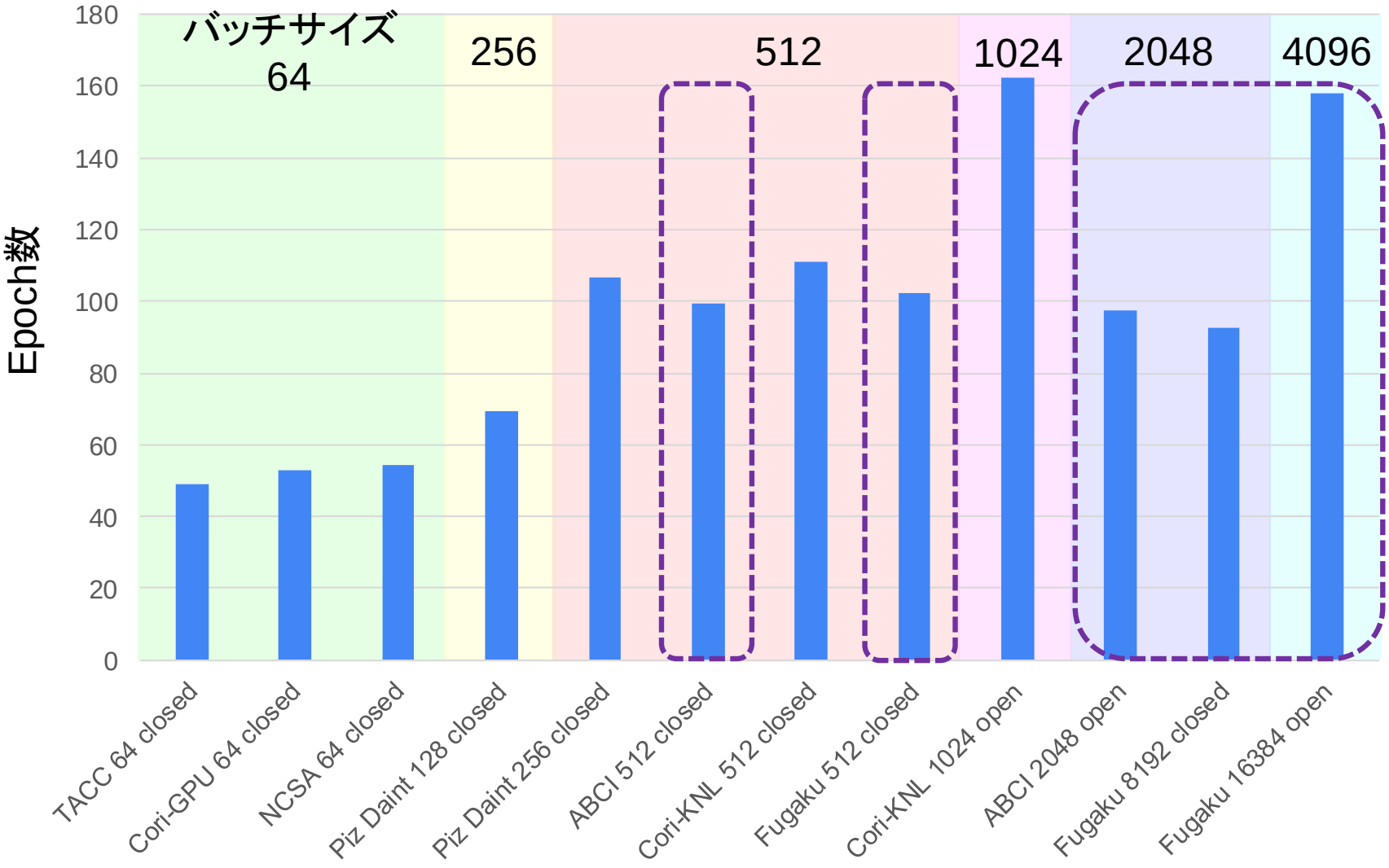


ABCI・富岳では他機関の4倍のバッチサイズを実現

バッチサイズとEpoch数

FUJITSU

- ABCI・富岳では、バッチサイズ2k/4kでのEpoch数増加を低減

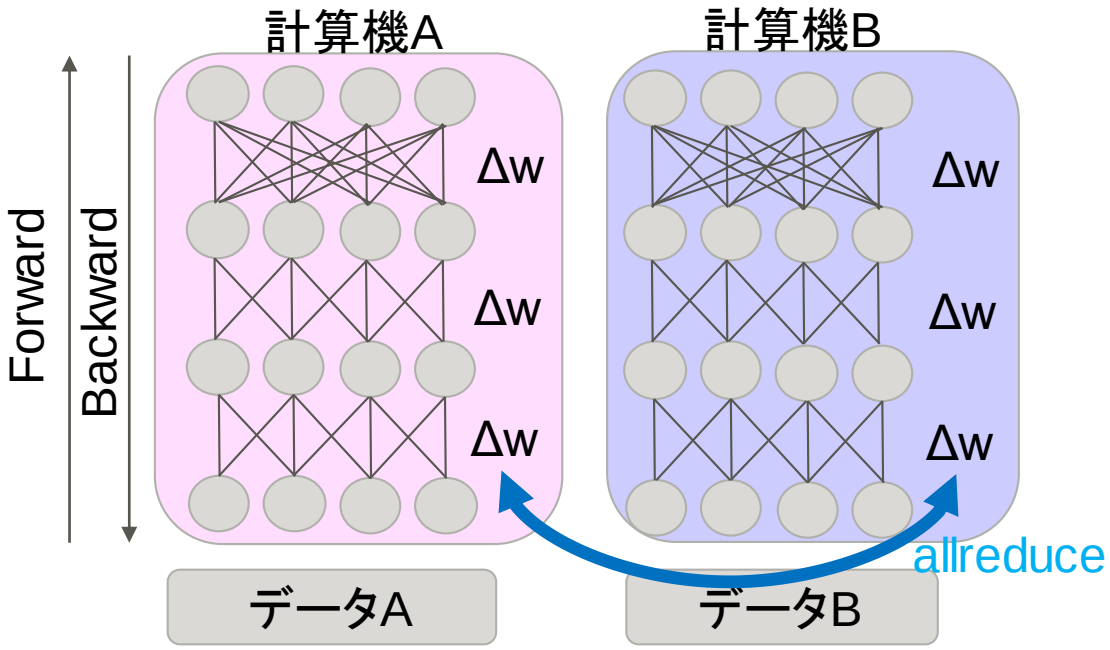


大規模AI学習の課題(2): モデル並列

FUJITSU

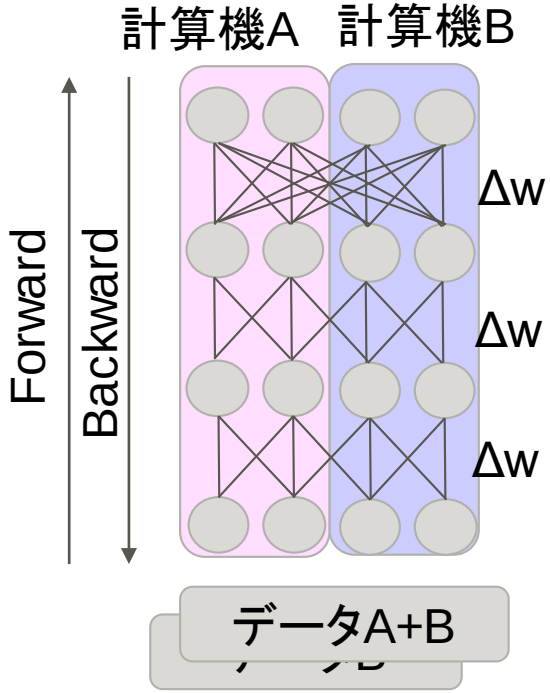
■ データ並列

- 勾配情報のみを集約
- 通信コストが低く、性能はスケール



■ モデル並列

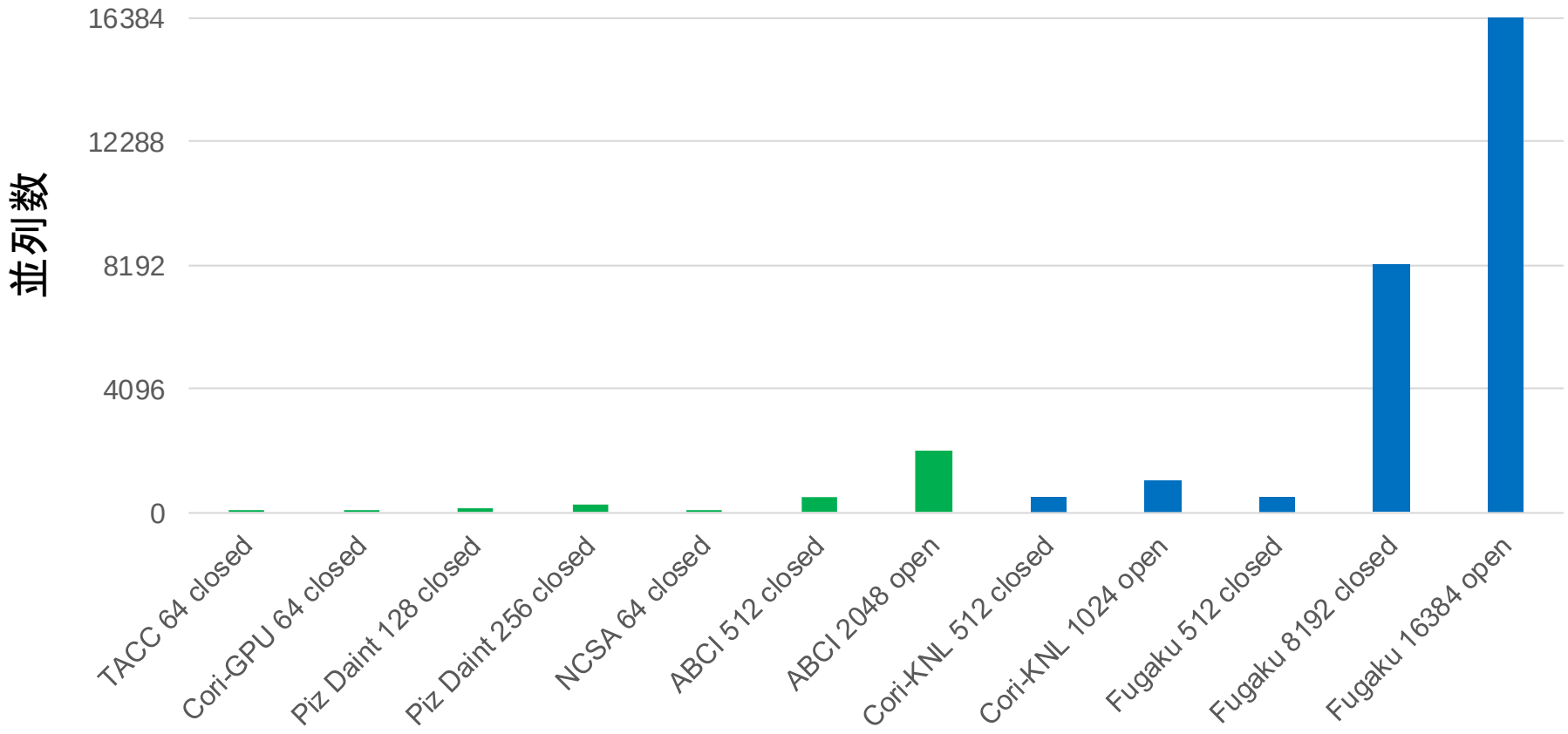
- NN計算そのものを並列化
- 袖通信コストが高く、高速化困難



モデル並列の適用で1.7倍の高速化に成功(4並列時)

モデル並列による計測

- 富岳の計測のみモデル並列を利用
- データ+モデル並列により、最大16,384並列での計測を実現



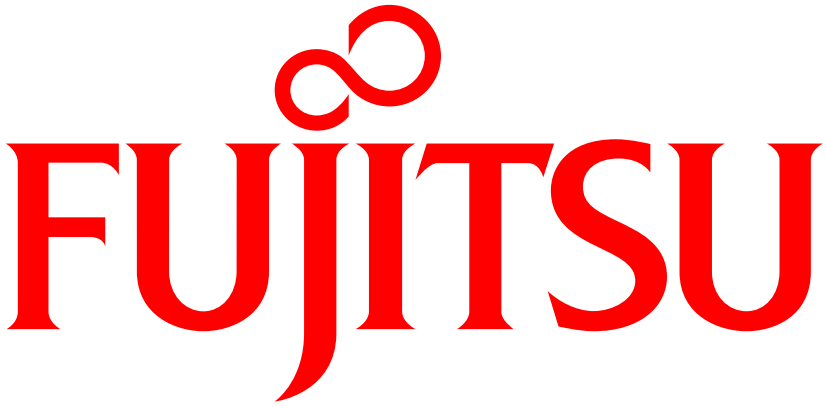
大規模環境での総合力で高速性を実現

おわりに

FUJITSU

- AIが求める計算需要にこたえるために
  - HPC技術によるAI学習高速化技術を開発
  - A64FX向けAIライブラリをoneDNNに実装、Pytorch/TensorFlowで利用可能
- MLPerf HPCによる大規模AI計算
  - MLPerf HPC v0.7 の特徴
    - 決められた問題サイズを解く時間を計測
    - バッチサイズの上限が並列数の上限を決める
    - 10,000ノードを超えるような大規模システムには向かないベンチマーク
  - 並列性の上界へのチャレンジ
    - データ並列: ハイパーパラメータチューニングによるバッチサイズ最大化
    - モデル並列: 通信オーバヘッドの削減
    - 16,384並列実行による高速化に成功
- 今後の課題
  - 大規模環境のメリットを活かした新しいDLアルゴリズムの開発

人類が解けていない未踏の課題にチャレンジ



shaping tomorrow with you