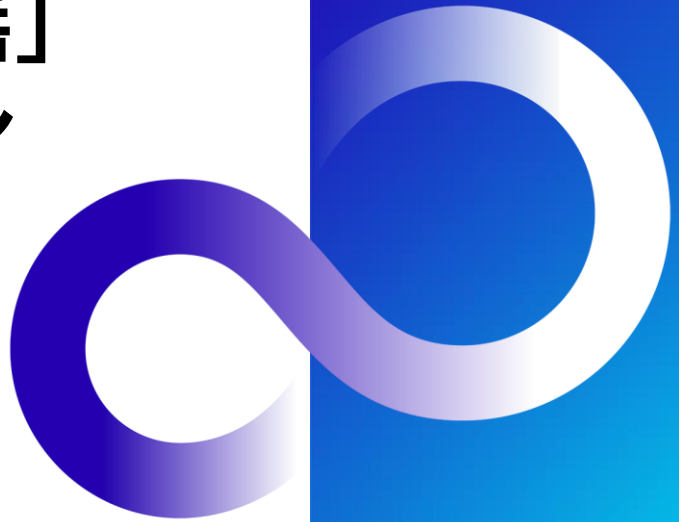
スーパーコンピュータ「富岳」で学習した大規模言語モデル Fugaku-LLM



2024年8月6日

富士通株式会社　人工知能研究所

シニアプロジェクトディレクター

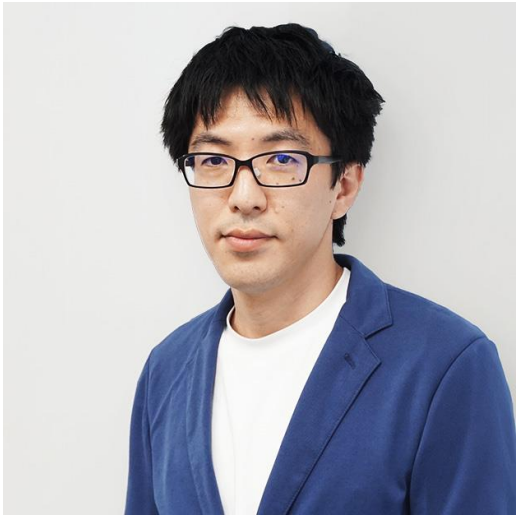
白幡　晃一

自己紹介

FUJITSU

白幡 晃一 (しらはた こういち)

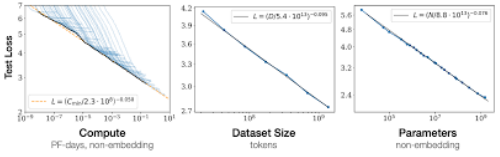
富士通株式会社　人工知能研究所
シニアプロジェクトディレクター



- 2015年3月　東京工業大学 大学院情報理工学研究科 博士後期課程修了 博士（理学）
- 2015年4月　株式会社富士通研究所 入社
- 2020年11月　スーパーコンピュータ「富岳」およびABCIを用いて機械学習処理性能ベンチマークMLPerf HPCで世界最高性能を達成
- 2023年5月　スーパーコンピュータ「富岳」政策対応枠で大規模言語モデル分散並列学習手法の開発

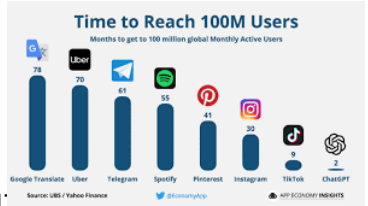
ここ数年の生成AIの歩み

2019/06/22 MicrosoftがOpenAIに10億ドル(当時 約1100億円)を投資
2020/01/23 OpenAIが言語生成モデルに関するScaling Lawの論文を発表 →



2021/01/05 OpenAIが画像+言語モデルCLIPと画像生成モデルDALL-Eを発表
2022/05/23 Googleが画像生成モデルImagenを発表 →

2022/07/22 BigScience (HuggingFace,CNRS,GENCI)が多言語モデルBloomを発表
2022/08/04 精華大学がGLM-130Bを発表
2022/08/22 Stability AIが画像生成モデルStable Diffusionを発表
2022/11/30 OpenAIがChatGPTを発表
2023/02/02 公開後約２ヶ月で1億ユーザを突破 →



2023/02/03 自民党AIの進化と実装に関するプロジェクトチーム (第１回)
2023/03/14 ChatGPT PlusでGPT-4が利用可能に
2023/03/16 Microsoft 365 CopilotでWord,Outlook,TeamsなどからAIアシスタントが利用可能に
2023/05/24 「富岳」政策対応課題の開始

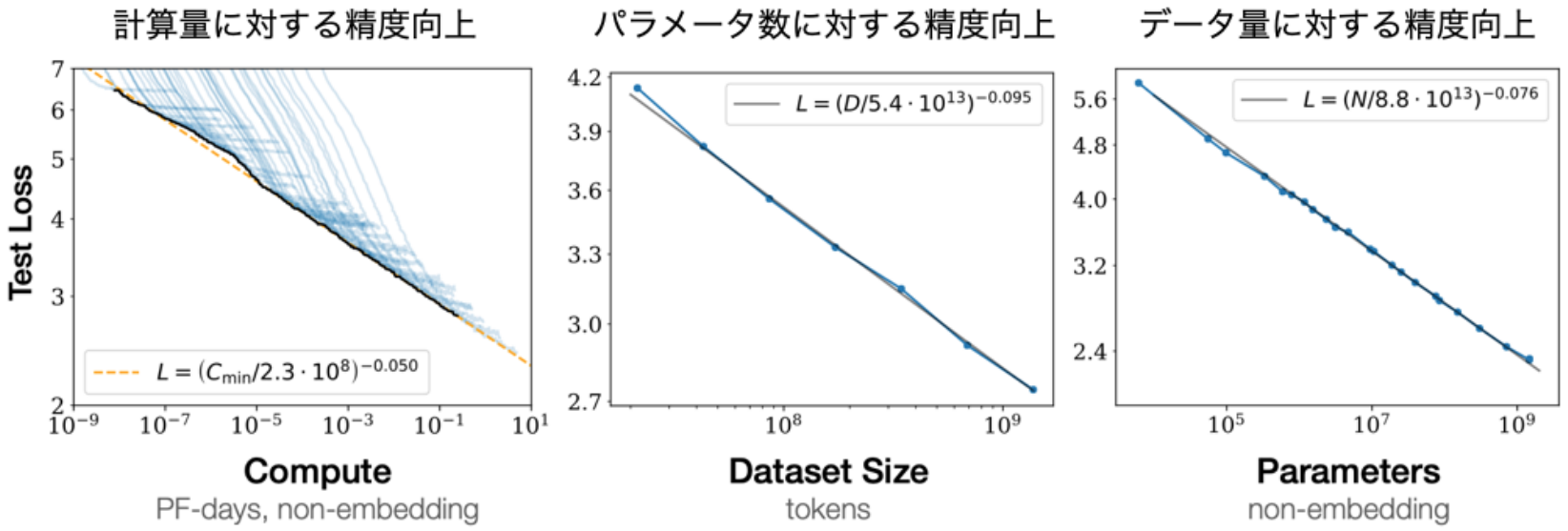
2023/07/18 MetaがLLaMA2を公開 →
2023/10/03 産総研の生成AI開発支援プログラムに国立情報学研究所(NII)とELYZA社が採択
2023/12/19 東工大・産総研が Swallow-7B,13B,70Bを公開
2024/02/15 GoogleがGemini Pro 1.5を発表 →

2024/03/04 AnthropicがClaude3を発表
2024/04/18 MetaがLLaMA3を公開
2024/05/10 Fugaku-LLM公開

予測可能な費用対効果(スケール則)

FUJITSU

- 計算量 ∝ パラメータ数 x データ量
- 計算量 ∝ GPU数 x 計算時間
- データやモデルの質を改善をすればもっと安くなる
- 工夫をしなくても最低でもこの費用対効果の下限は保証される



出典：Scaling Laws for Neural Language Models

スーパーコンピュータ「富岳」

RIKEN

FUJITSU

富岳

Fugaku

4期連続世界4冠達成

HPCG、Graph500において9期連続の世界第1位

機械学習処理ベンチマークMLPerf HPCでも世界１位を獲得

世界１位
- HPCG
- Graph500
- TOP500 ('20.06-'21.11)
- HPL-AI ('20.06-'21.11)

高性能 & 高効率
- A64FX (ARM)
- 5 PB memory
- 158,976 nodes
- 442 ペタ フロップス*
  (ベンチマーク性能)
- 30M W

*1ペタ フロップス (FLOPS) = 1秒間に10^{15}(1000兆)回の浮動小数点演算

スーパーコンピュータ「富岳」は
理化学研究所様と富士通が共同開発しました

photo credit: RIKEN

Fugaku-LLMの研究チーム

Rio Yokota

Collaborators

GPT-Fugaku Team


Noriyuki Kojima


Kazuto Ando


Koji Nishiguchi


Jungo Kasai


Keisuke Sakaguchi


Shukai Nakamura

DL4Fugaku Team @ R-CCS


Aleksandr Drozd


Mohamed Wahib


Kento Sato


Jens Domke


Emil Vatai

DL4Fugaku Team @ LLNL


Nikoli Dryden


Tal Ben Nun

Fujitsu


Kenichi Kobayashi


Tatsuya Hiraoka


Naoto Fukumoto


Akihiko Kasagi


Koichi Shirahata


Kentaro Kawakami


Masafumi Yamazaki


Hiroki Tokura


Takumi Honda


Tsuguchika Tabaru

各機関と企業の役割

FUJITSU

東京工業大学：全体総括、大規模言語モデルの並列化および通信の高速化（3種類の並列化を組み合わせた通信性能の最適化、TofuインターコネクトD上での集団通信の高速化）

東北大学：学習用データの収集、学習モデルの選択

富士通：演算高速化と通信の高速化（TofuインターコネクトD上での集団通信の高速化、パイプライン並列の性能最適化）、事前学習と学習後のファインチューニング

理化学研究所：大規模言語モデルの分散並列化および通信の高速化（TofuインターコネクトD上での集団通信の高速化）

名古屋大学：3D形状生成AIへのFugaku-LLMの応用方法の検討

サイバーエージェント：学習用データの提供

Kotoba Technologies：深層学習フレームワークの「富岳」への移植

「富岳」でGPTを学習するには？

FUJITSU

- 高性能計算分野の課題
  - 深層学習フレームワークMegatron-DeepSpeedの「富岳」への移植
  - 小規模な行列の積をバッチ処理する部分の高速化
  - TofuDネットワーク上での集団通信の高速化
  - FP16を用いても安定に学習できる方法の開発
- 自然言語処理分野の課題
  - 言語データの収集・クリーニング
  - 弁護士による法務確認
    - 研究成果（ソースコード・モデル・データ）の公開における著作権や契約等の制限を確認し、合法的な公開体制を構築
  - 事後学習・微調整の手法の検討

研究成果①：スーパーコンピュータ「富岳」における大規模言語モデルにおける学習の計算性能を大幅に向上

・深層学習フレームワークMegatron-DeepSpeedを「富岳」へ移植し、CPU上の行列演算ライブラリを高速化
→ 高速化前と比べて６倍の演算速度を実現　（110秒かかっていたものが18秒に）

1693389241.318550480.fcc.pytorch.y.r1.13_for_a64fx.tar のプロファイル結果

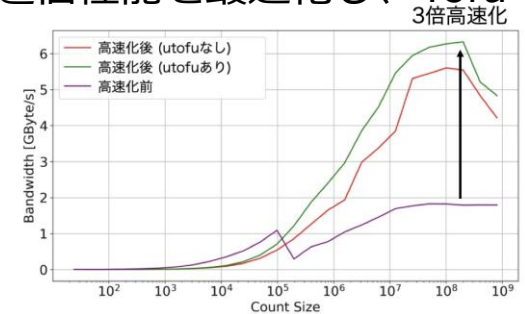
Name	Self CPU %	Self CPU	CPU total %	CPU total	CPU time avg	# of Calls
aten::bmm	18.07%	110.819s	18.08%	110.845s	24.055ms	4608
aten::bmm	18.17%	108.802s	18.17%	108.832s	23.618ms	4608
aten::bmm	18.53%	110.858s	18.53%	110.890s	24.065ms	4608
aten::bmm	19.15%	110.594s	19.16%	110.625s	24.007ms	4608
aten::bmm	18.33%	108.646s	18.34%	108.679s	23.585ms	4608

→

1701935794.711074240.fcc.pytorch.y.r1.13_for_a64fx.tar.gz のプロファイル結果

Name	Self CPU %	Self CPU	CPU total %	CPU total	CPU time avg	# of Calls
aten::bmm	3.56%	18.273s	3.57%	18.302s	3.972ms	4608
aten::bmm	3.64%	18.394s	3.64%	18.423s	3.998ms	4608
aten::bmm	3.57%	18.154s	3.57%	18.185s	3.946ms	4608
aten::bmm	3.58%	17.959s	3.59%	17.990s	3.904ms	4608
aten::bmm	3.61%	18.341s	3.62%	18.373s	3.987ms	4608

・「富岳」向けに3種類の並列化を組み合わせて通信性能を最適化し、TofuインターコネクトD上での集団通信を高速化
→ 高速化前と比べて３倍の通信速度を実現



・大規模言語モデルの学習には通常GPUが用いられており、世界中のGPU不足が社会問題となっているが、今回、「富岳」に搭載されている富士通製の国産CPUで大規模言語モデルを学習できることを示したことは、経済安全保障の観点からも重要な成果

「富岳」でのTransformer最適化

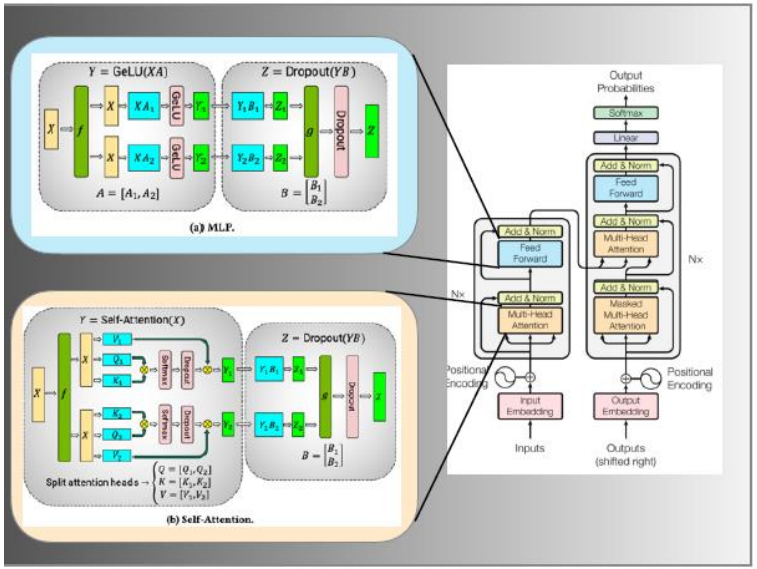
FUJITSU

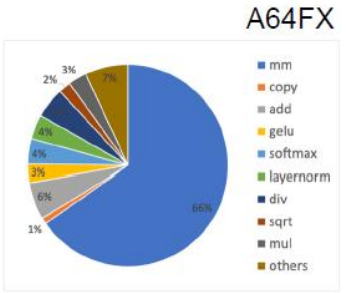
- Transformerの性能を「富岳」上で最適化するため、ソフトウェアスタックの各レイヤーの性能分析と最適化
  - 特に、**密行列積の高速化**と、**通信性能の最適化**

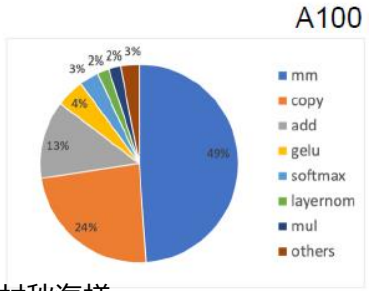
Transformer (GPT-x)	**Transformerの性能測定、ボトルネックの解析**
並列化 (Megatron-DeepSpeed)	**3種類の並列化を組み合わせた「富岳」向け通信性能最適化**
深層学習フレームワーク (PyTorch)	**富士通が高速化した「富岳」向けフレームワークを使用。LLM向け高速化**
数学ライブラリ	**密行列積ライブラリのTransformer向け高速化**

GPTの演算時間の内訳

- 計算の多くは密行列-密行列の積
  - → A64FXでは66%、A100では49%の時間がこの部分に費やされていた
  - → 理論ピークの1/3の性能になっており大幅に向上できる可能性があった







ある共通の入力 x に対しそれぞれの変換行列を適用して
$Q=XW_Q$, ,$K=XW_K$, $V=XW_V$ を用意する。
自分自身の要素との注目度合いを抽出する。

2022年度第2回計算科学フォーラム
https://hpcic-kkf.com/forum/2022/kkf_02/data/yokota_kkf2022-02_v2.pdf

提供：東工大 横田研 中村秋海様

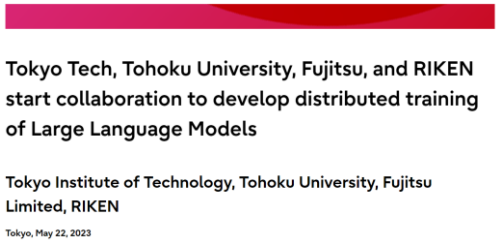
Implementation of Batch Matrix Multiplication

FUJITSU

Implementation of Batch Matrix Multiplication for Large Language Model Training on A64FX CPUs, Hiroki Tokura et al., COOL Chips 27

- Large Language Models (LLMs) represent a significant leap
- LLM training is also carried out on the supercomputer Fugaku.
- LLM process is dominated by matrix multiplications
  - 2 types of matrix multiplication
    - A large size matrix multiplication
    - Batch Matrix Multiplication (BMM): performing multiple matrix multiplications

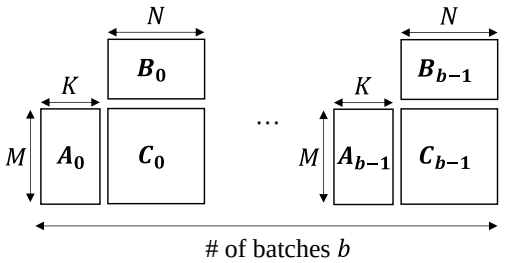
We propose **an efficient BMM implementation for A64FX CPUs.**



Tokyo Tech, Tohoku University, Fujitsu, and RIKEN start collaboration to develop distributed training of Large Language Models

Tokyo Institute of Technology, Tohoku University, Fujitsu Limited, RIKEN

Tokyo, May 22, 2023

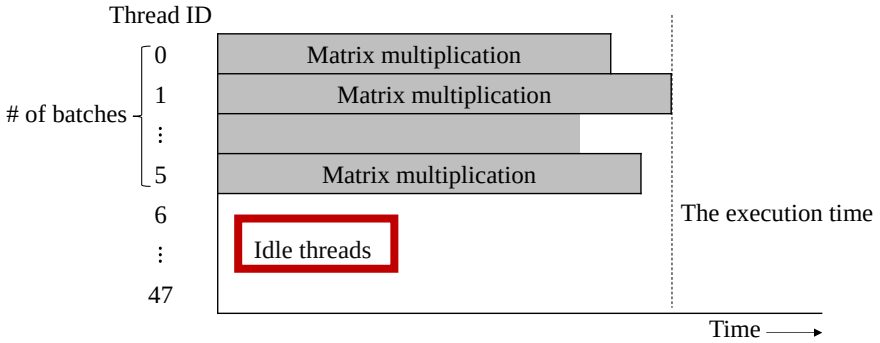


PyTorch original implementation

FUJITSU

Implementation of Batch Matrix Multiplication for Large Language Model Training on A64FX CPUs, Hiroki Tokura et al., COOL Chips 27

- PyTorch uses BLAS routines to accelerate matrix multiplications in LLMs
- A matrix multiplication is assigned per thread
  - The performance is decreased if the number of matrix multiplications is less than the number of cores

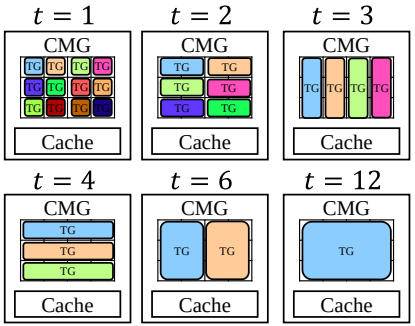


Batch matrix multiplication patterns appeared in the LLM training of our evaluation.

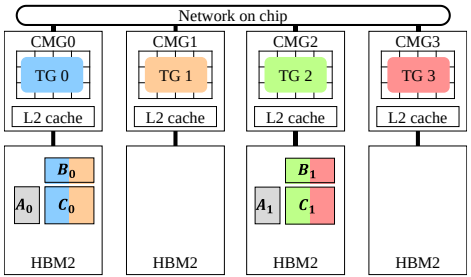
Pattern	1	2	3	4	5
Transpose of A	No	No	No	Yes	Yes
Transpose of B	No	Yes	Yes	No	No
# of matrix multiplications	6	6	6	6	6
M	144	144	2048	2048	2048
N	2048	2048	144	2048	2048
K	2048	2048	2048	144	144
LDA	2592	864	2048	2592	2592
LDB	2048	2048	2592	2592	864
LDC	144	144	2048	2048	2048

Our proposed implementation

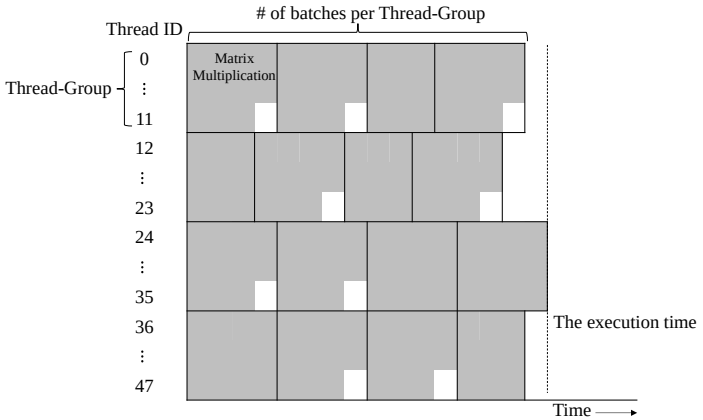
- A64FX CPU has 48 cores
  - 48 Threads are divided into Thread-Groups(TGs) every t threads
- A matrix multiplication is computed by g TGs
  - We experimentally determine the optimal values of t and g



The patterns of the Thread-Groups.



Example of TG assignment for $g = 2$

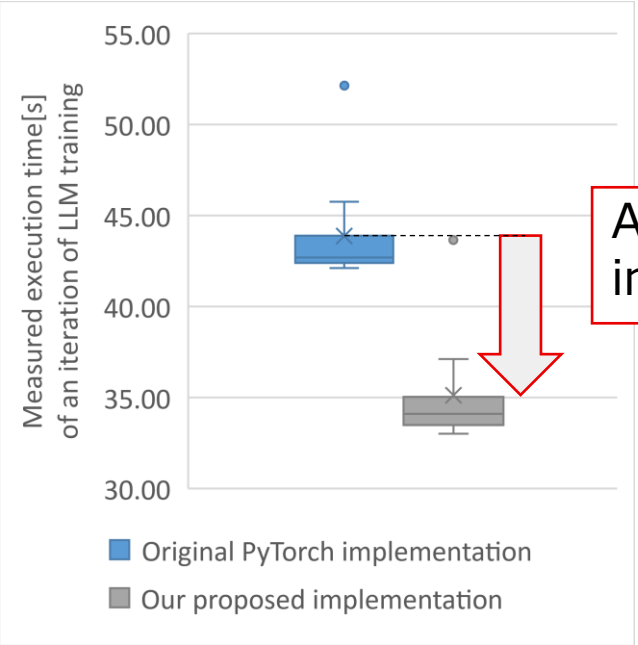


Evaluation result of overall LLM training

FUJITSU

Implementation of Batch Matrix Multiplication for Large Language Model Training on A64FX CPUs, Hiroki Tokura et al., COOL Chips 27

Our proposed implementation contributes to a 25% improvement in overall LLM training.



A 25% improvement

CPU: A64FX(48cores, 2.2GHz)
The language environment: tcsds-1.2.38

Fujitsu Processor A64FX Specifications

Cores	48
Frequency[GHz]	2.2
FP32 Peak	6.75
Flops[TFLOPS]	84

The optimal values of t and g of each pattern.

Pattern	1	2	3	4	5
t: the number of threads in a TG	12	4	4	4	4
g: the number of TGs for a matrix multiplication	2	2	2	2	2

GPTの並列学習方法

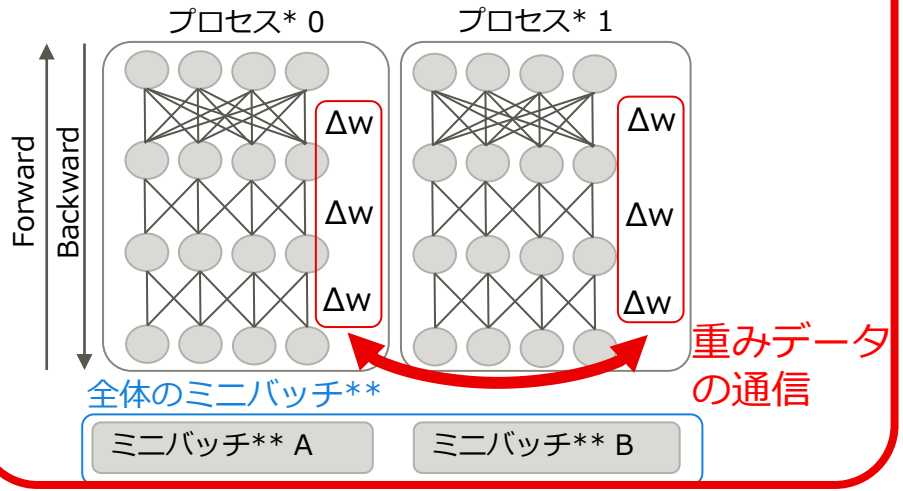
FUJITSU

○ 「富岳」でGPTを高効率に学習するには3種類の並列化を適切に組み合わせることが重要

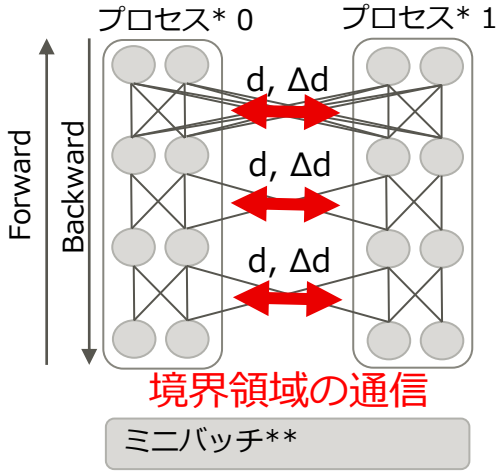
データ並列とテンソル並列

基本はデータ並列を使用

- データ並列:複数のデータを並列に計算
  - **長所: 通信時間の影響を受けにくい**
  - **短所: バッチサイズを上げすぎると学習が進まなくなる**



- テンソル並列: NN を分割して並列に計算
  - **長所: 学習精度の低下が起きない**
  - **短所: 通信時間による影響を受けやすい**

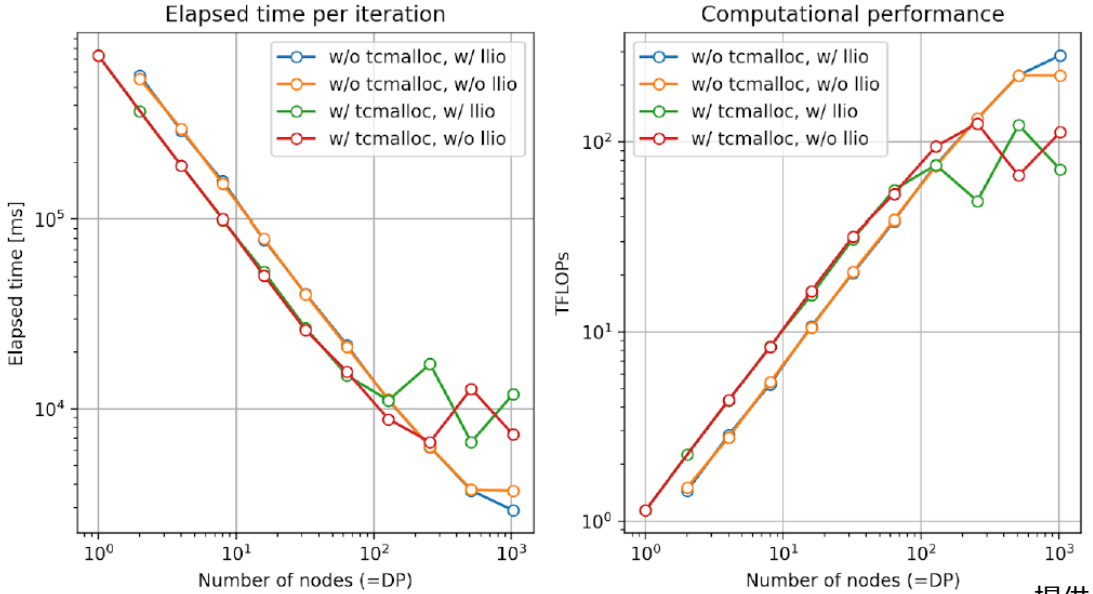


*プロセス: 1つのプロセッサに割り当てられる処理の単位 (1つのプロセッサに複数のプロセスを割り当てることも可能)
**ミニバッチ: 一度に処理させる入力データ(画像など)のサンプル数

データ並列のスケーラビリティ

FUJITSU

sequence-length=1024
per-cpu-batchsize=1, global-batch-size=1024
gradient-accumulation-steps=1024/#nodes
#parameters=**124M**



DP: データ並列のノード数

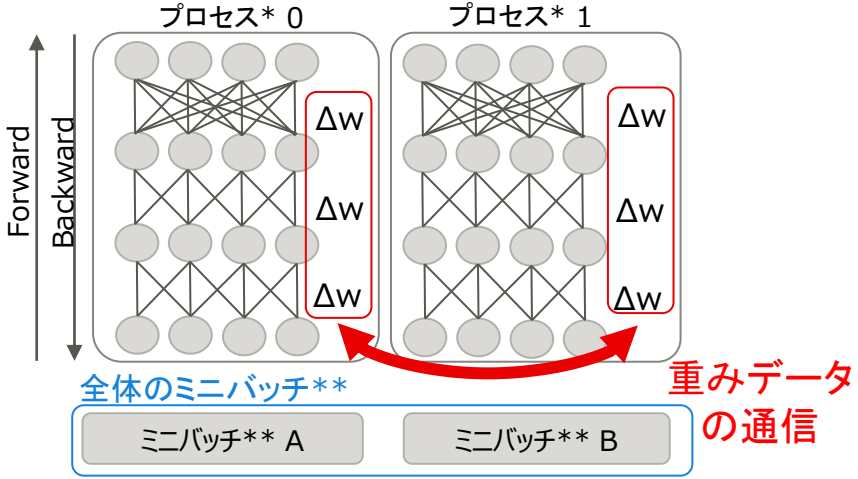
提供：理研 安藤 和人様

2022年度第2回計算科学フォーラム
https://hpcic-kkf.com/forum/2022/kkf_02/data/yokota_kkf2022-02_v2.pdf

データ並列とテンソル並列

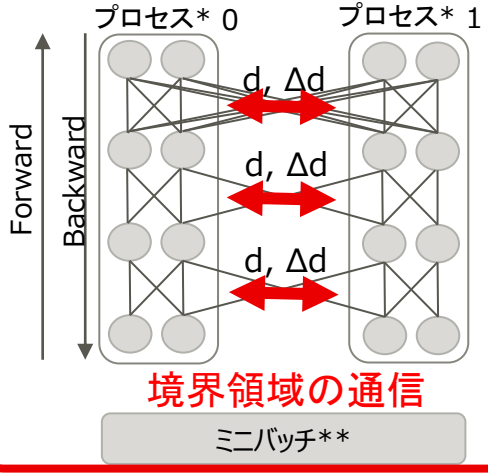
FUJITSU

- データ並列:複数のデータを並列に計算
  - **長所: 通信時間の影響を受けにくい**
  - **短所:バッチサイズを上げすぎると学習が進まなくなる**



テンソル並列も組み合わせて使う

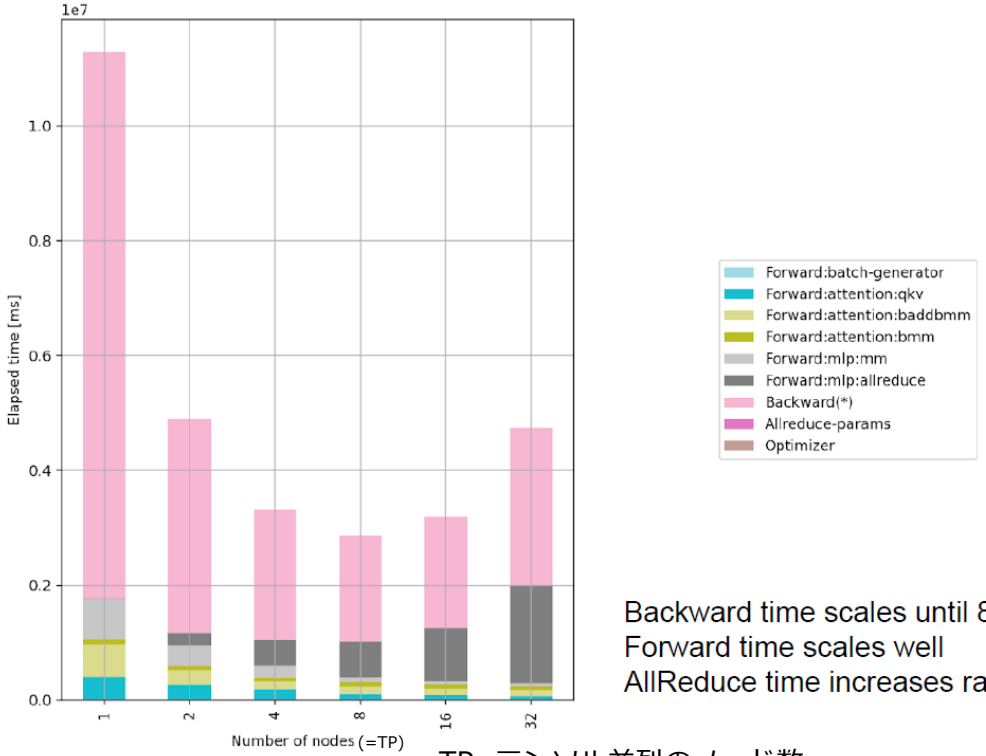
- テンソル並列: NN を分割して並列に計算
  - **長所: 学習精度の低下が起きない**
  - **短所: 通信時間による影響を受けやすい**



*プロセス: 1つのプロセッサに割り当てられる処理の単位 (1つのプロセッサに複数のプロセスを割り当てることも可能)
**ミニバッチ: 一度に処理させる入力データ(画像など)のサンプル数

テンソル並列のスケーラビリティ（実行時間の内訳）

FUJITSU



Backward time scales until 8 nodes
Forward time scales well
AllReduce time increases rapidly

TP: テンソル並列のノード数

提供：東工大 横田研 中村秋海様

2022年度第2回計算科学フォーラム
https://hpcic-kkf.com/forum/2022/kkf_02/data/yokota_kkf2022-02_v2.pdf

パイプライン並列

FUJITSU

- Megatron-LM の interleaved 1F1B を使用

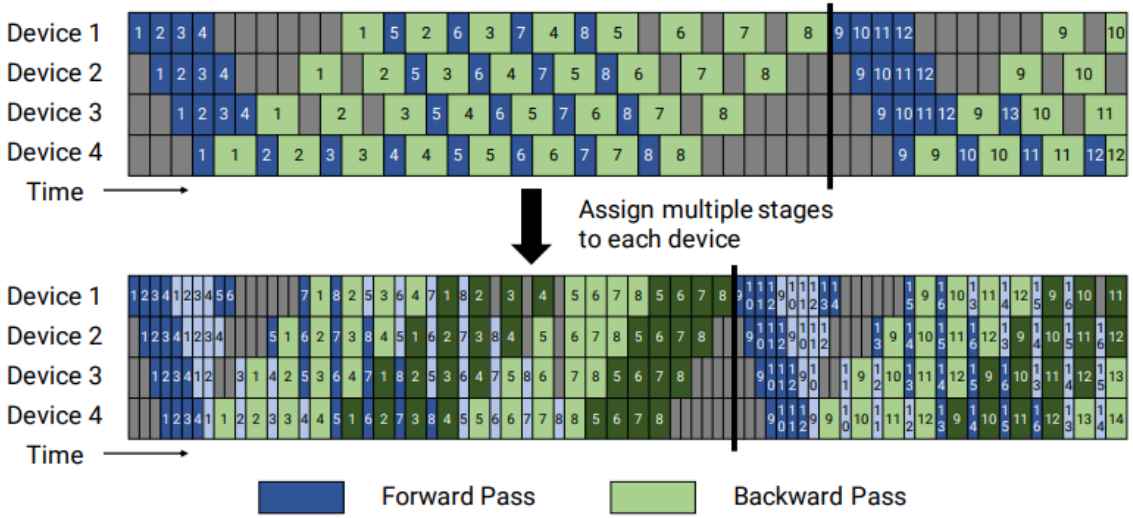


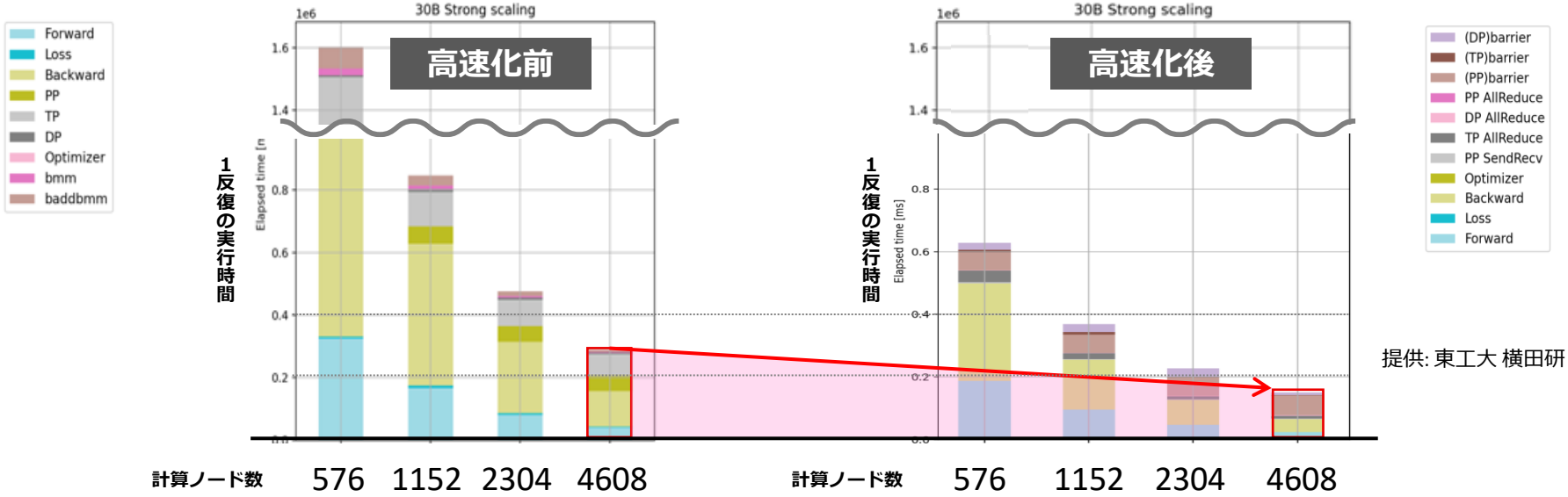
Figure 4: Default and interleaved 1F1B pipeline schedules. The top figure shows the default non-interleaved 1F1B schedule. The bottom figure shows the interleaved 1F1B schedule, where each device is assigned multiple chunks (in this case, 2). Dark colors show the first chunk and light colors show the second chunk. The size of the pipeline bubble is smaller (the pipeline flush happens sooner in the interleaved timeline).

https://arxiv.org/pdf/2104.04473.pdf

30Bパラメータでの学習の計算性能

FUJITSU

○計算効率を開発当初の10%から20%程度まで向上

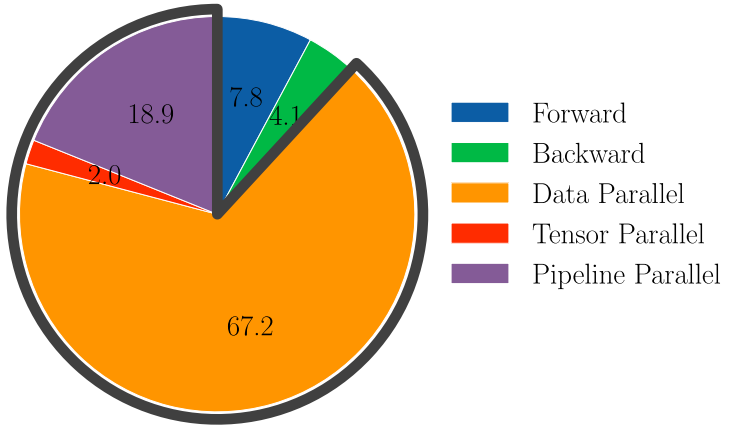


提供: 東工大 横田研

背景：言語モデル学習における時間の割合

FUJITSU

- 約9割が通信関連の時間
  - 通信時間の割合を減らすことが高速化につながる
- Tensor ParallelとData Parallel：Allreduce通信
- Pipeline Parallel：
  Send Recvによる隣接通信
  待機時間(バブル)を含む



富岳上でGPT-13Bの訓練における時間の割合
TP=6, PP=8, DP=64

富岳上の大規模機械学習におけるAll-reduce通信の高速化

FUJITSU

富岳上の大規模機械学習におけるAll-reduce通信の高速化,中村 秋海ら, IPSJ-HPC-193

❶ 富岳上でのAllReduceの高速化

- 双方向Ring-AllReduce
- 6次元メッシュ/トーラスのrankmap
- 繰り返し演算の高速化
- 計算時間の隠蔽

❷ AllReduceの高速化による言語モデル学習の高速化

- PyTorchへの組み込み
- 3D parallelism対応のrankmap
- 大メッセージ長の高速化

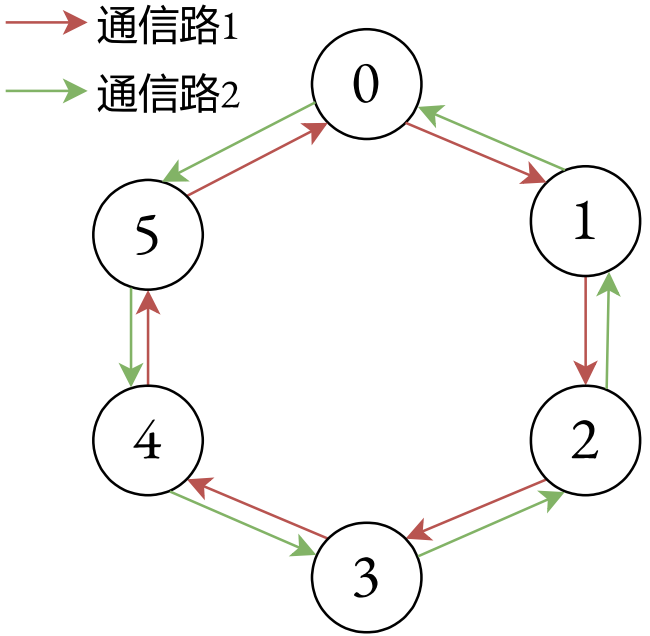
提案手法：双方向1次元Ring-AllReduce

FUJITSU

富岳上の大規模機械学習におけるAll-reduce通信の高速化,中村 秋海ら, IPSJ-HPC-193

ノード間接続が双方向独立通信可能

データ分割+双方向Ring状経路
→帯域を最大限に利用

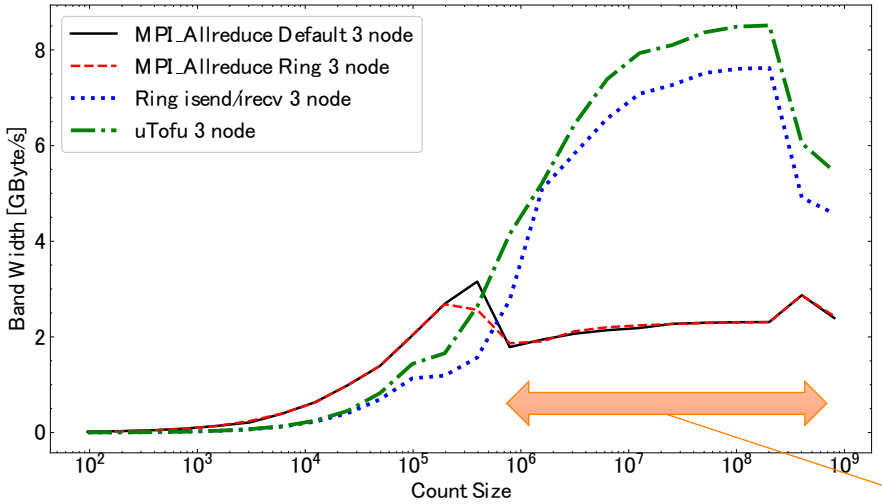


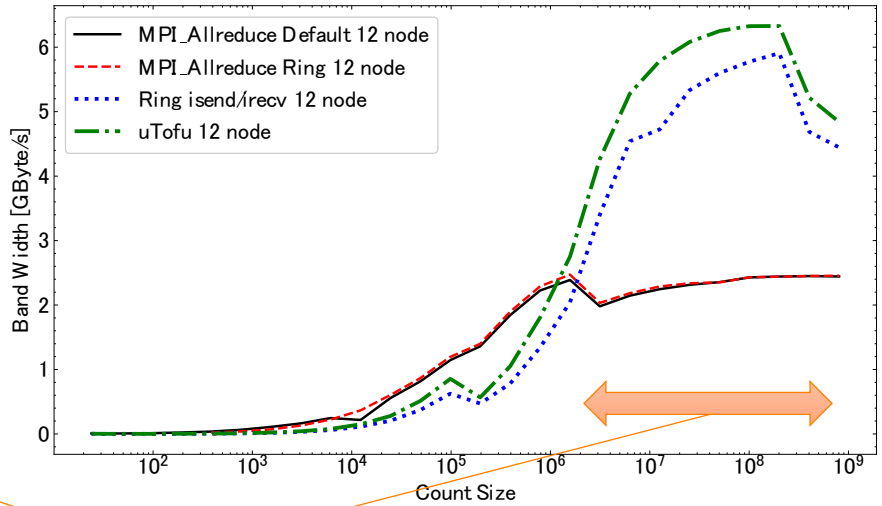
双方向Ring-AllReduceの通信路の例

実験1：結果(3ノード，12ノード)

FUJITSU

富岳上の大規模機械学習におけるAll-reduce通信の高速化,中村 秋海ら, IPSJ-HPC-193

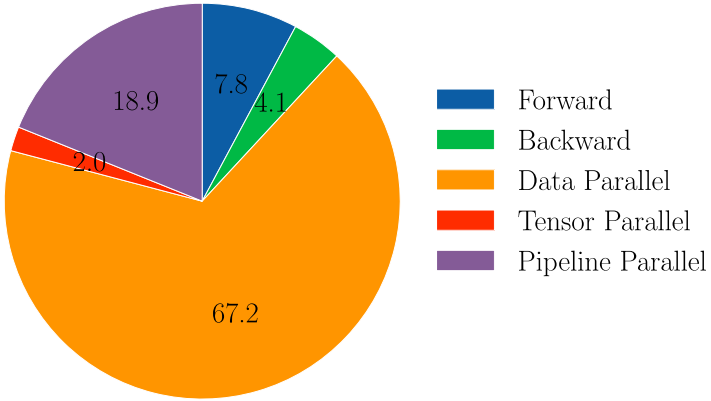




大メッセージ長領域
で速度で上回る

実験2：言語モデルの速度性能

富岳上の大規模機械学習におけるAll-reduce通信の高速化,中村 秋海ら, IPSJ-HPC-193



13Bモデルの時間内訳
TP=6, PP=8, DP=64

$$DP \times TP \times PP$$

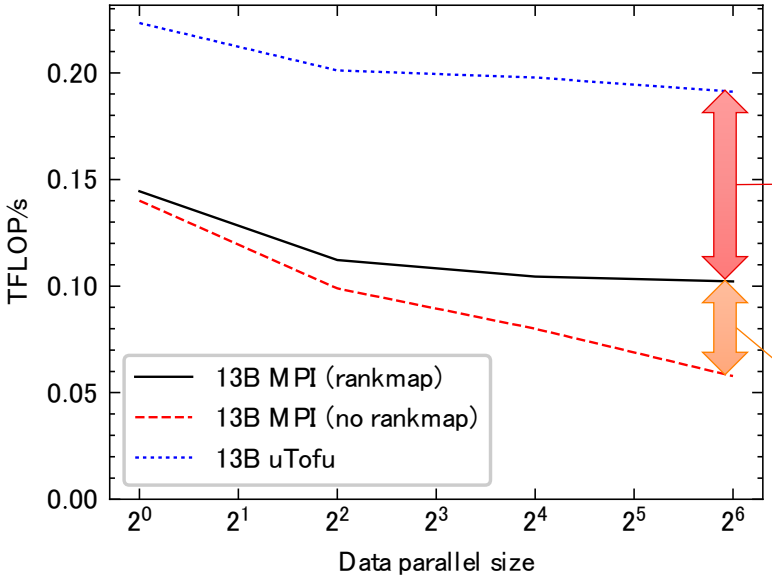
AllReduce　　隣接通信

高速化

AllReduce: Rankmap+提案AllReduce
隣接通信: Rankmap

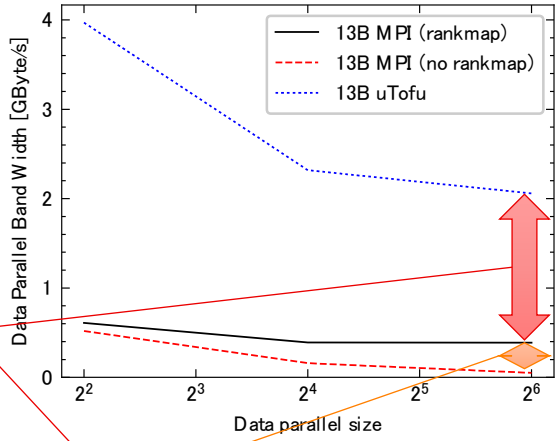
実験2: 結果(13Bモデル)

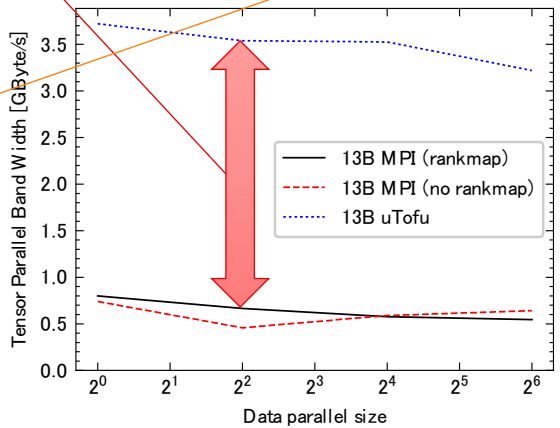
富岳上の大規模機械学習におけるAll-reduce通信の高速化,中村 秋海ら, IPSJ-HPC-193



AllReduce
アルゴリズム
による上昇

Rankmap
による上昇



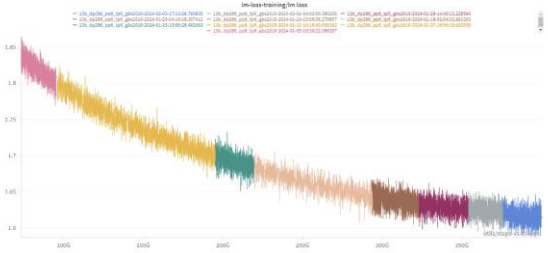


透明性と安全性を担保し、使いやすく日本語性能に優れた130億パラメータの大規模言語モデル

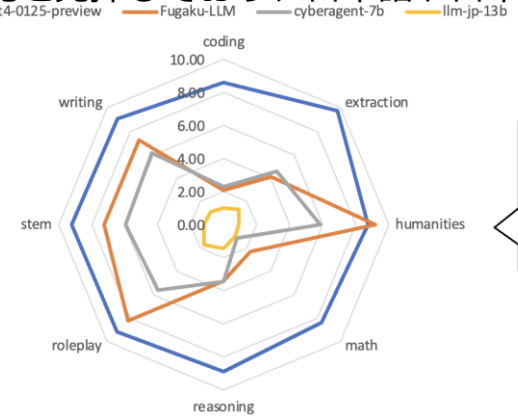
FUJITSU

・「Fugaku-LLM」は、130億パラメータのモデルを一から独自のデータで学習
→ 多くの国産モデルは、国外のオープンなモデルに対して日本語データで継続学習を実施しているのに対し、「Fugaku-LLM」は一から独自のデータで学習しているため、全学習工程を把握でき、透明性と安全性に優れている

・学習の際には「富岳」の1万3,824台の計算ノードが用いられ、約4,000億トークン学習データの約60%が日本語コンテンツ、その他英語、数学、コードの組み合わせで学習
（**事前学習約2か月、事後学習約2か月**）
→ これにより、日本語に強く、Japanese MT-Benchで平均スコア5.5と、国内の独自データで学習しているオープンなモデルにおいて最高性能を達成
→特に人文社会系のタスクでは9.18と高いベンチマーク性能を発揮しており、日本語や日本文化に根差した対話などが期待される



130億パラメータで
約4000億トークンを
学習完了
うち約60%が日本語データ



特に人文社会系のタスクでは9.18と高いベンチマーク性能を発揮しており、日本語や日本文化に根差した対話などが期待

今後の展開

・７者は、世界の研究者やエンジニアが大規模言語モデルの開発に活用できるよう、今回の取り組みで得られた研究成果をGitHubやHugging Faceを通じて公開し、ライセンスに従って誰もが研究および商業目的で利用することができる
→富士通は「Fugaku-LLM」を富士通の先端技術を無償で試せる「Fujitsu Research Portal」を通じて2024年5月10日より提供開始

・多くの研究者や技術者が基盤モデルの改善や新たな応用研究に参画することで、効率的な方法が創出され、科学シミュレーションと生成AIの連携による科学研究サイクルの飛躍的加速などAI基盤モデルを科学研究に活用する「AI for Science」や次世代の革新的な研究やビジネスの成果に繋がることを期待

Thank you