

deepspeed

DeepSpeed: 深層学習の訓練/推論の高速化のためのフレームワーク

<https://github.com/microsoft/DeepSpeed>

田仲 正弘 (Principal Researcher)

DeepSpeed Meetup in Japan
May 23

Model Scale

- 10+ Trillion parameters

Speed

- Fast & scalable training

Democratize AI

- Bigger & faster for all

Compressed Training

- Boosted efficiency

Accelerated inference

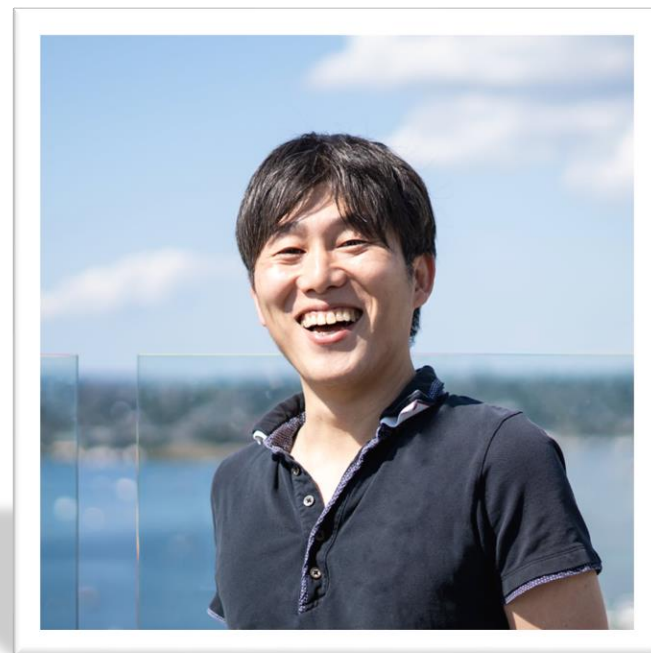
- Faster & cheaper

Usability

- Few lines of code changes

スピーカー

田仲 正弘 (Masahiro Tanaka)



<https://tohtana.github.io/>

- Principal Researcher, DeepSpeed team (2022.12-)
- 前職から大規模分散処理の研究開発（自前の深層学習自動並列化フレームワーク開発、PyTorch Annual Hackathon winner, 産経新聞 先端技術大賞, 他）
- DeepSpeedチームでの主要プロジェクト: DeepSpeed-Ulysses（長い系列のLLM学習並列化）, DeepSpeed-FastGen（テキスト生成の効率化）, ZeRO関連全般, Phi-3モデル, etc.

概要

What is DeepSpeed?

- 大規模かつ高速な**深層学習**を容易に実現する様々な機能を持ったソフトウェア
- オープンソースソフトウェアとしてGitHubで公開中
 - [DeepSpeed](#) (メインのレポジトリ)
 - [DeepSpeedExamples](#) (使用例).
 - [Megatron-DeepSpeed](#) (NVIDIAのMegatron-LMと結合したもの).
 - [DeepSpeed-MII](#) (DeepSpeed-FastGen) (DeepSpeedの高速な推論を容易に利用するためのツール)

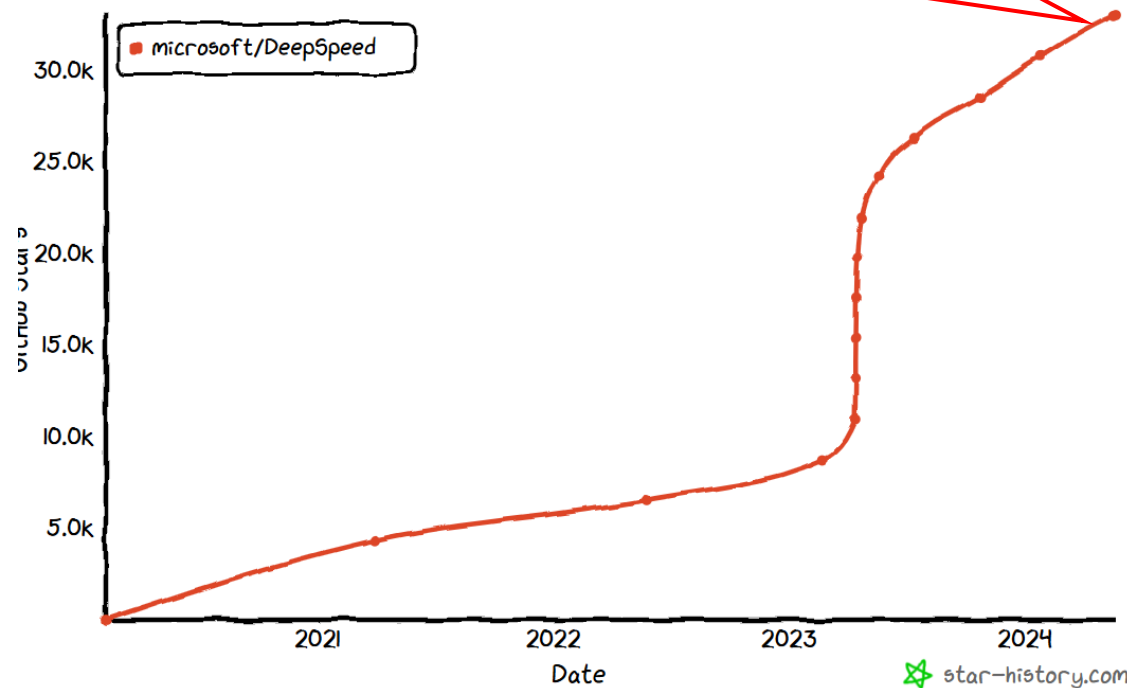


メインレポジトリのURL

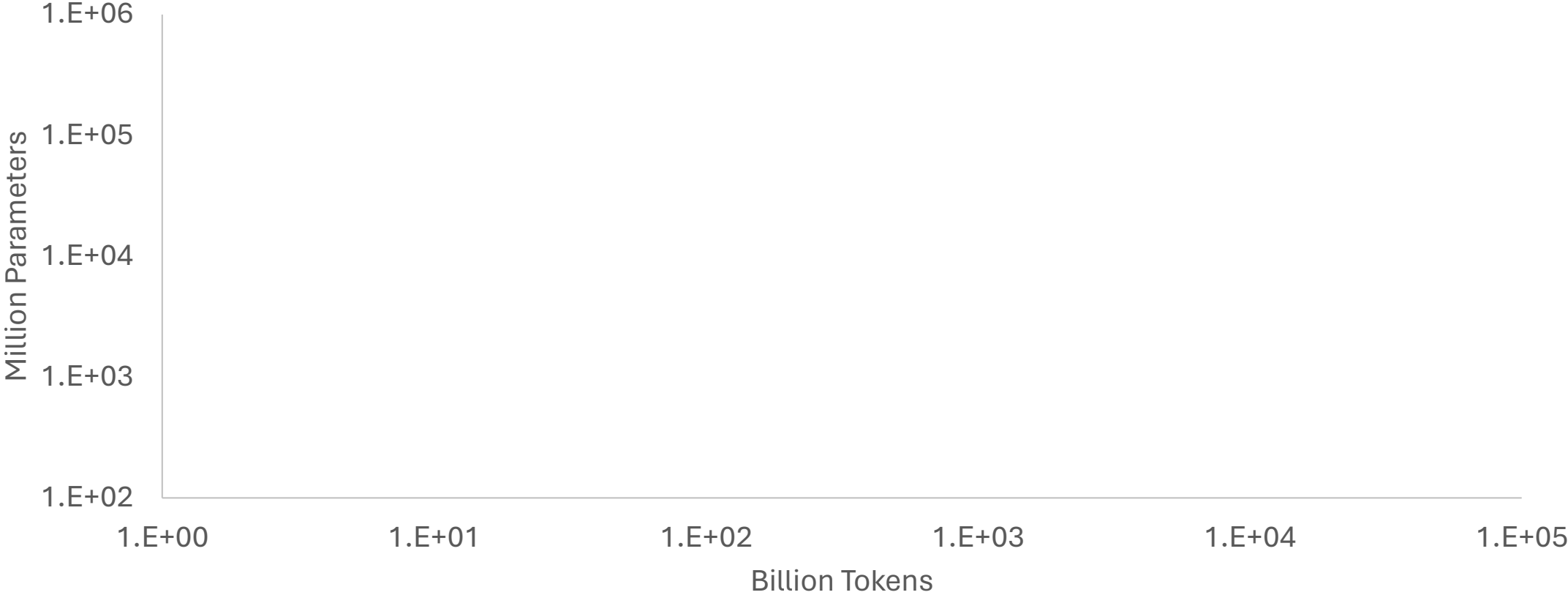
概要

- AI分野のオープンソースソフトウェアとして、もっとも広く使用されているものの一つ
 - 多くの大規模モデルを訓練するために使用
[Megatron-Turing NLG \(530B\)](#), [Jurassic-1 \(178B\)](#), [BLOOM \(176B\)](#), [GLM \(130B\)](#), [YaLM \(100B\)](#).
 - [Hugging Face Transformers](#), [Hugging Face Accelerate](#), [PyTorch Lightning](#), [MosaicML Composer](#), [Determined AI](#) など、多くの著名なオープンソースの深層学習フレームワークのバックエンドとして利用
- Webサイト (deepspeed.ai) でドキュメント・チュートリアルを提供

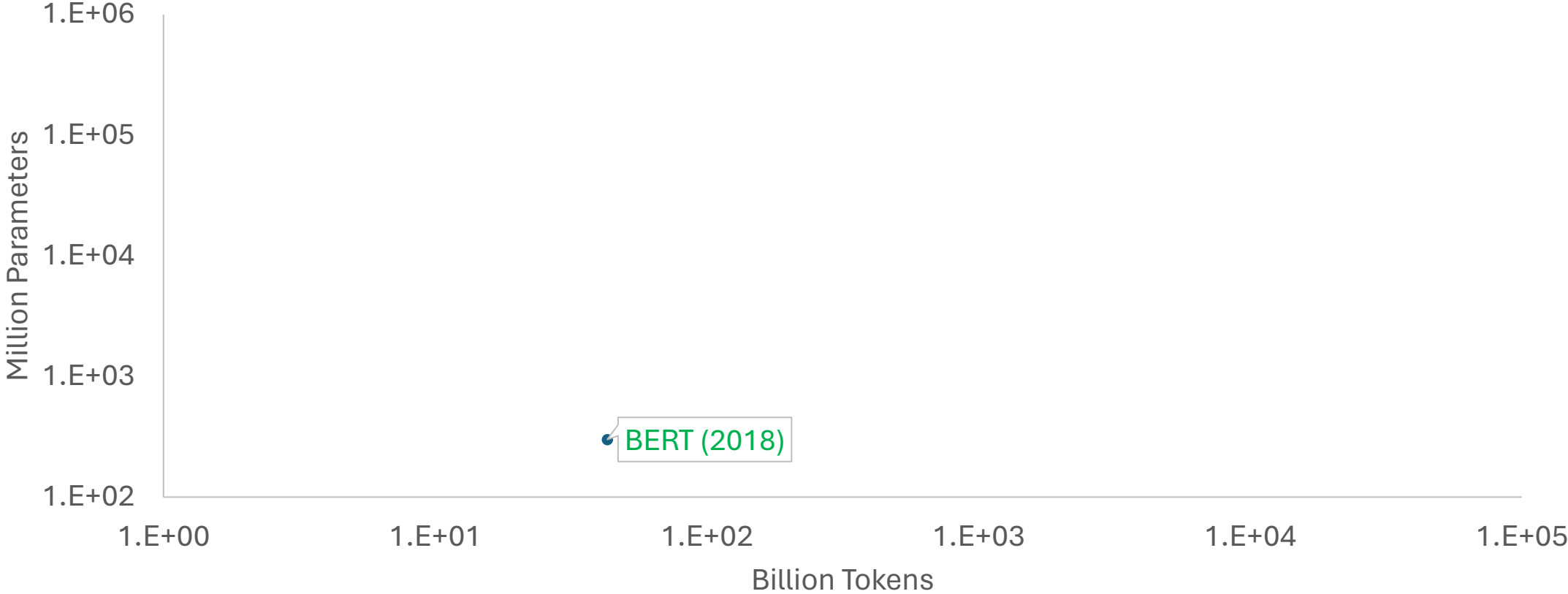
33,000スター（GitHubのMicrosoftのオープンソースソフトウェアとして10位）



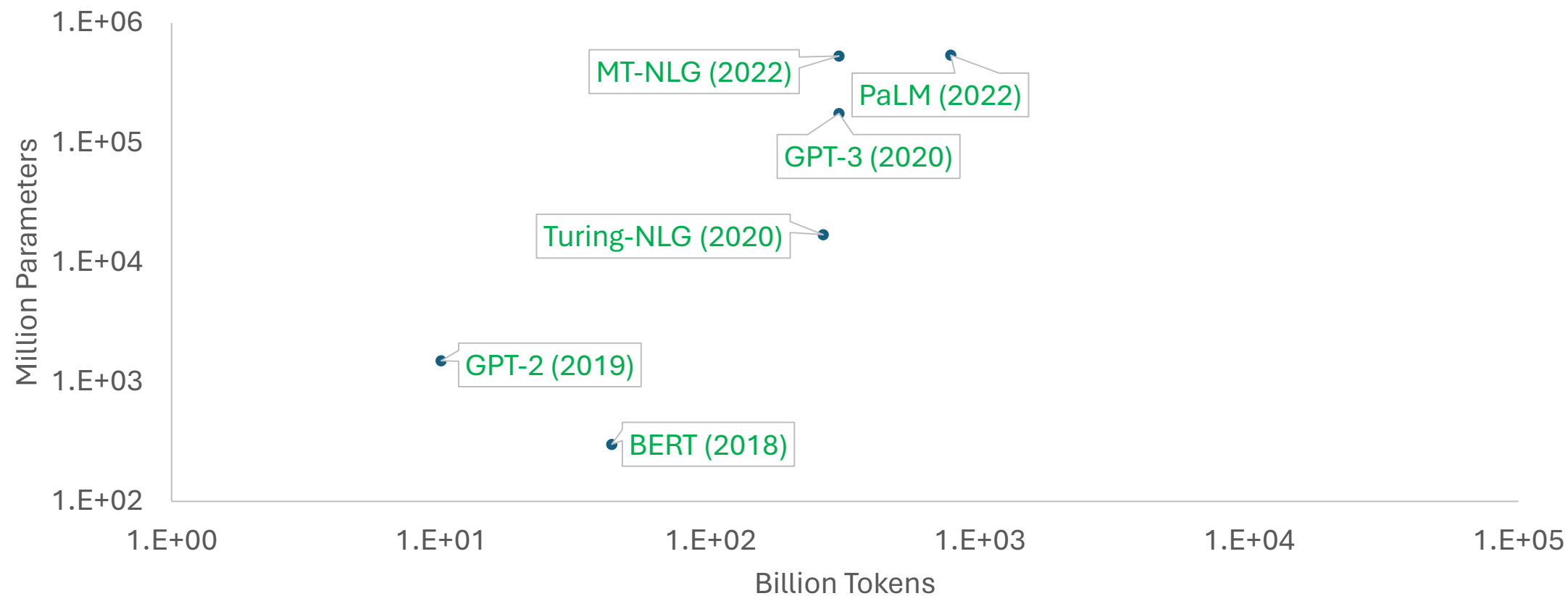
深層学習の規模拡大と性能向上



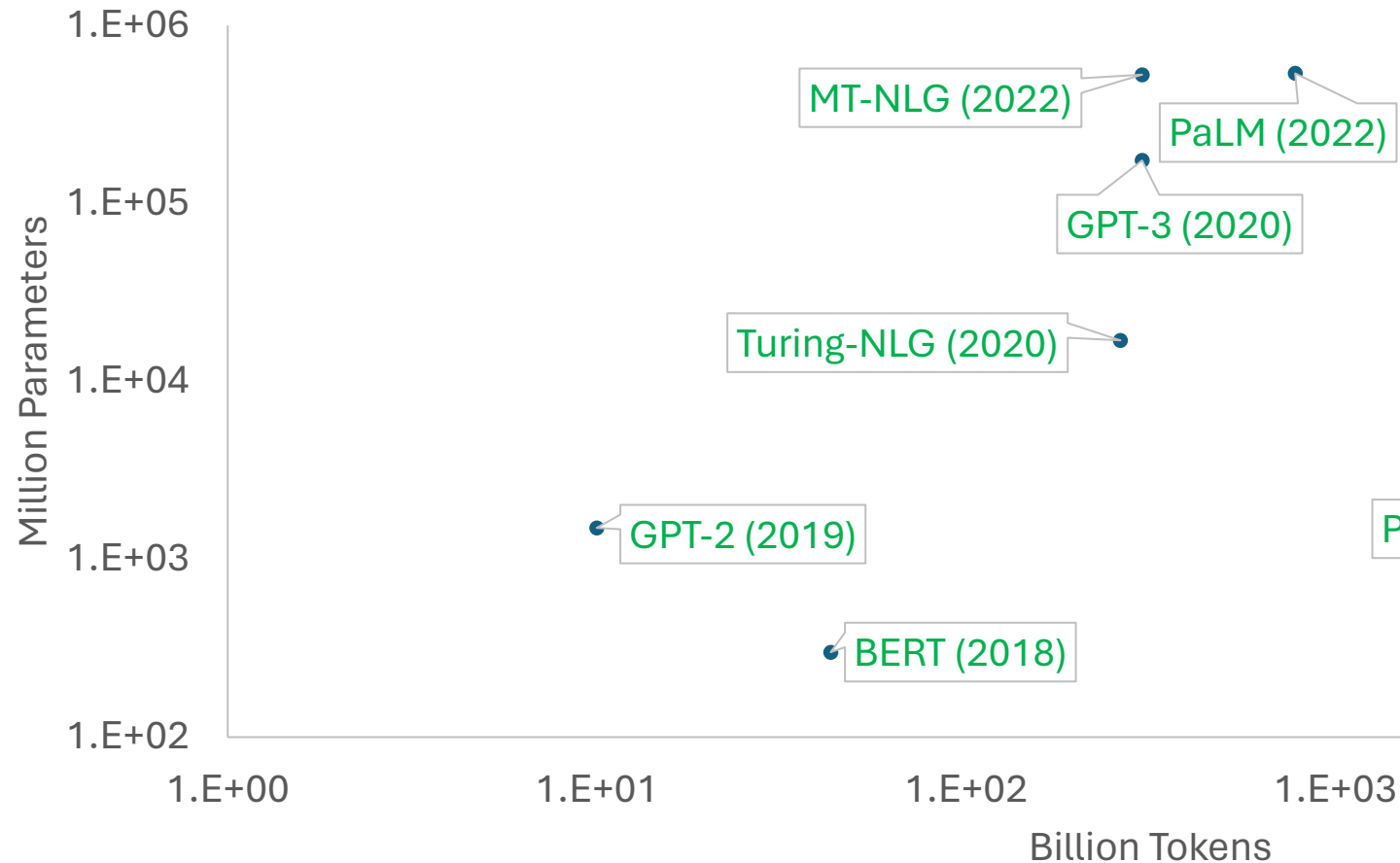
深層学習の規模拡大と性能向上



深層学習の規模拡大と性能向上



深層学習の規模拡大と性能向上



ZDNET Home / Innovation / Artificial Intelligence

Microsoft launches Phi-3 Mini, an AI model that is smaller but still rivals GPT-3.5

ars TECHNICA SUBSCRIBE

SMALL PACKAGES —
Microsoft's Phi-3 shows the surprising power of small, locally run AI language models

深層学習の課題 – 訓練 (Training) –

深層学習では、大量の訓練データを与えて、モデルを訓練するが、計算量・メモリ量・処理データ量がとにかく多い

- 現実的な時間で訓練を終えるには、GPU等のアクセラレータ（高価）が必要
- モデル規模が大きくなる（=学習パラメータが多くなる）、または訓練データの量が増えると、必要な計算量・メモリ量も増え、多数のGPUを用いた並列処理が必要
→ 数千億パラメータ規模のモデルの訓練には、数千GPUを用いても2ヶ月 (e.g., [MT-NLG](#))



DeepSpeedは深層学習の訓練における様々な技術的課題を解決

課題	DeepSpeed の機能
モデルが大きすぎてGPUメモリに収まらない	ZeRO (Zero Redundancy Optimizer)
数千GPU規模までスケールさせたい	3D parallelism
高い計算効率のモデルを使用したい	DeepSpeed-MoE
GPU間の通信が遅い	ZeRO++, Communication Compression
大規模なデータが必要	DeepSpeed Data Efficiency
ChatGPTにも使用されるRLHF 訓練を効率的に実行	DeepSpeed-Chat

深層学習の課題 –推論 (Inference) –

- 訓練されたモデルを使用するフェーズを推論という
- 実サービスで、多数のユーザによって実行されるのは推論 (e.g., new Bing)
- 訓練より計算量は少ないものの、低レイテンシ（高速な応答）、低コスト化などの要件が重要



DeepSpeedは推論のための様々な機能も提供

課題	DeepSpeedの機能
高速なテキスト生成	DeepSpeed-FastGen
高速・スケーラブルな推論	DeepSpeed Inference
簡単にデプロイしたい	DeepSpeed-MII
MoEモデルの推論	DeepSpeed-MoE
巨大モデルを高速に推論	DeepSpeed Compression

ZeRO, ZeRO-Offload, ZeRO-Infinity

Breaking the GPU Memory Wall for DL Training

ZeRO: Memory Optimizations Toward Training Trillion Parameter Models

Samyam Rajbhandari*, Jeff Rasley*, Olatunji Ruwase, Yuxiong He
{samyamr, jerasley, oluwase, yuxhe}@microsoft.com

ABSTRACT

Large deep learning models offer significant accuracy gains, but training billions to trillions of parameters is challenging. Existing solutions such as data and model parallelisms exhibit fundamental limitations to fit these models into limited device

common settings like mixed precision and ADAM optimizer [6]. Other existing solutions such as Pipeline Parallelism (PP), Model Parallelism (MP), CPU-Offloading, etc, make trade-offs between functionality, usability, as well as memory and compute/communication efficiency, all of which are crucial to

ZeRO-Offload: Democratizing Billion-Scale Model Training

Jie Ren*, Samyam Rajbhandari†, Reza Yazdani Aminabadi†, Olatunji Ruwase†
Shuangyan Yang*, Minjia Zhang†, Dong Li*, Yuxiong He†

†Microsoft, *University of California, Merced
{jren6, syang127, dli35}@ucmerced.edu, {samyamr, yazdani.reza, oluwase, minjiaz, yuxhe}@microsoft.com

Abstract

Large-scale model training has been a playing ground for a limited few requiring complex model refactoring and access to

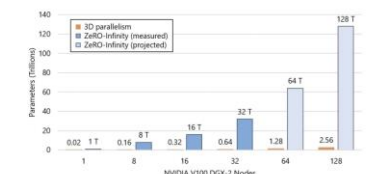
parameters. With the three orders of magnitude growth in model size since 2017, the model accuracy continues to improve with the model size [12]. Recent studies in fact show that larger models are more resource-efficient to train than

ZeRO-Infinity: Breaking the GPU Memory Wall for Extreme Scale Deep Learning

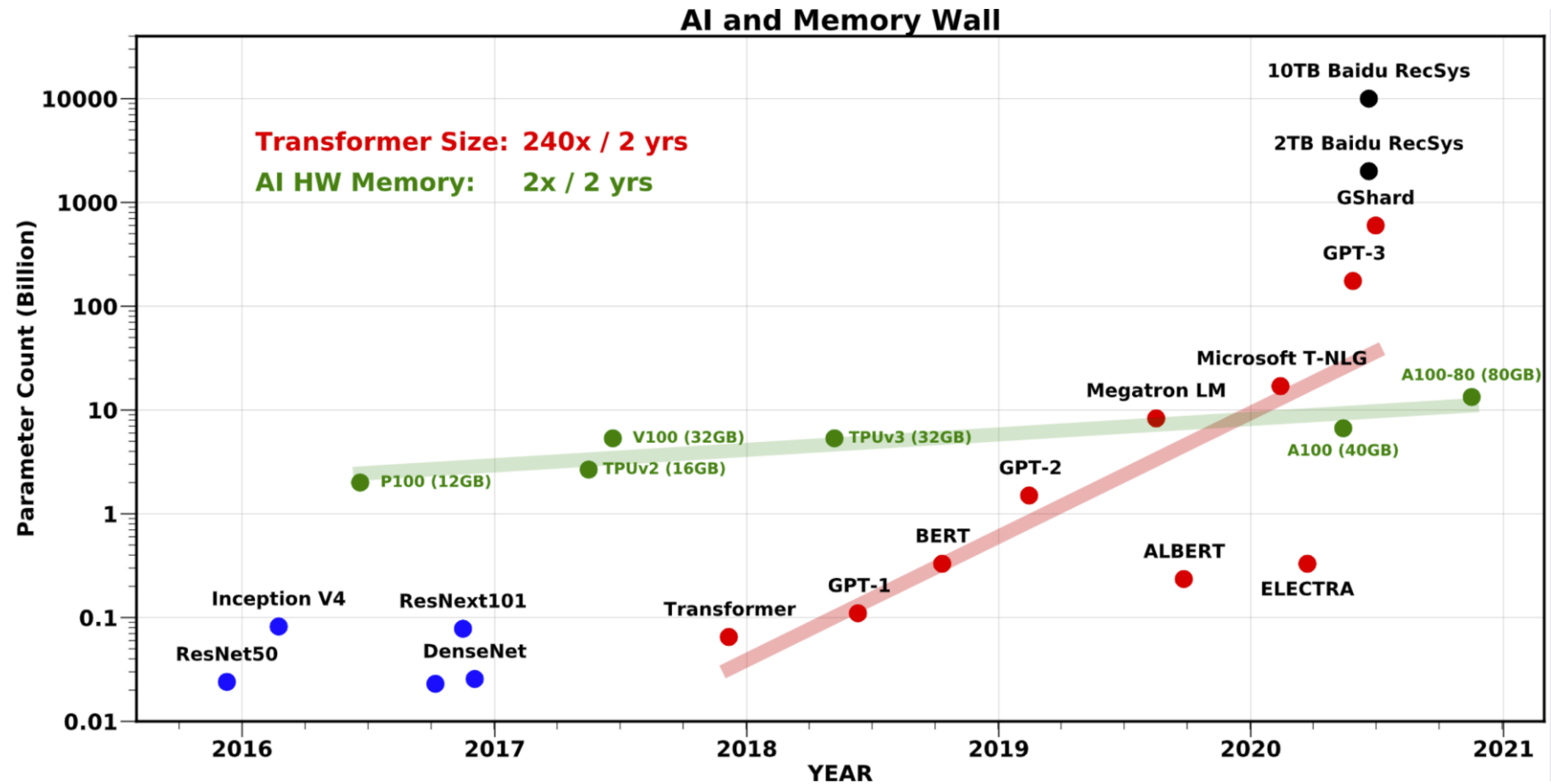
Samyam Rajbhandari, Olatunji Ruwase, Jeff Rasley, Shaden Smith, Yuxiong He
{samyamr, oluwase, jerasley, shsmi, yuxhe}@microsoft.com

ABSTRACT

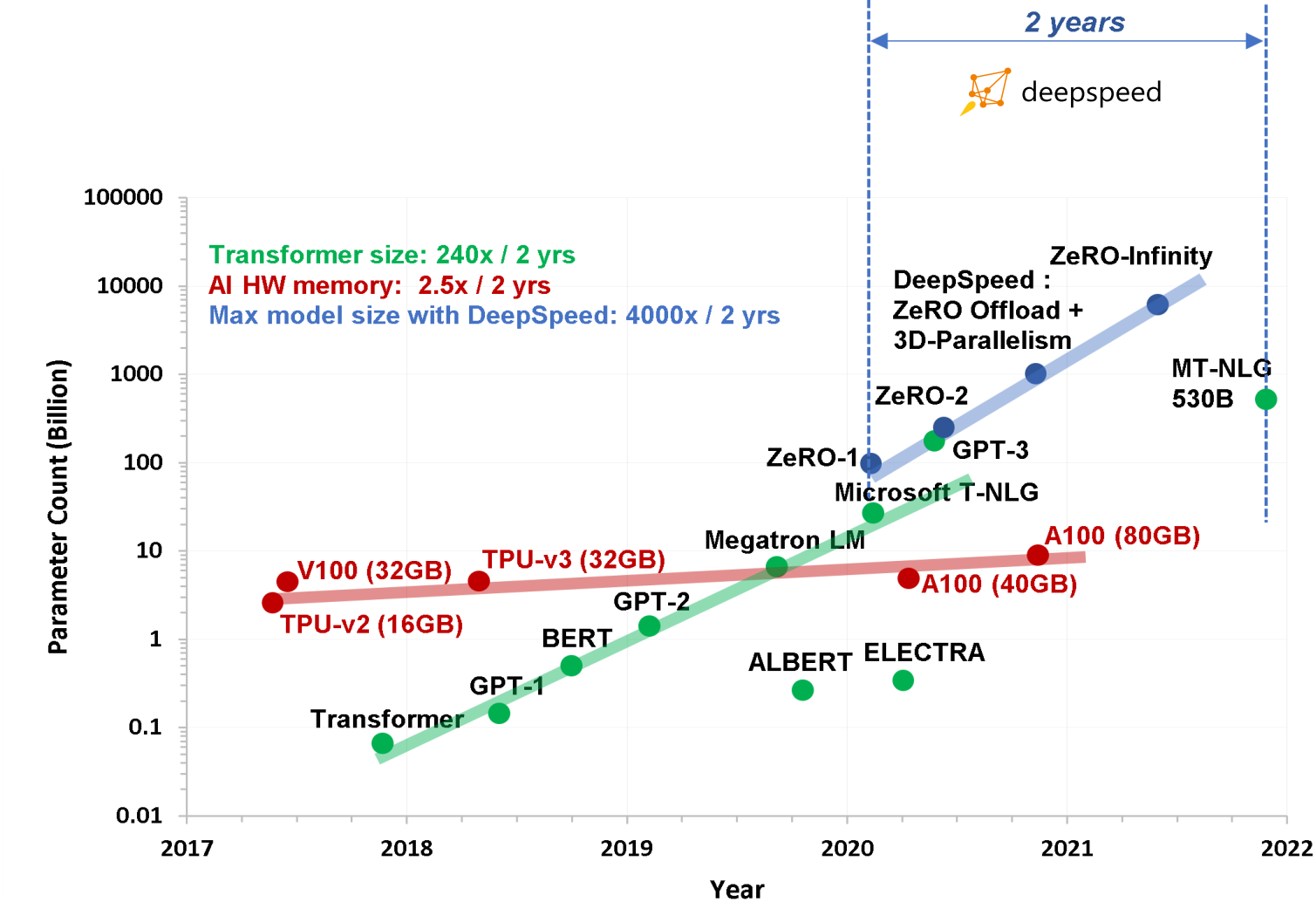
In the last three years, the largest dense deep learning models have grown over 1000x to reach hundreds of billions of parameters, while the GPU memory has only grown by 5x (16 GB to 80 GB). Therefore, the growth in model scale has been supported primarily through system innovations that allow large models to fit in the aggregate GPU memory of multiple GPUs. However, we are getting close to the GPU memory wall. It requires 800 NVIDIA V100 GPUs just to fit a trillion parameter model for training, and such clusters are simply out of reach for most data scientists. In addition, training models at that scale requires complex combinations of parallelism



大規模モデル学習の概観



DeepSpeed: Reshaping the Large Model Training Landscape



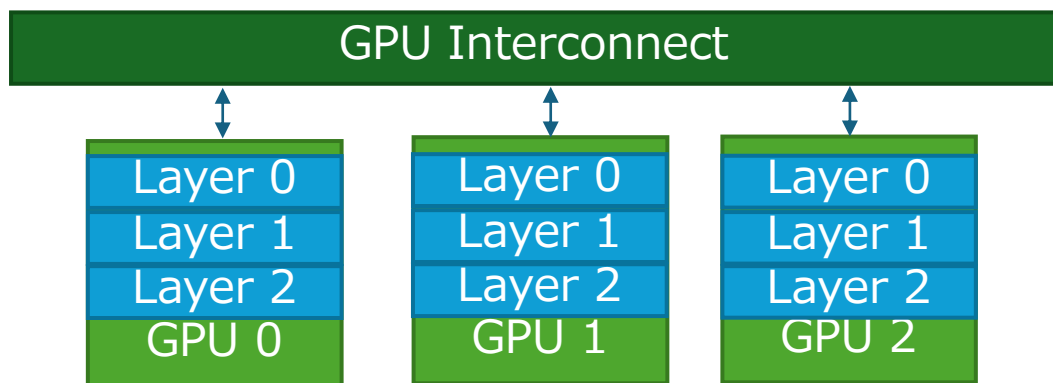
System capability to efficiently train models with *trillions of parameters*

*AI and Memory Wall. (This blogpost has been written in... | by Amir Gholami | riselab | Medium

What is ZeRO?

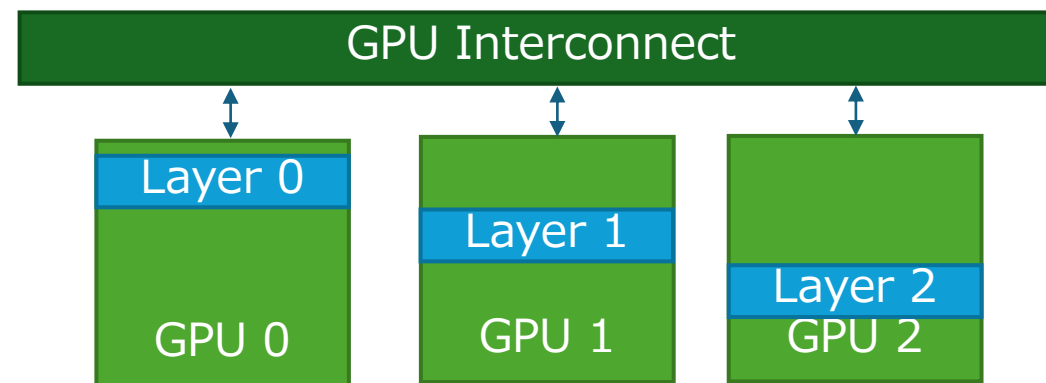
- データ並列のメモリ利用を効率化
→ 巨大モデルを高い効率で学習、かつどのようなモデルアーキテクチャでも適用可能

通常データ並列



Model States mapping in **Data Parallel** Training

ZeRO (**Z**ero **R**edundancy **O**ptimizer)

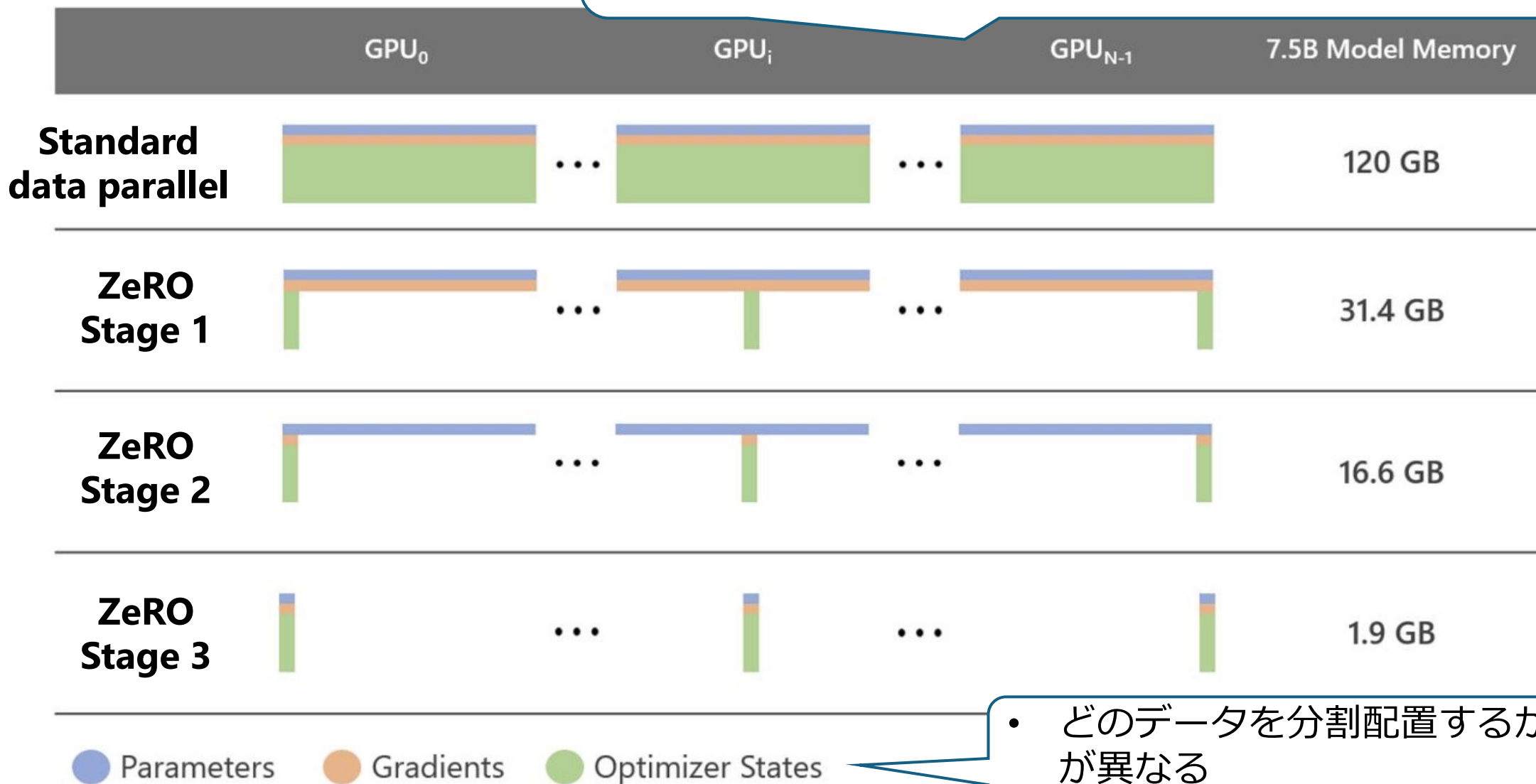


Model States mapping in **ZeRO** Training

- 通常データ並列では重複して持つ学習パラメータを、重複のないように各GPUに格納
- データが必要になった際に、GPU間の通信で必要な部分だけデータを収集し、使い終わったら捨てる

What is ZeRO?

- パラメータを集めるのに通信オーバーヘッドが生じる
- 複数の動作モード (Stage) : 省メモリ効果と通信オーバーヘッドのトレードオフが異なる

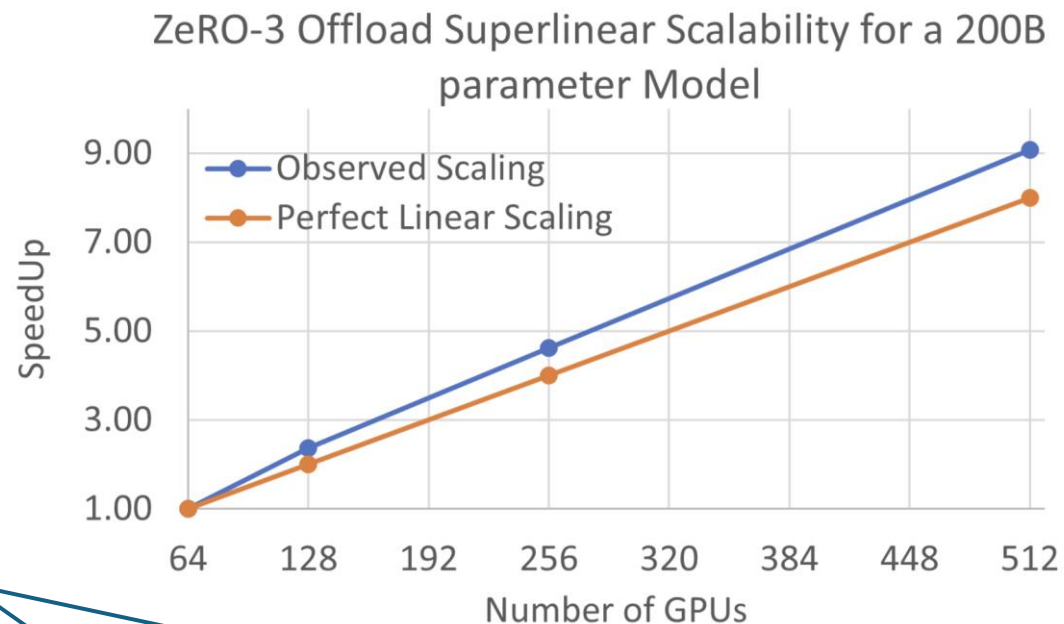
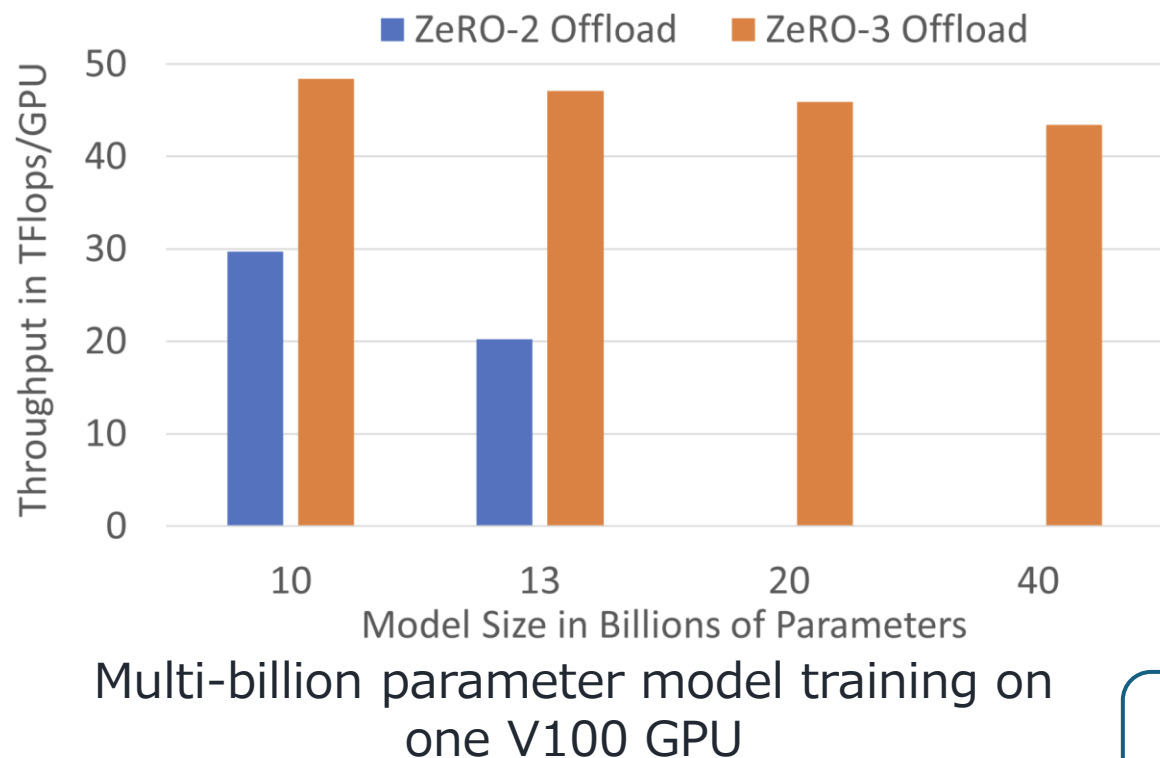


- どのデータを分割配置するかが異なる

7.5B parameter model, Adam optimizer, 64 GPUs

ZeRO-Offload - 誰もが巨大なモデルを学習できるように -

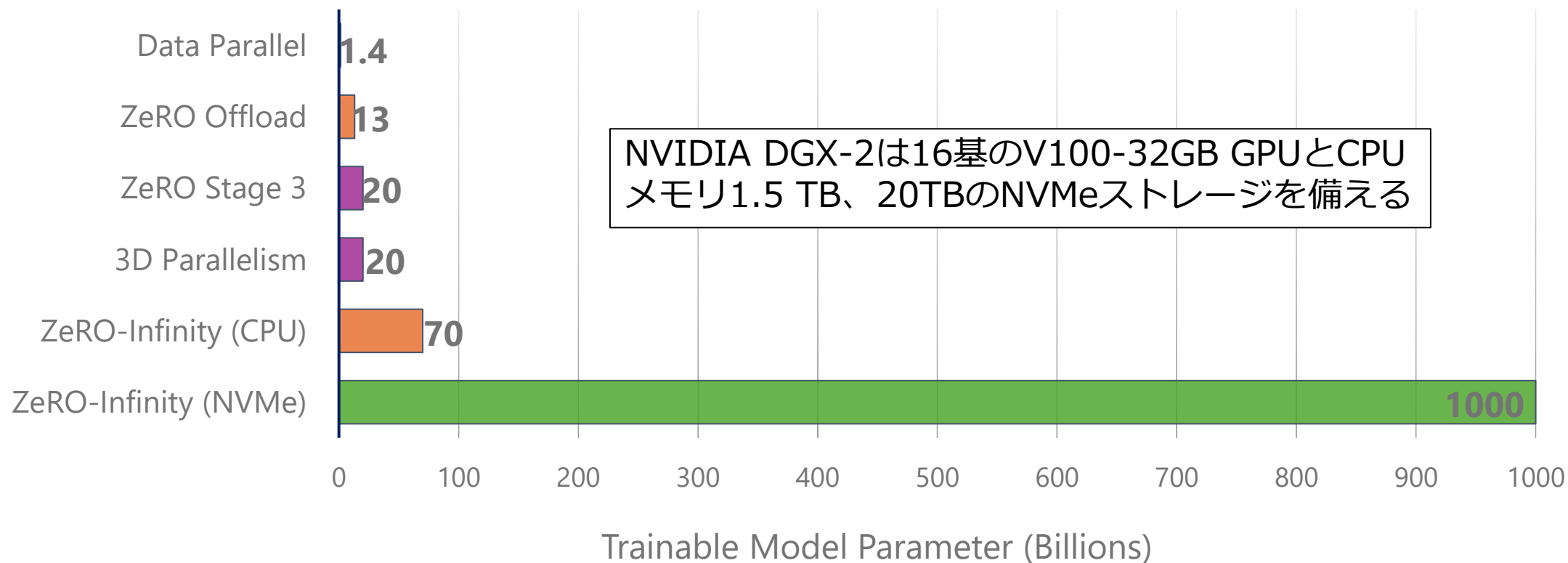
- 学習パラメータ等のデータをGPUメモリからCPUメモリにオフロード
- ZeRO-3との組合せにより、**400億パラメータのモデルを一基のGPUだけで学習**



各種の技術的工夫でオフロードのオーバーヘッドを減少

ZeRO-Infinity - 極限のモデルサイズへの挑戦 -

- CPUメモリとNVMeストレージの両方にオフロード
- 多くの省メモリ技術を組合せ: ZeRO Stage 3, チェックポイントニングなど



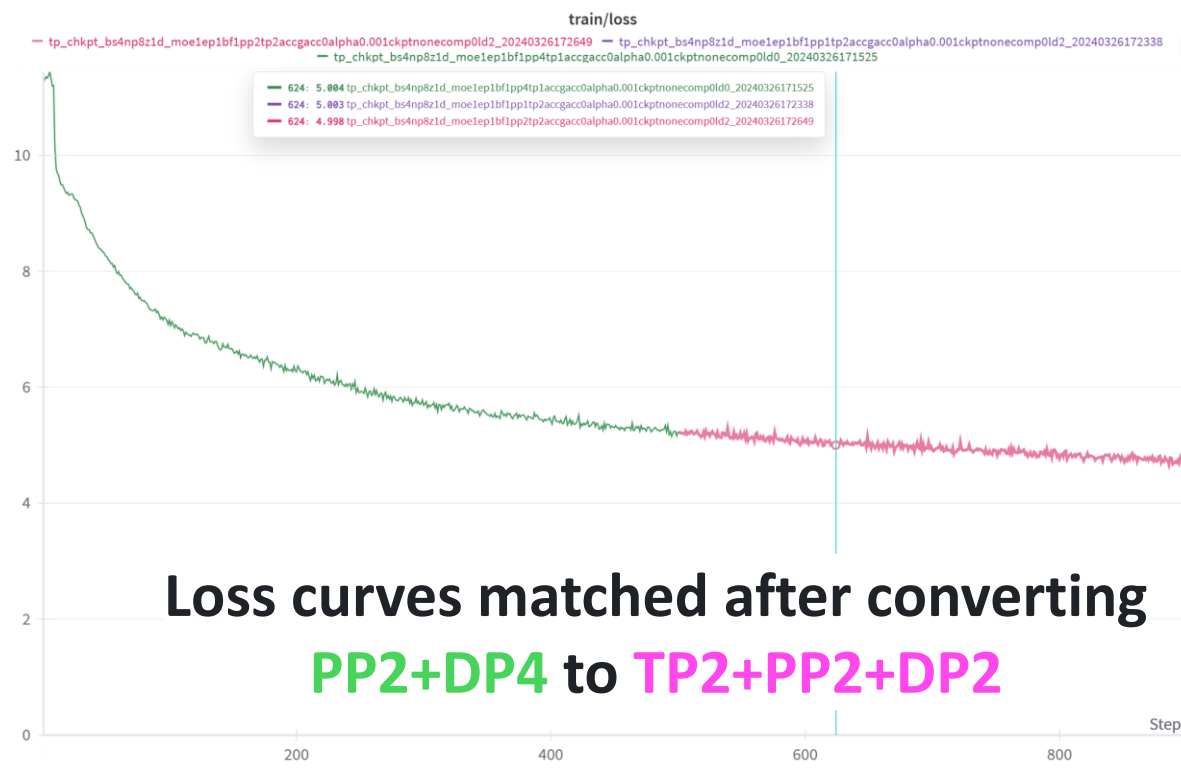
1基のNVIDIA DGX-2で学習できる最大モデルサイズ

Universal Checkpoint

- 学習の途中で実行環境や設定を変更したいことはよくある
 - 利用可能なGPU数が増減
 - 大規模学習をメインで走らせるのと並行して、（より少ないGPUで）ハイパーパラメータ探索をしたい
- Universal Checkpoint: 学習途中で保存したモデル及びOptimizerの状態（Checkpoint）を異なる並列設定で読み込んで学習を再開
 - ZeROの分割数
 - テンソル並列・パイプライン並列・シーケンス並列

Universal Checkpoint

- BLOOM (1760億パラメータのオープンモデル)、先日公開された Phi-3などでも活用
- 正確な変換を実証済み



ZeRO++

Towards the next level of scalability of ZeRO

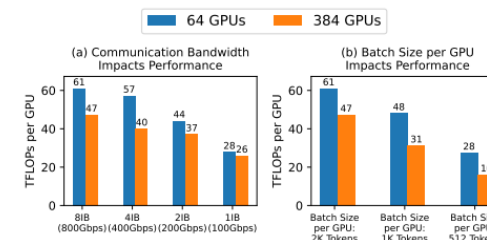
ZeRO++: Extremely Efficient Collective Communication for Giant Model Training

Guanhua Wang*, Heyang Qin*, Sam Ade Jacobs, Connor Holmes, Samyam Rajbhandari
Olatunji Ruwase, Feng Yan¹, Lei Yang², Yuxiong He
Microsoft

{*guanhuawang, heyangqin, samjacobs, connorholmes, samyamr, olruwase, yuxhe*}@microsoft.com

ABSTRACT

Zero Redundancy Optimizer (ZeRO) has been used to train a wide range of large language models on massive GPUs clusters due to its ease of use, efficiency, and good scalability. However, when training on low-bandwidth clusters, or at scale which forces batch size per GPU to be small, ZeRO's effective throughput is limited because of high communication volume from gathering weights in forward pass, backward pass, and averaging gradients. This paper introduces three communication volume reduction techniques, which we collectively refer to as ZeRO++, targeting each of the communication



ZeROの通信量

- ZeRO Stage 3 はモデルのパラメータを分割して配置
-> 分割配置されたデータを集めるための通信がオーバーヘッドになる
- 通信データ量の内訳 (モデルパラメータを M とする):
 1. Forward (パラメータのall-gather) : M
 2. Backward (パラメータのall-gather) : M
 3. gradientのreduce-scatter: M
- **合計の通信量: $3M$**

どのように通信量を減らすか？

ZeRO++の工夫

1. Forward all-gather (size M)

Accurate & Efficient Quantization

(size $0.5M$)

2. Backward all-gather (size M)

Heterogeneous Partitioning (hpZeRO)

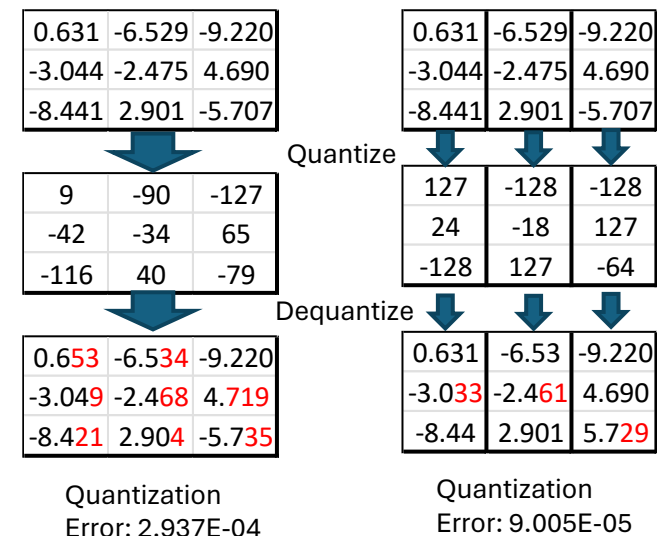
(size 0)

3. Backward reduce-scatter (size M)

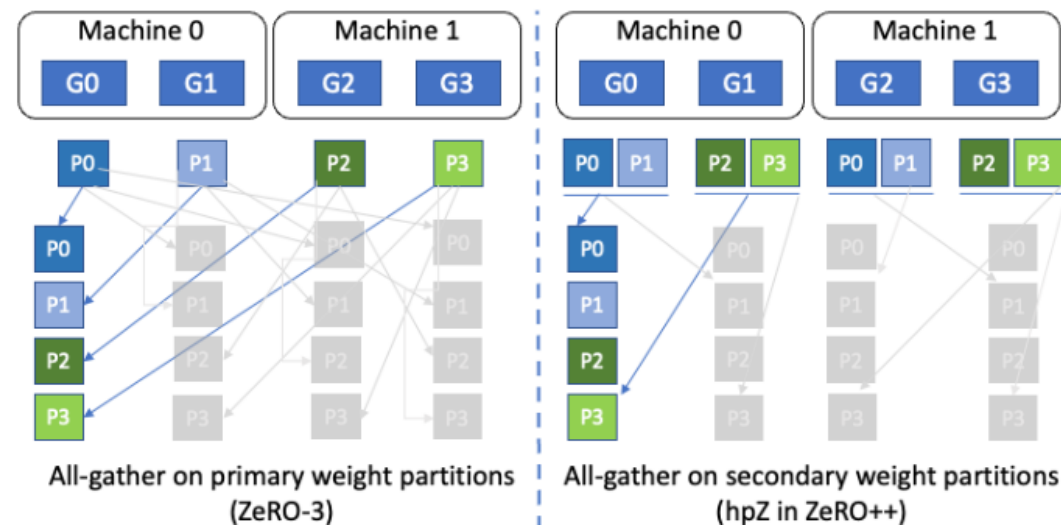
Novel Quantized Collective

(size $0.25M$)

合計の通信量削減: $3M \rightarrow 0.75M$



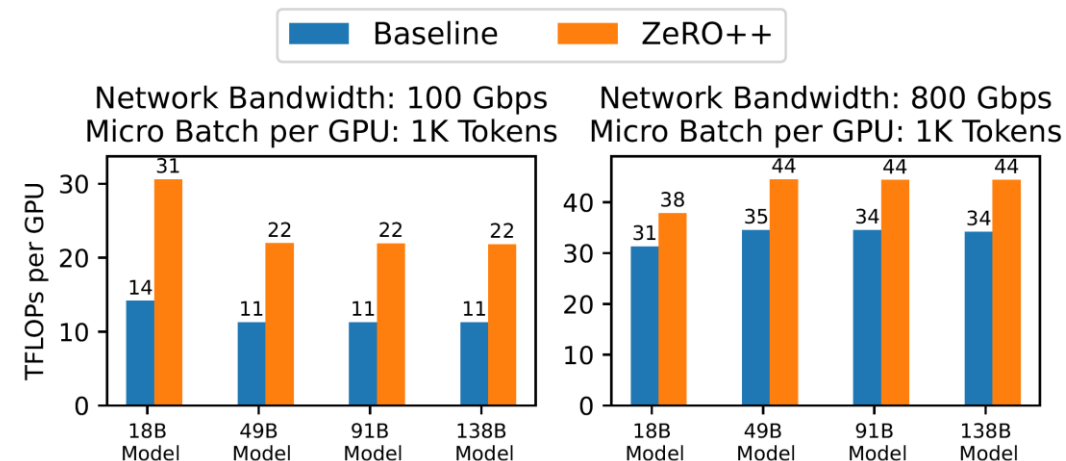
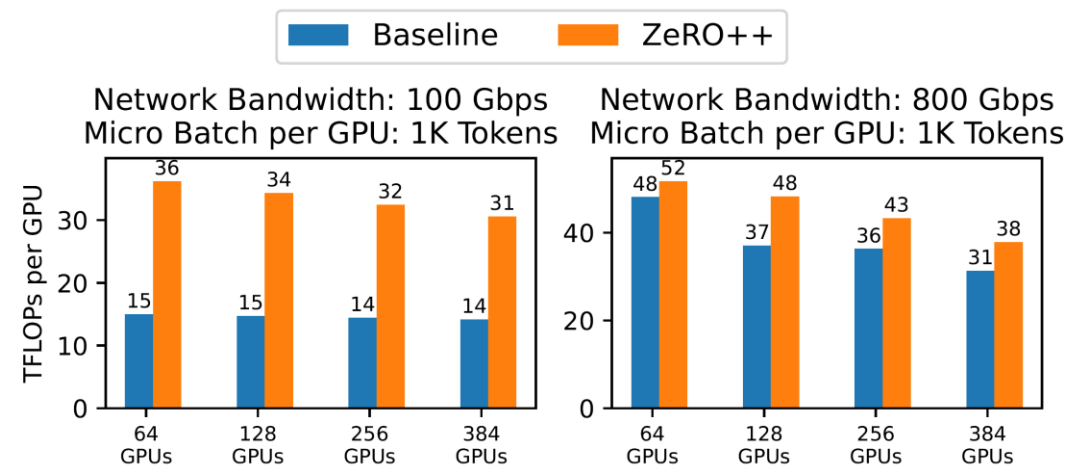
パラメータのQuantization



階層的なパラメータの分割

E2E Evaluation

- 通信帯域が限られているときに特に大きな高速化
- 異なるGPU数での評価
 - 100Gbps: 121% - 140% speedup
 - 800Gbps: 8% - 30% speedup
- 異なるモデルサイズでの評価
 - 800Gbps: up to 30% speedup
 - 100Gbps: over 100% speedup
 - 10Gbps: up to 300% speedup



DeepSpeed Mixture of Experts (MoE)

Improving Compute Efficiency for DL scaling

DeepSpeed-MoE: Advancing Mixture-of-Experts Inference and Training to Power Next-Generation AI Scale

Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, Yuxiong He *Proceedings of the 39th International Conference on Machine Learning*, PMLR 162:18332-18346, 2022.

Abstract

As the training of giant dense models hits the boundary on the availability and capability of the hardware resources today, Mixture-of-Experts (MoE) models have become one of the most promising model architectures due to their significant training cost reduction compared to quality-equivalent dense models. Their training cost saving is demonstrated from encoder-decoder models (prior works) to a 5x saving for auto-aggressive language models (this work). However, due to the much larger model size and unique architecture, how to provide fast MoE model inference remains challenging and unsolved, limiting their practical usage. To tackle this, we present DeepSpeed-MoE, an end-to-end MoE training

A Hybrid Tensor-Expert-Data Parallelism Approach to Optimize Mixture-of-Experts Training

Siddharth Singh
ssingh37@umd.edu
Department of Computer Science,
University of Maryland
College Park, Maryland, USA

Olatunji Ruwase
olruwase@microsoft.com
Microsoft, Inc.
Redmond, Washington, USA

Ammar Ahmad Awan
ammar.awan@microsoft.com
Microsoft, Inc.
Redmond, Washington, USA

Samyam Rajbhandari
samyamr@microsoft.com
Microsoft, Inc.
Redmond, Washington, USA

Yuxiong He
yuxhe@microsoft.com
Microsoft, Inc.
Redmond, Washington, USA

Abhinav Bhatele
bhatele@cs.umd.edu
Department of Computer Science,
University of Maryland
College Park, Maryland, USA

ABSTRACT

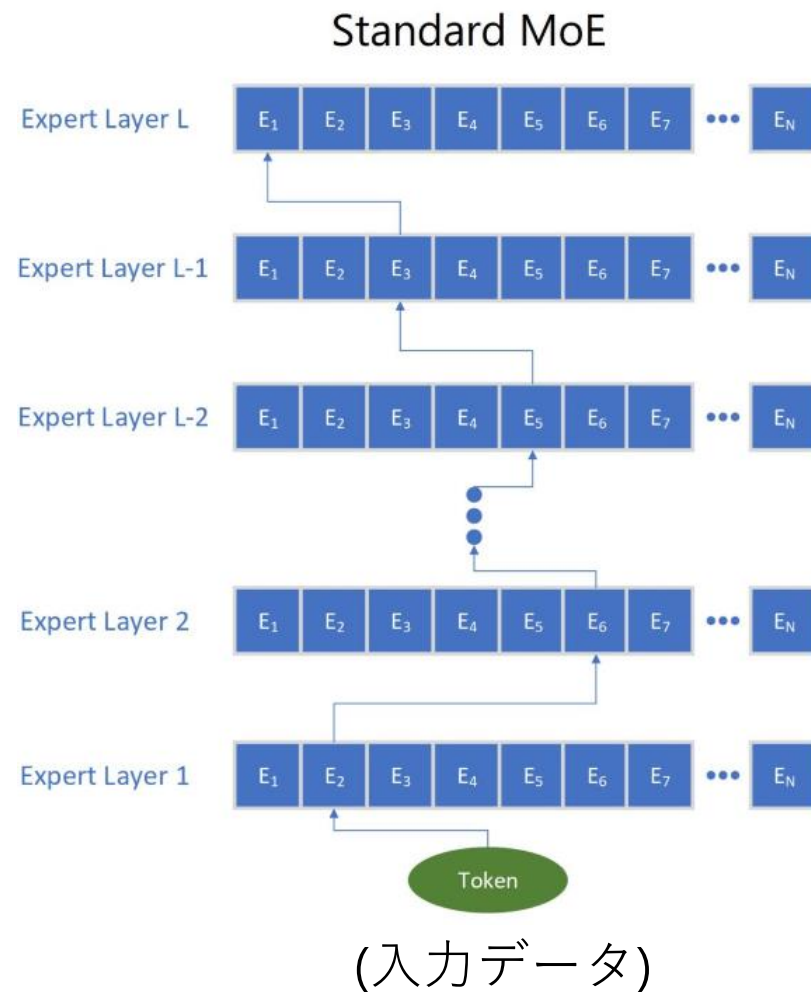
Mixture-of-Experts (MoE) is a neural network architecture that adds sparsely activated expert blocks to a base model, increasing the number of parameters without impacting computational costs.

1 INTRODUCTION

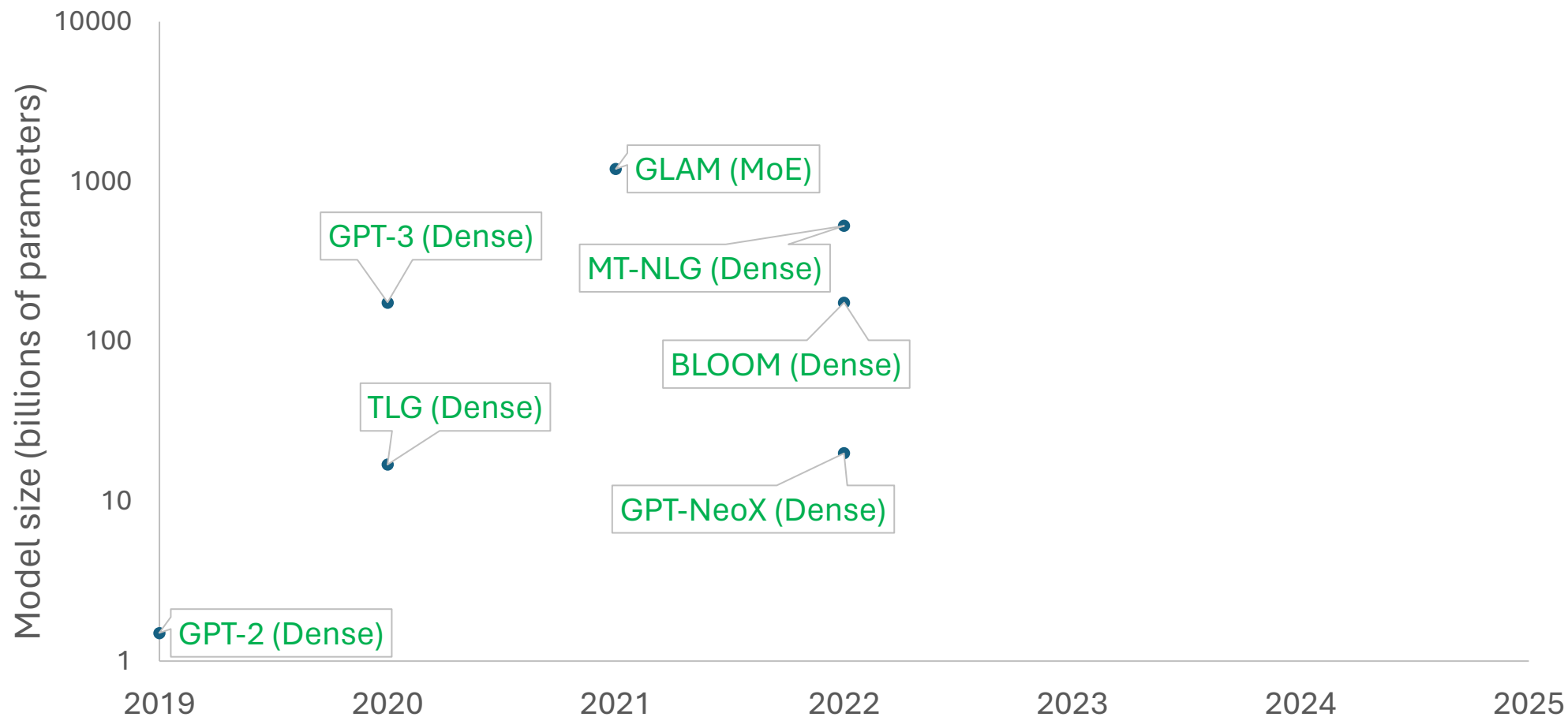
Contemporary state-of-the-art AI algorithms have come to rely on neural networks such as GPT-3 [4] and MT-NLG [34] with hundreds of billion of parameters. However, training or running inference

Mixture of Experts (MoE)

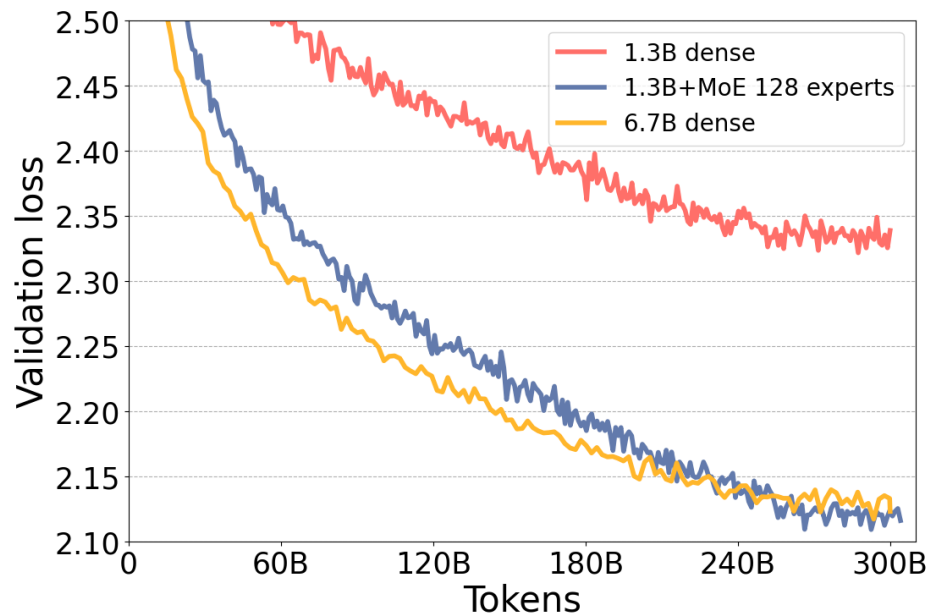
- ZeROや各種の並列化を用いても、大規模モデルの訓練は膨大な計算量、時間、コストが必要
- Mixture of Experts (MoE) と呼ばれる新しいタイプのモデルアーキテクチャでは、入力データに応じてモデルの一部のみを計算



MoEモデルの規模拡大



DeepSpeed-MoE



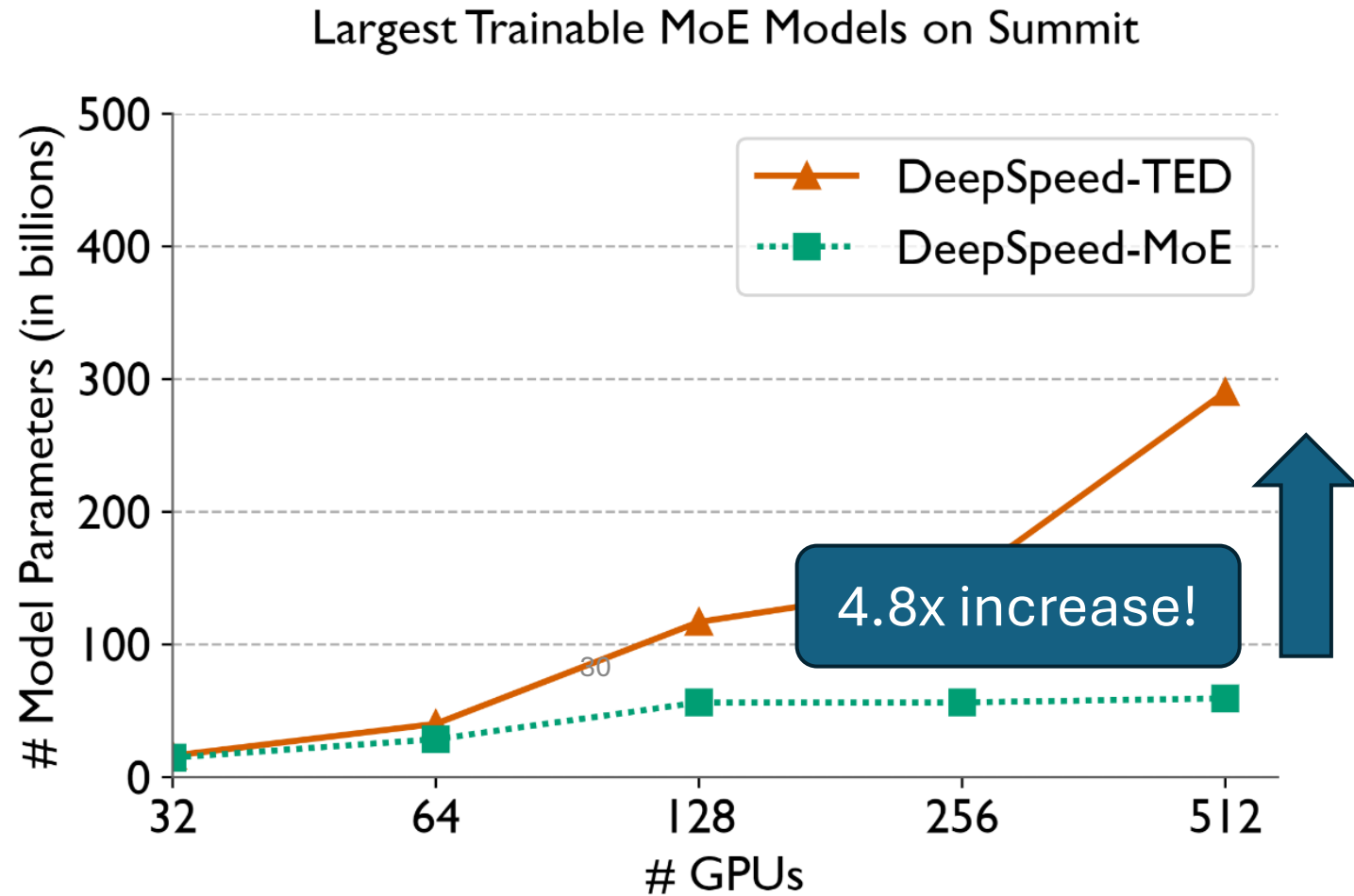
1/5 の計算コストで
同等のモデル品質を達成

Case	LAMBADA: completion prediction	PIQA: commonsense reasoning	BoolQ: reading comprehension	RACE-h: reading comprehension	TriviaQA: question answering	WebQs: question answering
Dense NLG:						
1.3B	63.65	73.39	63.39	35.60	10.05	3.25
6.7B	71.94	76.71	67.03	37.42	23.47	5.12
Standard MoE NLG:						
1.3B+MoE-128	69.84	76.71	64.92	38.09	31.29	7.19

Deepspeed-TED: 複数の並列化手法の組み合わせ

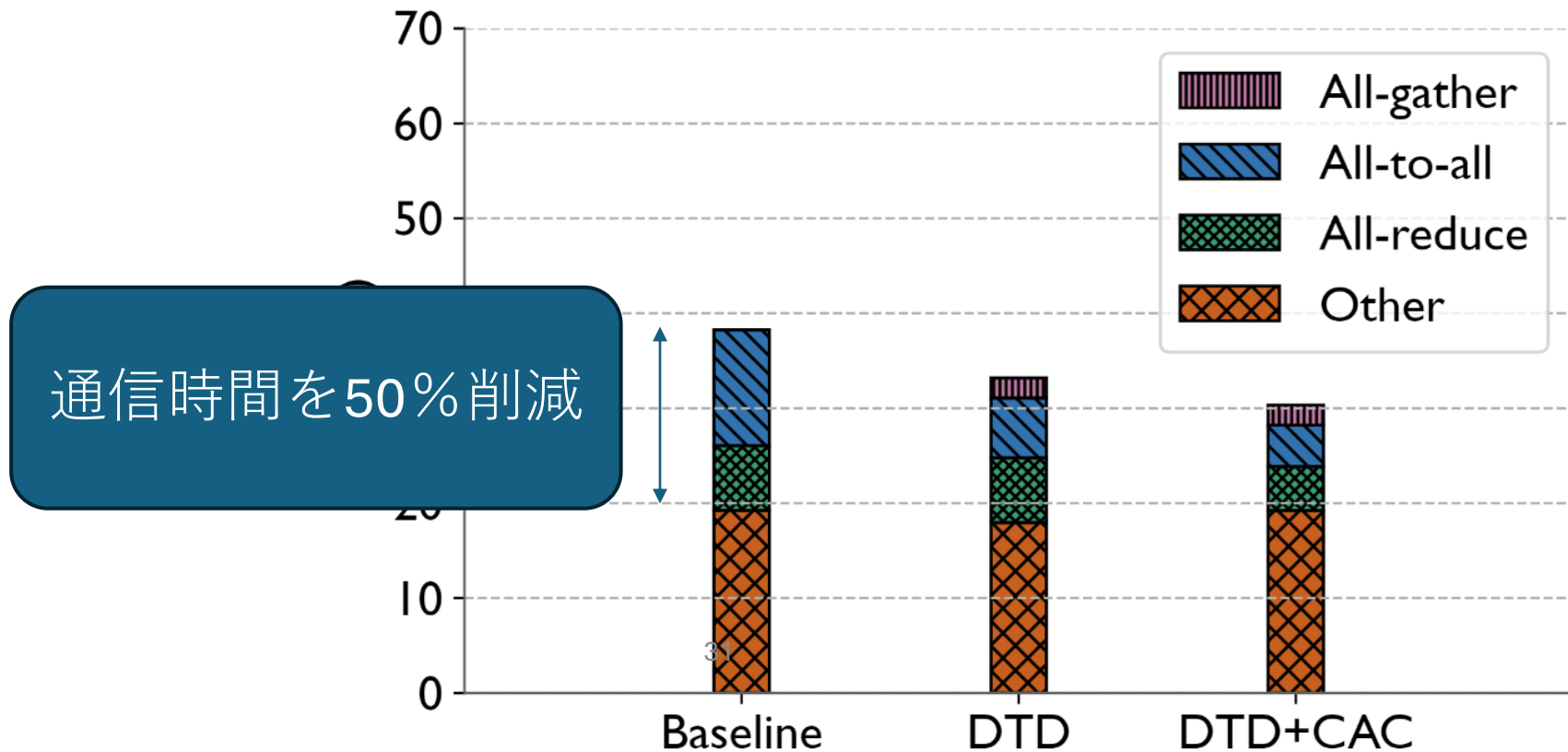
- 異なる並列化手法との組み合わせにより、通信コストを削減
 - **T** – Tensor Parallelism (Megatron-LM [3])
 - **E** – Expert Parallelism (DeepSpeed-MoE [4])
 - **D** – Sharded Data Parallelism (ZeRO [5])

さらに大きなモデルを訓練可能に



- エキスパート数: 128
- テンソル並列はノード内に限定

Results



1 イテレーション当たりの時間
(6.7Bベースモデル、16 experts on 128 GPUs of Summit)

DeepSpeed-FastGen: High-throughput Text Generation for
LLMs via MII and DeepSpeed-Inference

Connor Holmes, Masahiro Tanaka, Michael Wyatt, Ammar Ahmad Awan, Jeff
Rasley, Samyam Rajbhandari, Reza Yazdani Aminabadi, Heyang Qin, Arash
Bakhtiari, Lev Kurilenko, Yuxiong He

Microsoft DeepSpeed (www.deepspeed.ai)

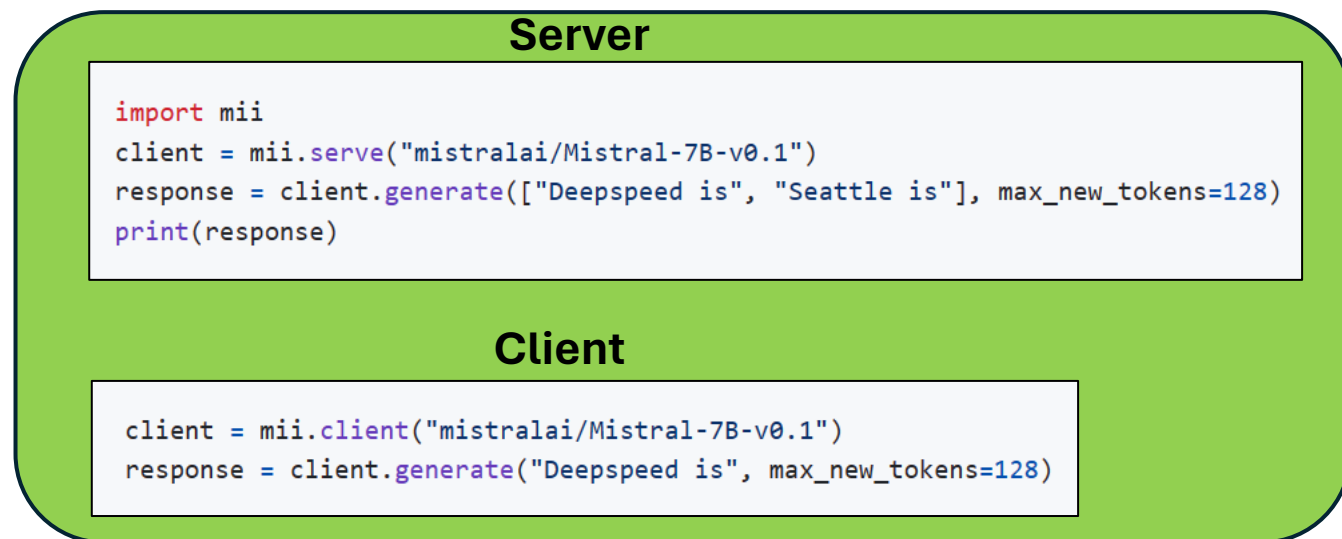
DeepSpeed-FastGen

DeepSpeed-FastGen

- テキスト生成を高速・高効率に実行するためのフレームワーク

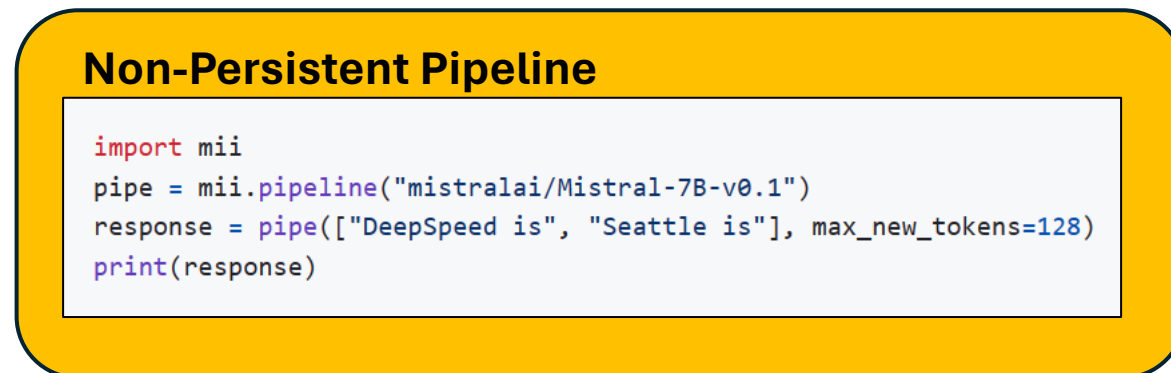
- **Persistent deployment:**

- サーバ-クライアント方式
- 高性能、プロダクション向け



- **Non-persistent deployment:**

- 簡便に実行

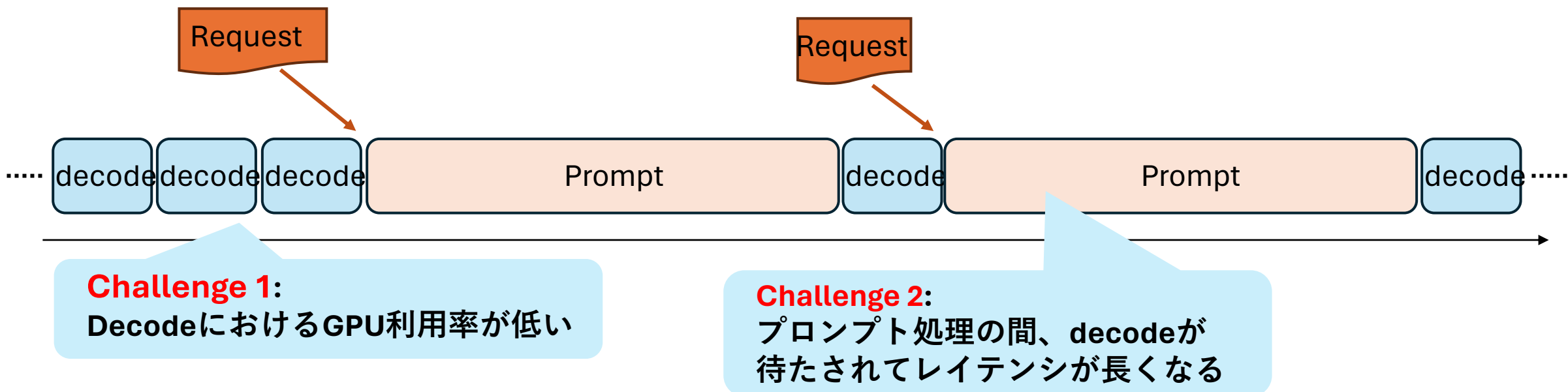


- 特に大規模なLLMの推論は非常にコスト高、効率化は重要

LLM のテキスト生成

- テキスト生成のフェーズ
 - **Prompt processing:** プロンプト全体を一括して処理（リクエストあたり 1 回）
 - **Decode:** トークン 1 件ずつ処理 (1 トークン生成ごとに1回)

Iteration-level Scheduling (vLLM)



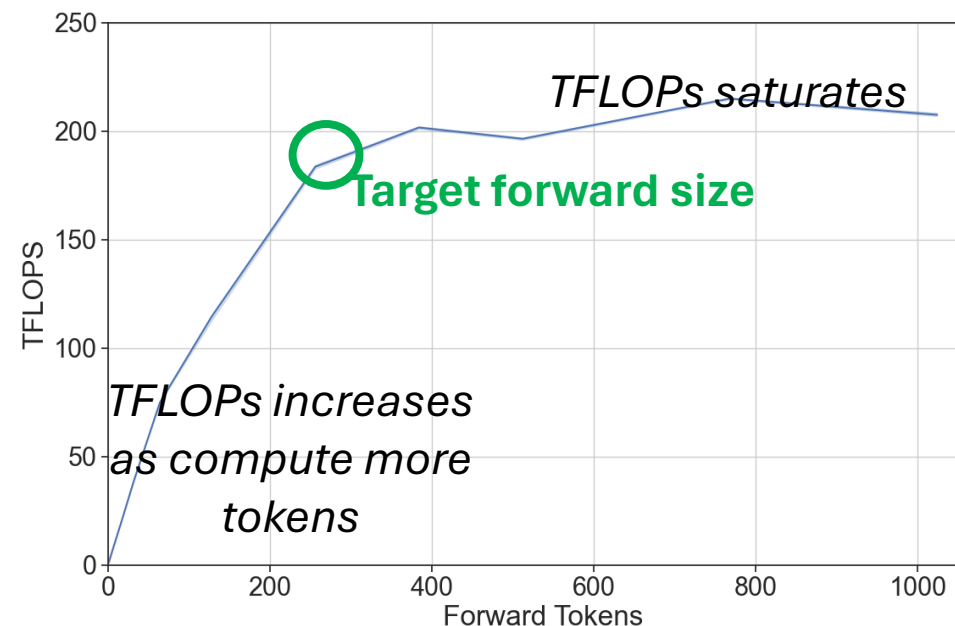
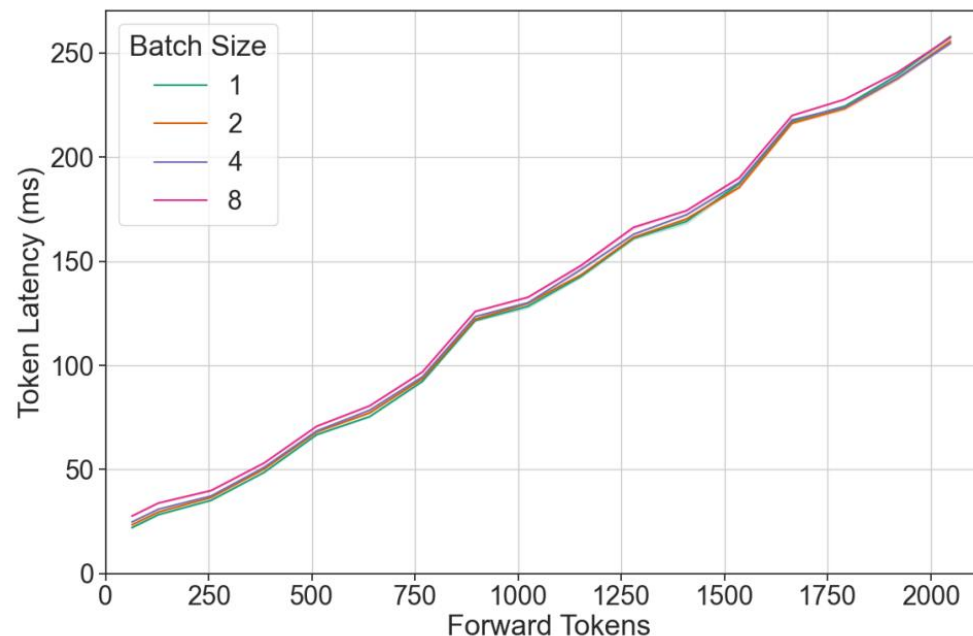
計算負荷の特徴

1. レイテンシーはトークン数で決まる

→長いプロンプトが与えられると、並行して実行されているDecodeが結果を返すのが遅くなる

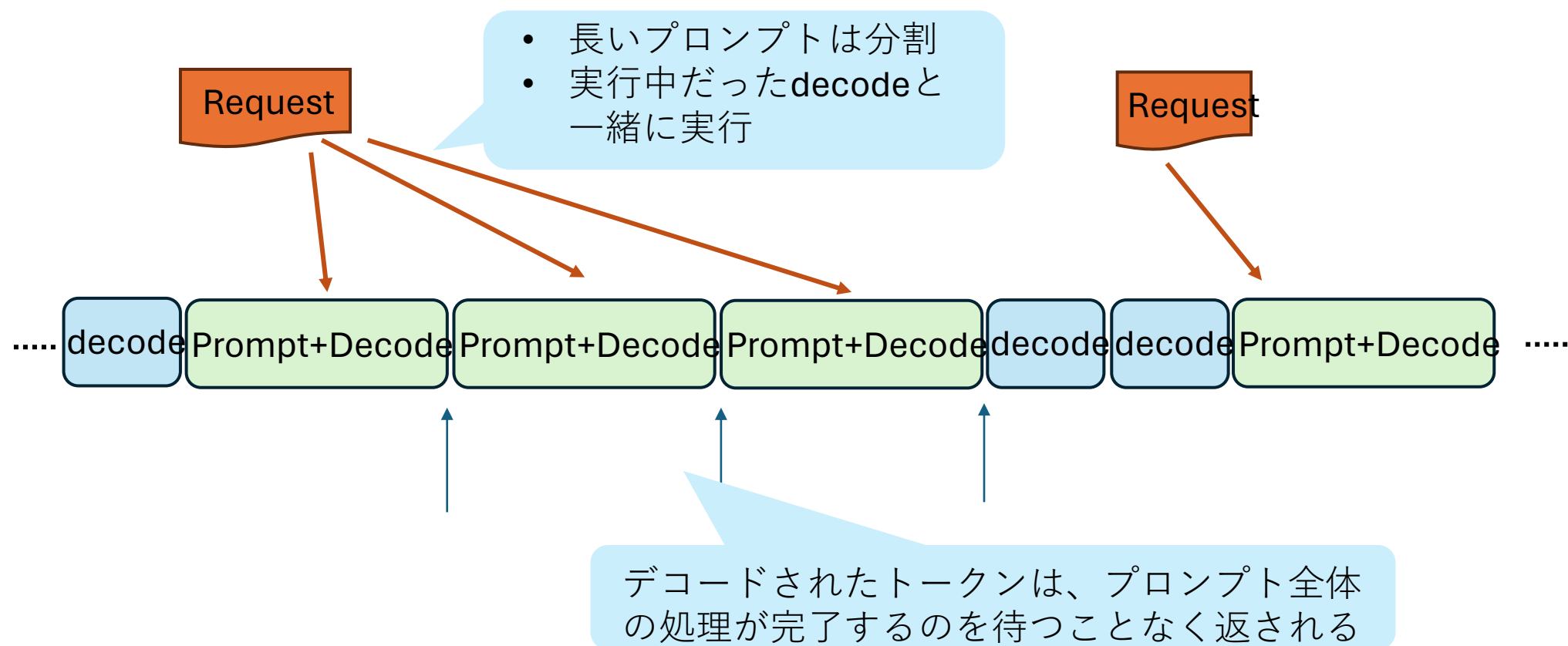
2. GPU実行効率はトークン数を増やすと向上するが、どこかで頭打ちになる

→適切なトークン数で上限を設定できるはず



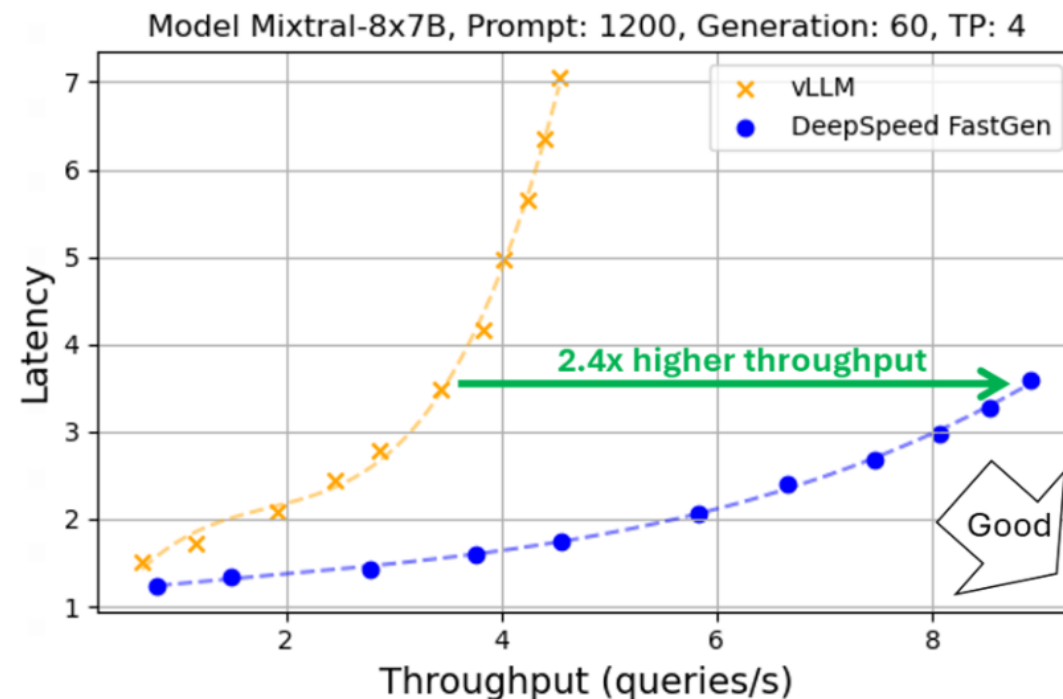
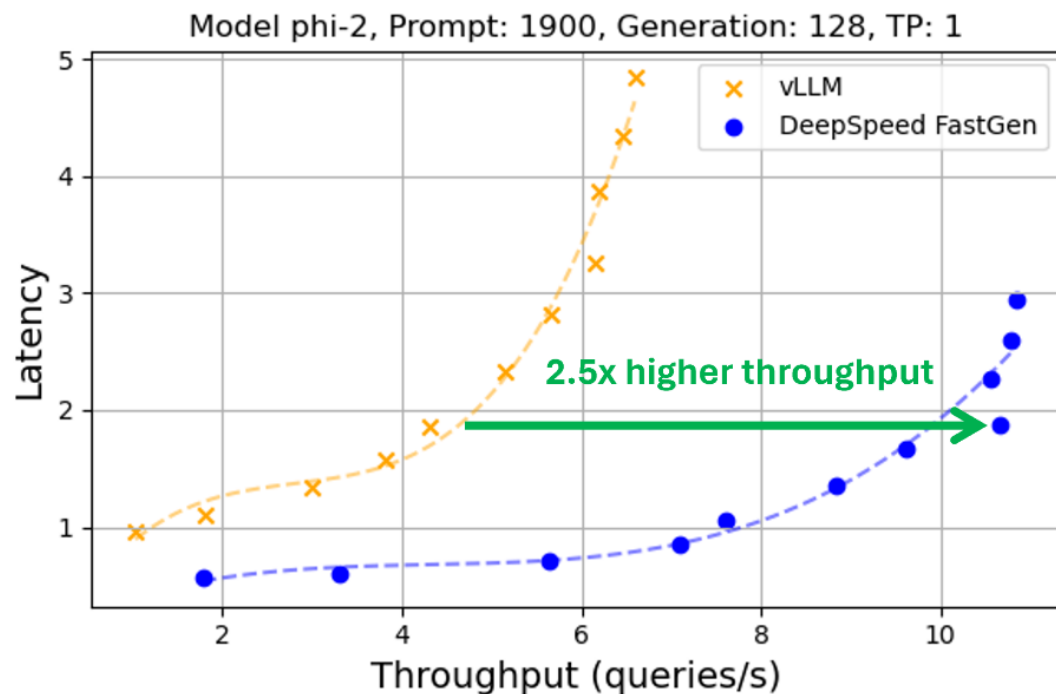
Dynamic SplitFuse: Prompt & Decode Composition

- 一回に計算するトークン数を **target forward size** 呼ぶ一定のサイズで制限
 - 長いプロンプトは分割
 - 短いプロンプトは結合
 - Prompt processingとdecodeを結合



性能評価

- SplitFuse によりスループット・レイテンシが顕著に改善



Phi-2, Llama2, Falcon 等の評価の詳細:

<https://github.com/microsoft/DeepSpeed/blob/master/blogs/deepspeed-fastgen/2024-01-19/README.md>

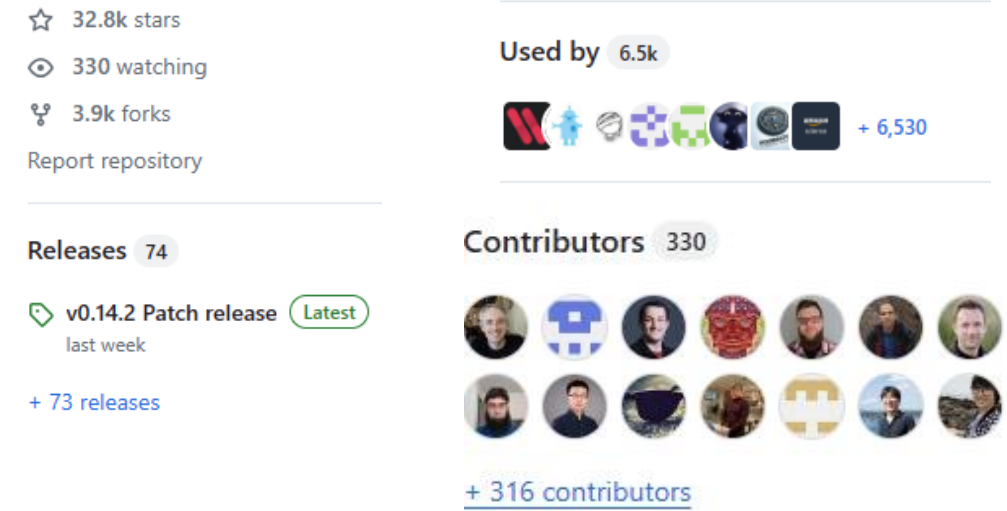
<https://github.com/microsoft/DeepSpeed/tree/master/blogs/deepspeed-fastgen>

進行中のその他のプロジェクト

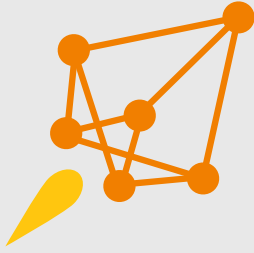
- 長い系列の並列計算
- PyTorchのコンパイル機能との統合
- I/O 高速化
- 多様なアクセラレータのサポート

オープンソース活動へのご参加をお待ちしています！

- 実は非常に小さいチーム (~20 人)
- OSSコミュニティからの貢献が重要



- 様々なチャンネルで情報発信・コミュニケーションをしています
 - 最新情報を知りたい → Twitterアカウント
[DeepSpeed \(@MSFTDeepSpeed\)](#)
[マイクロソフトDeepSpeed \(@MSFTDeepSpeedJP\)](#) (**日本語アカウント**)
 - バグレポート等 → [GitHubのIssues](#)
 - 開発に協力する → [GitHubのPull Request \(PR\)](#)
 - 質問・ディスカッション → [GitHubのDiscussion](#)
- GitHubのご連絡・お問い合わせは、英語もしくは英語 + 日本語でお願いします



deepspeed

www.deepspeed.ai

Follow us on X:
@MSFTDeepSpeed / @MSFTDeepSpeedJP

Thank You!

ZeRO-Infinity in Action