



第111回お試しアカウント付き並列プログラミング講習会
「第1回 ディープラーニング分散学習ハッカソン」
分散学習基礎講座

Kazuhiro Yamasaki, Deep Learning Solution Architect, NVIDIA, 01/24/2019



AGENDA

Data ParallelismとModel Parallelism
分散学習特有の問題

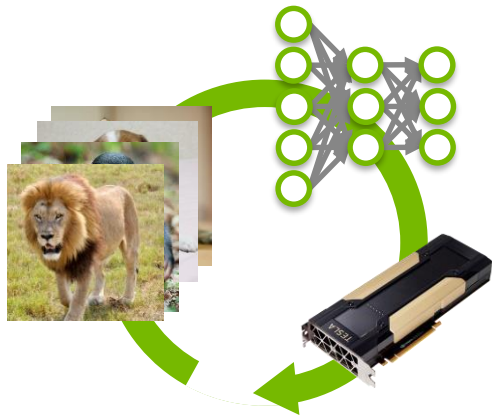
The background features a complex network of thin, light green lines connecting various glowing green nodes of different sizes. The nodes are scattered across the dark blue and black background, creating a sense of interconnectedness and data flow. The overall aesthetic is futuristic and technical.

DATA PARALLELISMと MODEL PARALLELISM

分散学習とは

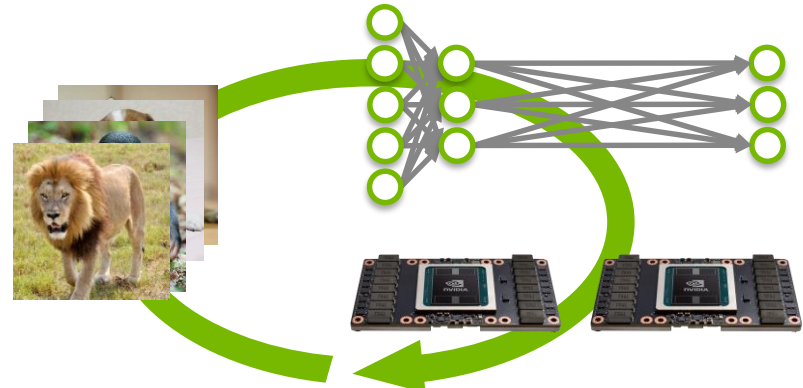
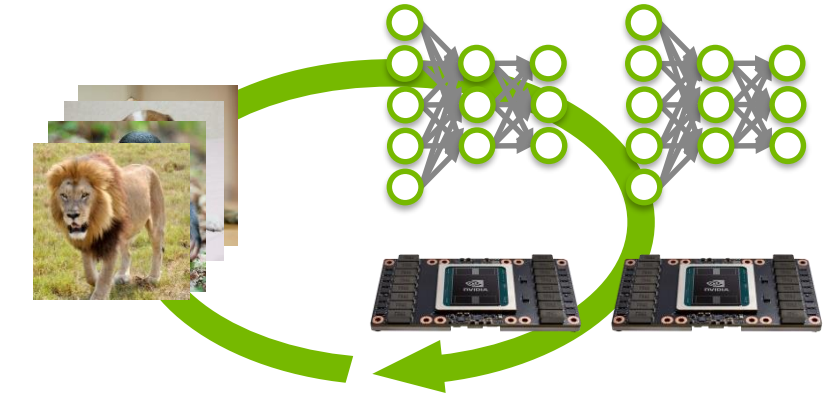
深層学習において必要な計算を、多数の計算機に分散させる

異なるデータで同じモデルを並行
処理し、実行効率を上げる



モデルを分割し、1GPUに収まらない
計算を実行可能にする

Data Parallelism



Model Parallelism

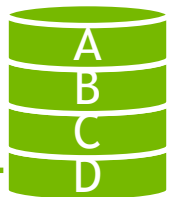
DATA PARALLELISM

考え方

シングルGPUの場合



Forward



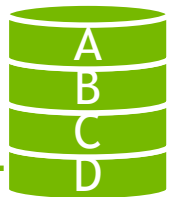
各デバイスに異なるデータを割り当て

Update

データ並列の場合



Forward

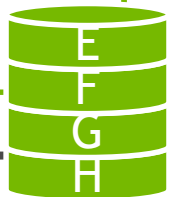


Backward

Update



Forward



Backward

1iteration

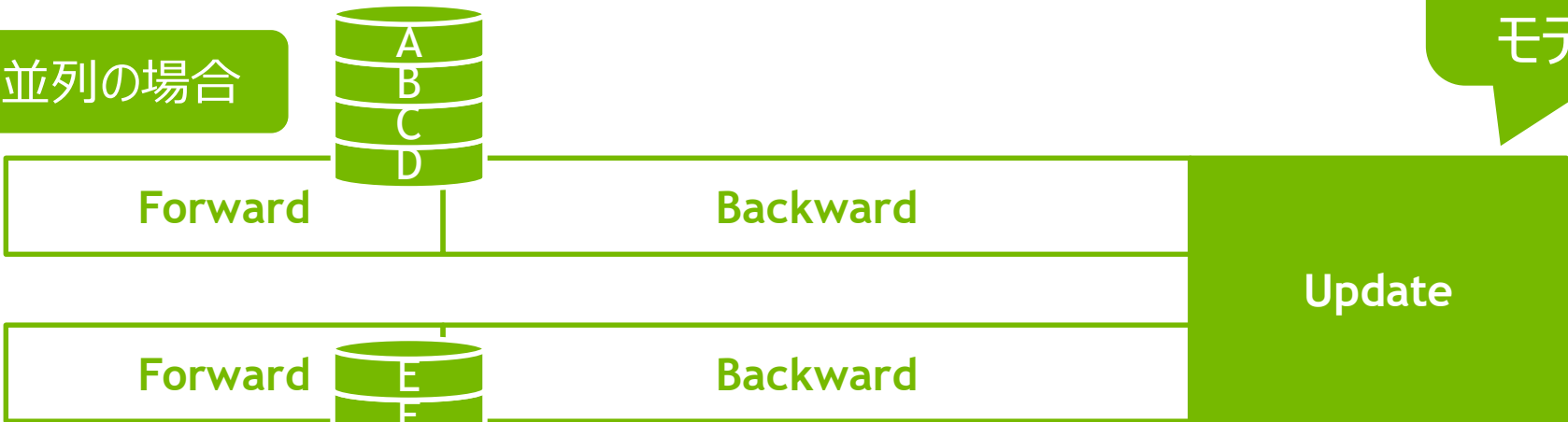
DATA PARALLELISM

考え方

シングルGPUの場合



データ並列の場合



計算した
勾配を共有し
モデル更新



1iteration

DATA PARALLELISM

考え方

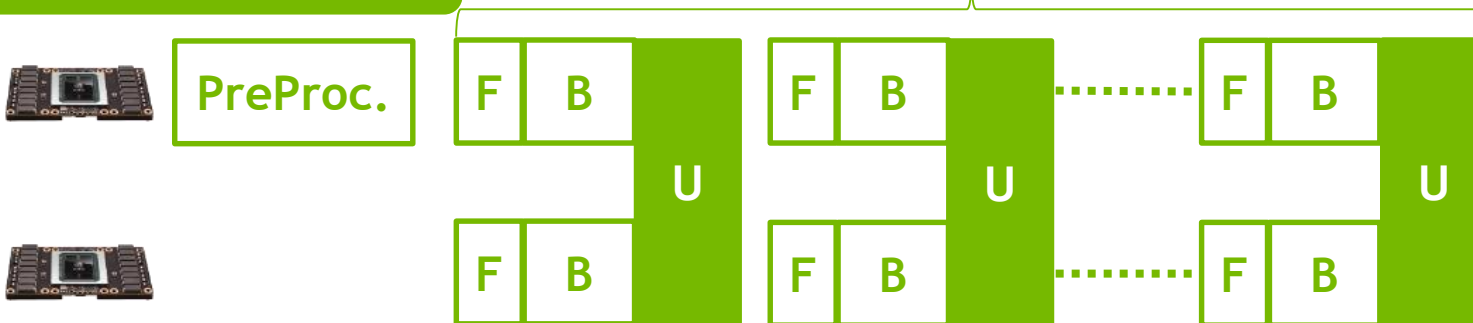
シングルGPUの場合

例: 200 iter.



データ並列の場合

例: 100 iter.



並列化によって
総処理時間が
短くなることを期待

time

DATA PARALLELISM

方式: 同期更新 vs. 非同期更新 & パラメータサーバ vs. p2p

- 同期更新: 全GPUの勾配計算完了を待って、モデルを更新
- 非同期更新: 一つ以上の勾配計算が完了したら、その時点の情報でモデルを更新
- パラメータサーバ: 一つ以上のプロセスでモデルを管理、各GPUから勾配を受け取り更新
- Peer-to-peer: GPU間で直接勾配情報を通信し、モデル更新

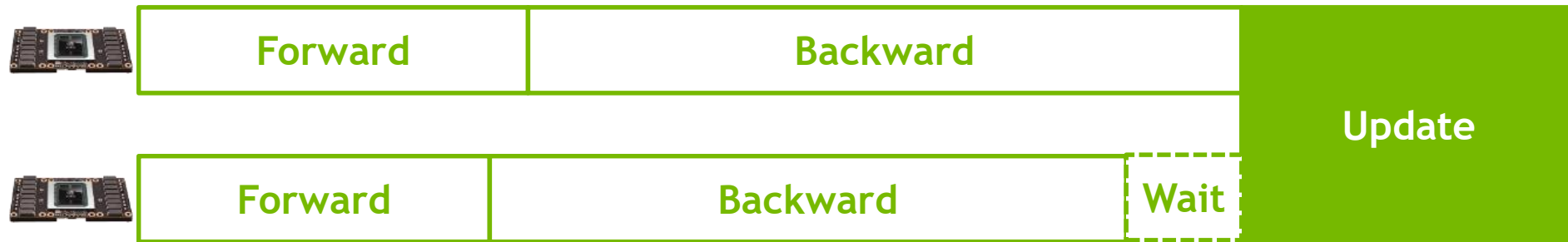


1iteration

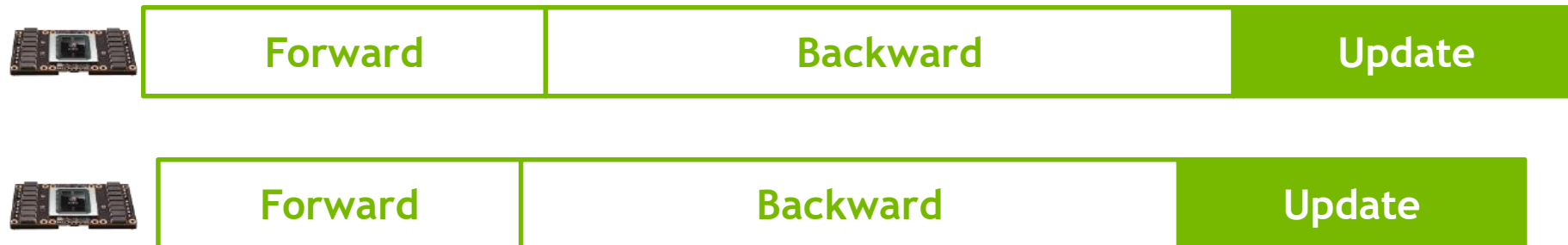
DATA PARALLELISM

方式: 同期更新 vs. 非同期更新 & パラメータサーバ vs. p2p

同期更新のイメージ

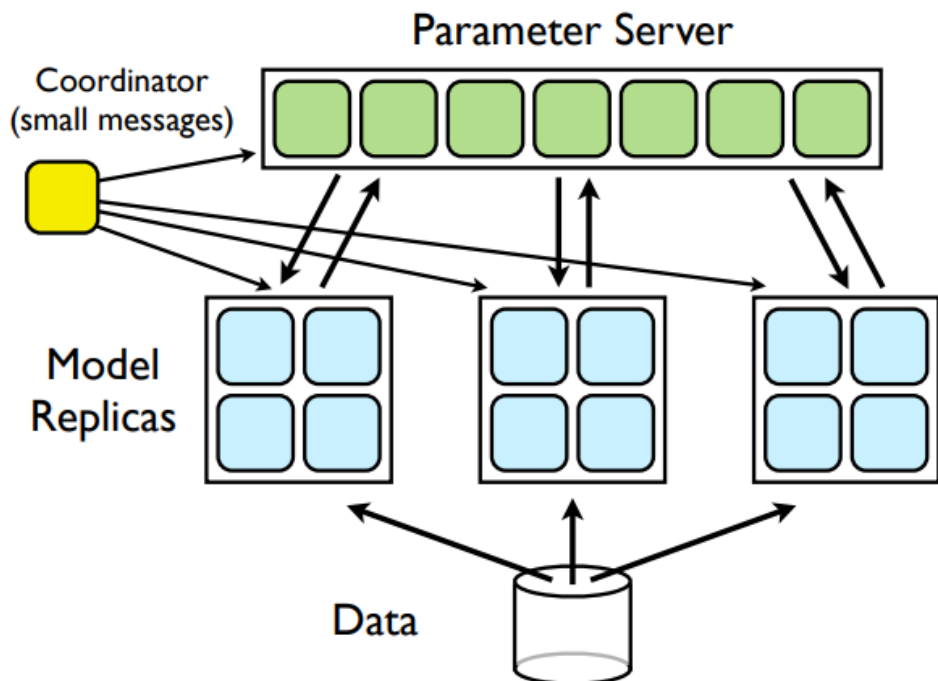


非同期更新のイメージ

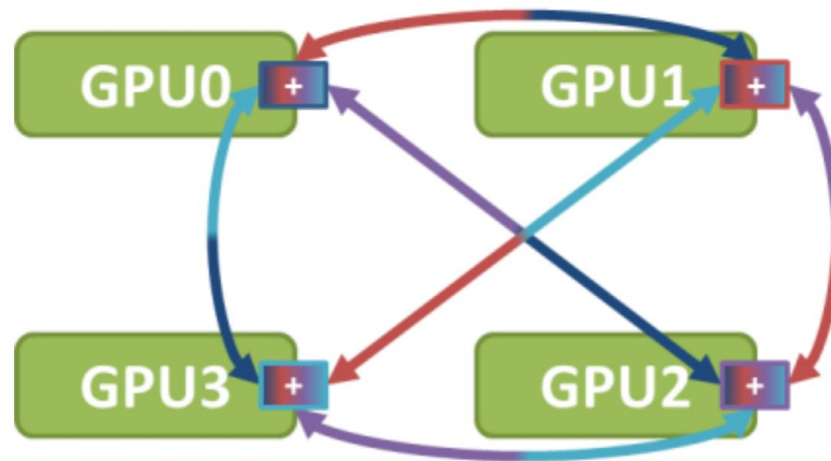


DATA PARALLELISM

方式: 同期更新 vs. 非同期更新 & パラメータサーバ vs. p2p



パラメータサーバのイメージ
(Large Scale Distributed Deep Networks より)

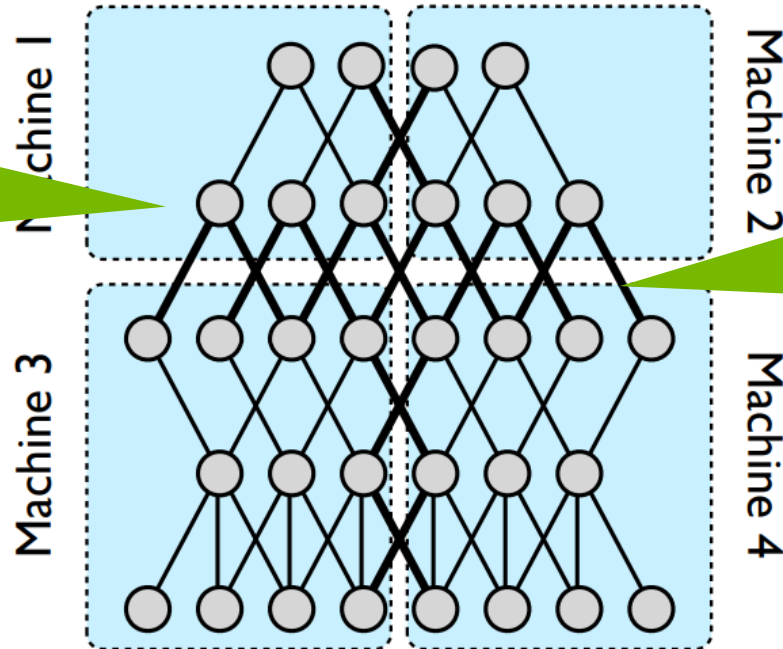


Peer-to-peerでのイメージ

MODEL PARALLELISM

考え方

モデルの各部を
別GPU上にロード

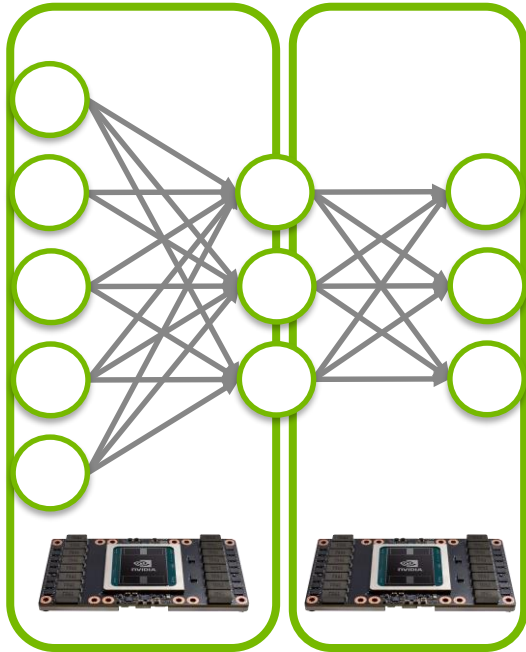


必要に応じてGPU間
通信することで処理を
継続

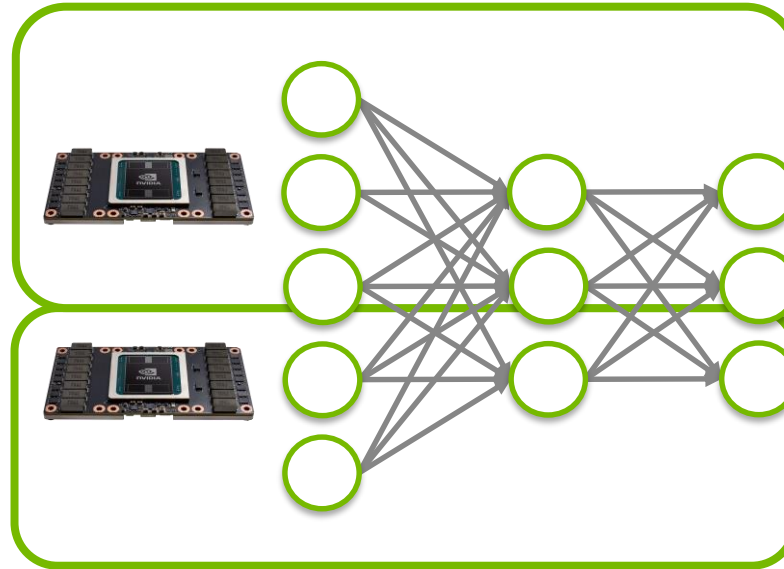
モデル並列のイメージ
(Large Scale Distributed Deep Networks より)

MODEL PARALLELISM

どうモデルを分割するか



各レイヤーを別GPUに割り当て



各GPUで各レイヤーの一部を担当

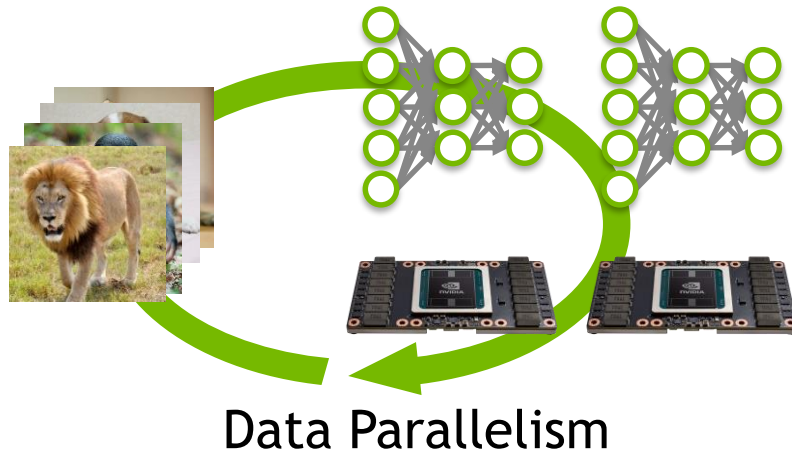
他、畳込みのチャンネルを別GPUに割り当てなどなど.....

A network diagram with green nodes and lines on a dark background. The nodes are represented by small green circles, and the lines are thin green lines connecting the nodes. The background is dark blue/black with some faint, larger green circles.

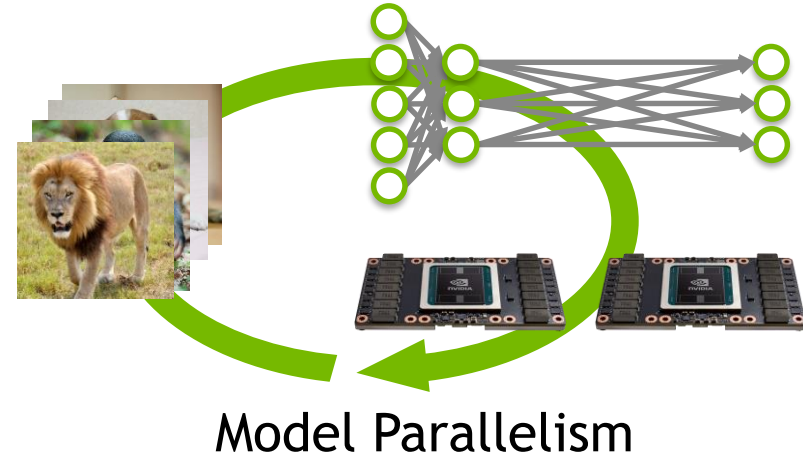
分散学習特有の問題

各問題設定での課題

データ並列とモデル並列



- 大規模化に伴う精度低下
- 非同期更新時の精度低下

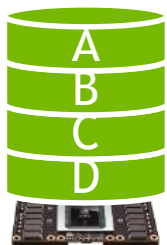


- 通信コストの増大

DATA PARALLELISMでの問題

大規模化に伴う精度低下: large batch size problem

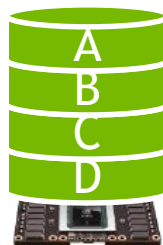
シングルGPUの例



Batchsize=4

GPU0

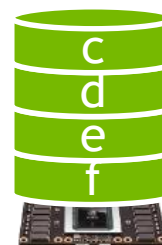
データ並列の例



GPU0



GPU1



GPU7



Batchsize=4x8=32

分散度が上がるほど
トータルバッチサイズが増加

DATA PARALLELISMでの問題

大規模化に伴う精度低下: large batch size problem

- スケーラビリティと精度のトレードオフ

- (正確には同期更新の場合の話)

- バッチサイズ増: モデル精度が低下する

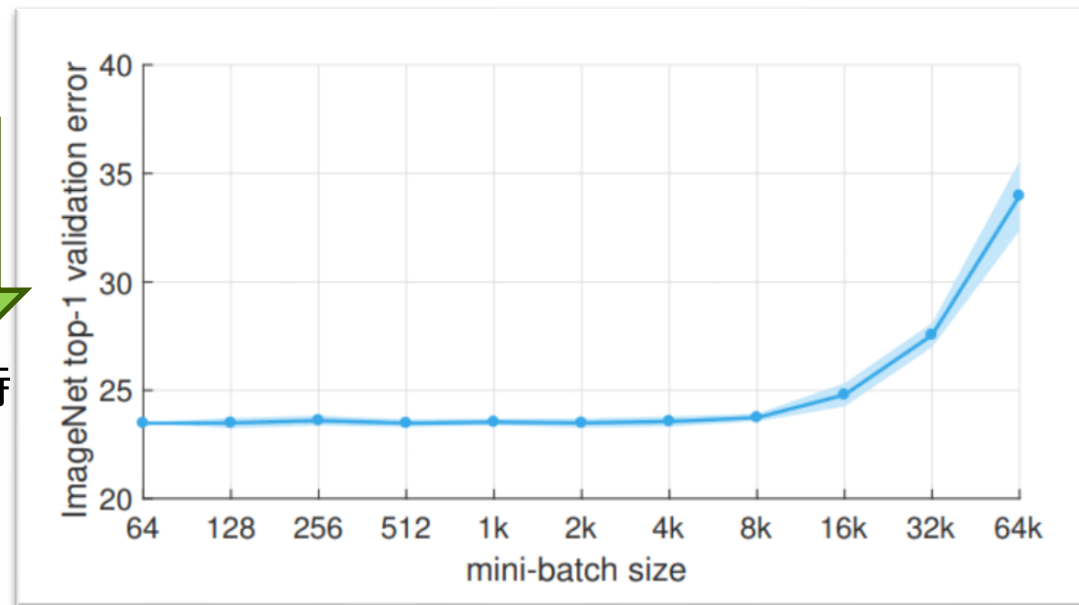
- バッチサイズ増 \equiv ノード数増
- Facebook: 8Kバッチ(256GPU)まで精度維持
- PFN: 32Kバッチ(1024GPU)まで精度維持

- 性能: 勾配交換の時間が増加する

- 1024GPU: 2割以上が通信時間



ImageNet, ResNet50, 32バッチ/GPU



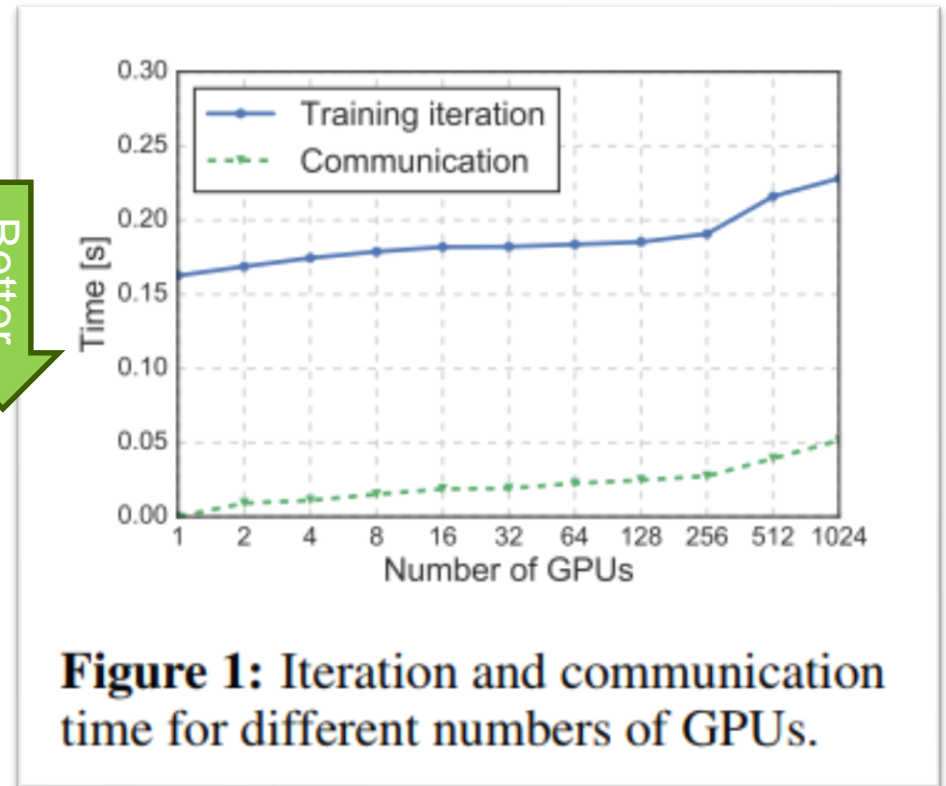
(*) Priya Goyal, et al., Facebook, “Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour”

DATA PARALLELISMでの問題

大規模化に伴う精度低下: large batch size problem

ImageNet, ResNet50, 32バッチ/GPU

- スケーラビリティと精度のトレードオフ
 - (正確には同期更新の場合の話)
- バッチサイズ増: モデル精度が低下する
 - バッチサイズ増 \equiv ノード数増
 - Facebook: 8Kバッチ(256GPU)まで精度維持
 - PFN: 32Kバッチ(1024GPU)まで精度維持
- 性能: 勾配交換の時間が増加する
 - 1024GPU: 2割以上が通信時間



DATA PARALLELISMでの問題

大規模化に伴う精度低下: large batch size problem

- 学習率を下げることに、バッチサイズを増やすことは、(SGD的に)等価

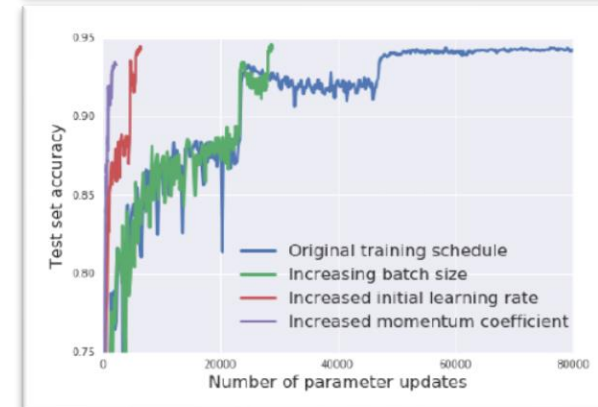
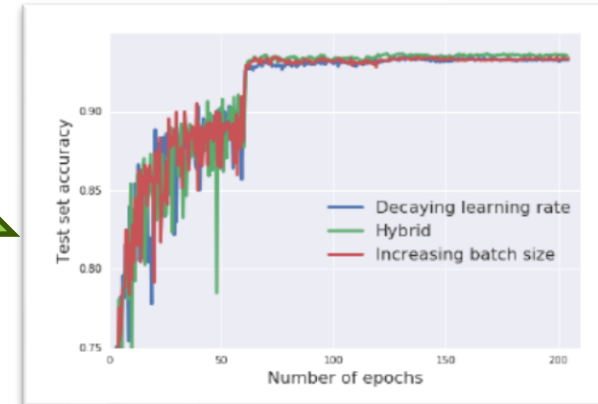
- Noise scale:

$$g = \frac{\epsilon}{1-m} \left(\frac{N}{B} - 1 \right)$$

- ϵ : ラーニングレート
- B: バッチサイズ

- 数十epoch経過後、学習率を下げる代わりに、バッチサイズを増やすことでも精度を保てる

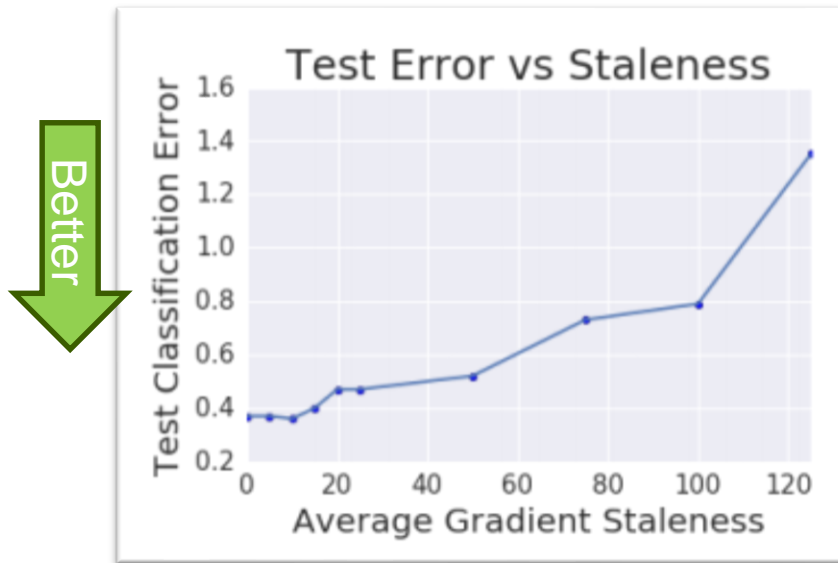
- 分散学習の計算ノードを追加、ということでも可能



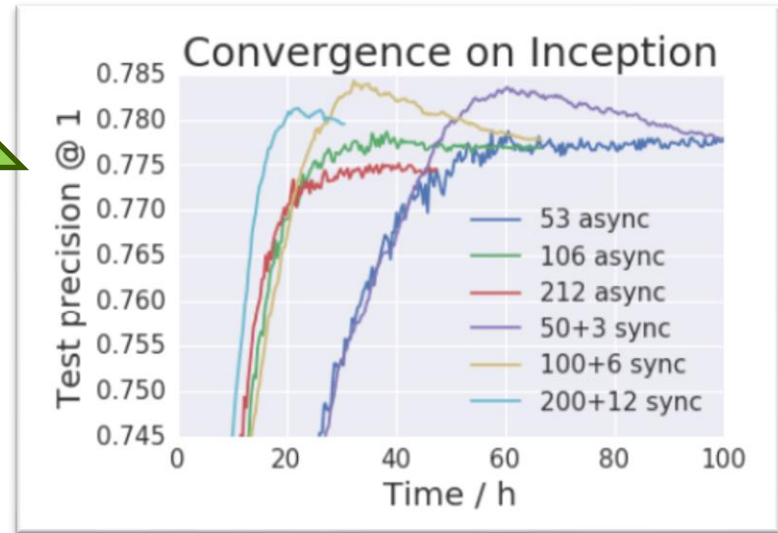
DATA PARALLELISMでの問題

非同期更新時の精度低下: staleness

Staleness: 非同期に計算された勾配が古く、最新モデルが古い勾配で更新されてしまう



Better



対策: ワーカー数を冗長に設定: $N + b$

- N ワーカーから勾配が届いたら、残りの b ワーカーを待たずに、モデル更新

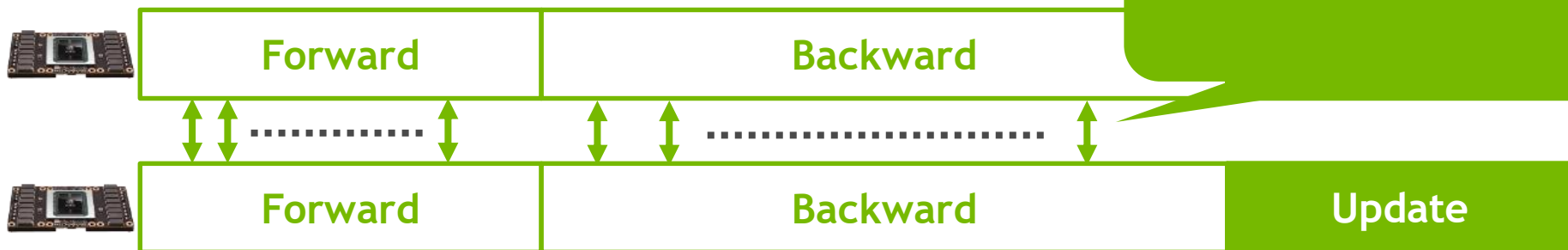
MODEL PARALLELISMでの問題

通信コストの増大

データ並列の場合



モデル並列の場合

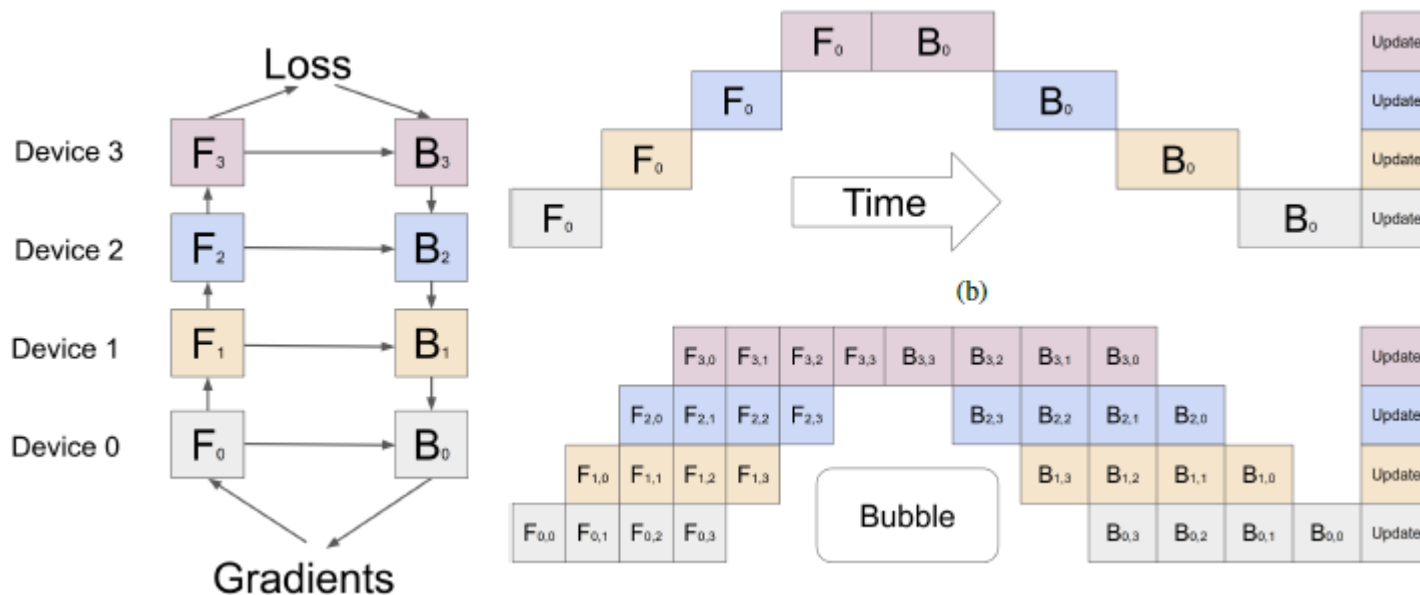


MODEL PARALLELISMでの問題

通信コストの増大

巨大モデルをレイヤーで分割して、別プロセッサに割り当てる

- ミニバッチを分割・パイプライン化することで、プロセッサのアイドル時間を短縮する





まとめ

KEY TAKEAWAYS

分散学習のエッセンス

1. 分散学習にはdata parallelismとmodel parallelismがある
 1. 組み合わせることもある
2. 分散させると、単一GPUで学習していたときとは異なる問題が生じる
 1. 適切な対策を実施する必要あり

