7A-01

分散深層学習を高速化させるFPGA Ring-Allreduceの検討

田仲 顕至† 有川 勇輝† 伊藤 猛† 寺田 和彦†
森田 和孝‡ 三浦 史光‡ 寺本 純司‡ 坂本 健†

日本電信電話株式会社NTT先端集積デバイス研究所†
日本電信電話株式会社NTTソフトウェアイノベーションセンタ‡

1. はじめに

ディープラーニング（DL）は様々なアプリケーションが開発されているが、DLモデルの学習には膨大な計算リソースが必要となり、アプリ開発のボトルネックとなっている。そこで、DLモデルの学習を並列処理する分散深層学習が注目を集めており、中でも、データ並列同期更新型が良好な性能を示している[1, 2]。このアプローチでは、ミニバッチ毎に各ワーカーノードで計算された勾配を毎ステップ集団通信（Allreduce）により共有する必要があり、ボトルネックとなっている。そのため、様々な通信ボトルネックの解消手法が提案されている

2. 既存手法

ボトルネック解消法として、Ring-Allreduce、もしくは、Ring-Allreduceを2次元に拡張した2D-Torus Allreduceが採用する事例が複数報告されている [3, 4]。Ring-Allreduceでは、Reduce処理が行われるノードがネットワークで数珠つなぎに接続される。各ノード内のGPUで計算された勾配情報はネットワークを介して隣接するノードのGPUへ送信されReduce処理が施され、さらに隣接するノードへ送信される。Ring-Allreduceの場合、すべてのノードでのReduce処理後にはRing-Allreduce開始ノードに合計された勾配データが到着する。開始ノードはこの合計勾配データをRingネットワークにてBroadcatすることで、全ノードで勾配データを共有する。2D-Torus Allreduceも同様の手順にて実行されるが、多くの実装方法が提案されているため、ここでは説明を割愛する。

この方法では、各ノード内のGPUの内部メモリまでデータが送られてしまいデータ移動のコストが多分に生じる。また、各GPUは計算処理（Reduce）と通信処理を交互に実行する必要があり、大きなオーバーヘッドが発生する。加えて、計算処理と通信処理が交互に行われるので、待ち時間の多い手法である。これらの要因から従来のRing-Allreduceを用いた分散深層学習では、通信の高速化のために通信精度を落とすなど、アルゴリズムの可変を余儀なくされていた。

Distributed deep Learning acceleration with FPGA Ring-Allreduce
† NTT Device Technology Laboratories, NTT Corporation
‡ NTT Software Innovation Center, NTT Corporation

3. 提案手法

我々はRing-AllreduceをFPGA NICにオフロードすることを提案する。各GPUは勾配データをFPGAに送信し、その後の、勾配データの計算処理（Reduce）と通信処理をFPGA NICで行う。このような形態をとることで、データ移動のコストは低減し、また、GPUをDL計算処理に集中させることでオーバーヘッドも低減できる。提案するFPGA内部アーキテクチャを図1に示す。

更に、Ring-AllreduceをFPGA NICにオフロードすることによって、勾配計算とAllreduceを非同期に実行することが可能となる。提案手法では、GPUが誤差逆伝播を各パラメータの勾配が出力され次第、順次FPGA NICへの転送を開始する。ここで、転送はFPGA NICのDirect Memory Access Controller（DMAC）によって実行されるため、GPUは勾配計算を止めること無くDMAを実行でき、また、Allreduceを開始することもできる。加えて、Allreduce終了後のデータにおいてもFPGA NICのDMACがGPU内部メモリにデータを書き込むため、GPUとFPGA-NICは非同期に計算とAllreduceを実行することができる。我々の提案するパラメータ毎の計算と通信のオーバーラップ（PCCO, Parameter based Computing/Communication Overlap）の実行手順を図2に示す。

4. 評価

提案手法の評価のために、我々はCPU（Intel, Core i7 5930K）、memory（32 GB）、GPU（Nvidia, Tesla P100）、HCA（Mellanox, ConnectX-4 HCA）、IB Switch（Mellanox, Switch IB-2）、InfinibandEDR（MCP1600）、FPGA（Xilinx: VCU118）、DMA Controller（Xilinx, XDMA）、EhternetMAC（Xilinx, CMAC）、100G Ethernet（100GBASE-SR4）を用いて、4ノード各1GPU、1FPGA、1HCAの分散深層学習システムを構築した。

図1

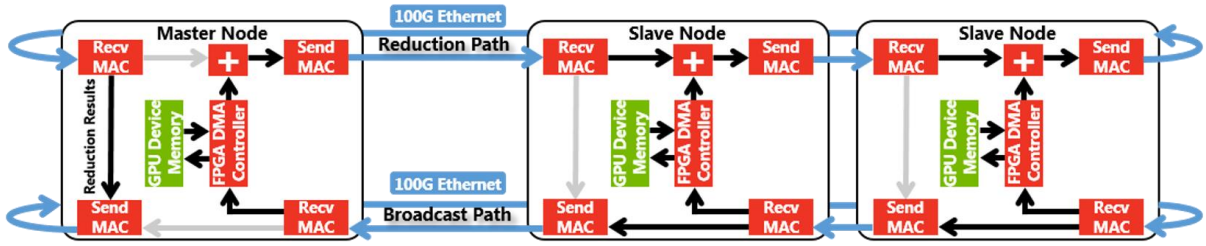

図2

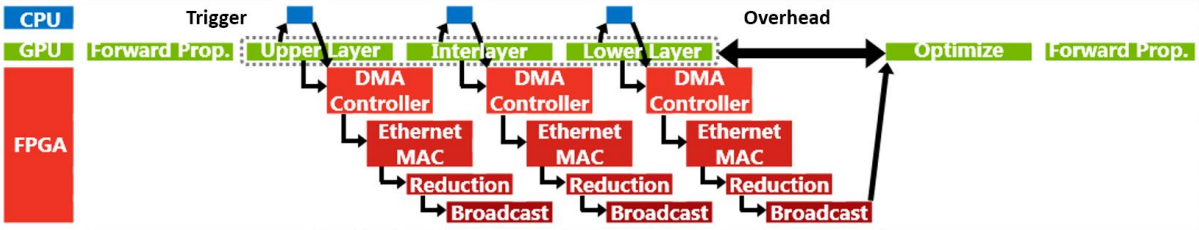

図3

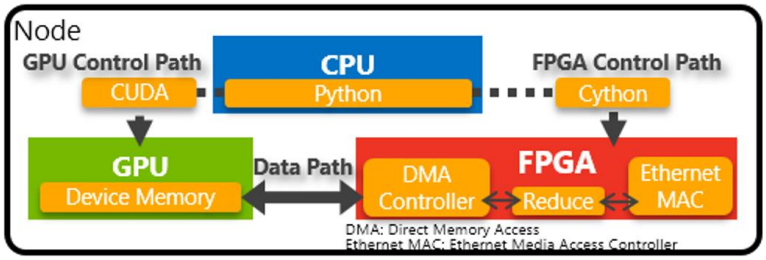

図4

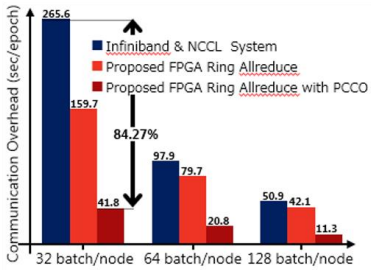

図 1: 作成した FPGA Ring-Allreduce の回路とデータ移動の模式図。図 2: 勾配計算と集団通信を極力オーバーラップさせた提案手法（PCCO）のタイムチャート。図 3: 今回の性能評価に用いた 1 ノードの構成とソフトウェアの役割。図 4 : Infiniband を用いたシステム、FPGA Ring-Allreduce を採用したシステム、FPGA Ring-Allreduce と PCCO を採用したシステムの集団通信オーバーヘッドを様々なバッチサイズで比較した。

システムの構成を図3に示す。このシステムを用いて、従来のInfiniband、GPU Direct RDMAを用いた分散深層学習と、提案するシステムを用いた分散深層学習で、学習精度、Allreduce時間、学習全体の時間で比較した。

学習精度に関わるアルゴリズムの変更は施していないので、同程度の学習精度が確認された。集団通信のオーバーヘッドは分散深層学習でよく使用される 32 bach/node で 84.27 %削減できた。その他のバッチサイズであっても大幅に削減できることが示された（図 4）。全体の学習時間としては 7 %の高速化が実現できた。

5. まとめと今後の課題

本研究は分散深層学習のボトルネックである集団通信時のデータ移動が最小となるようなFPGA Allreduce を提案し、また、そのアーキテクチャに適した分散深層学習スケジューリングに関しても提案した。その結果、学習精度の劣化無く、大幅な高速化に成功した。今後は、ノード内の計算リソース・ノード数を増大させた場合のスケーラビリティに関して調査する。

参考文献

[1] Tal Ben-Num, and Torsten Hoefler: Demystifying Parallel and Distributed Deep Learning: An In-Depth Concurrency Analysis, arXiv:cs.LG/1802.09941, (2018).

[2] Takuya Akiba, Shuji Suzuki, and Keisuke Fukuda: Extremely Large Minibatch SGD: Training ResNet-50 on ImageNet in 15 Minutes, Deep Learning at Supercomputer Scale (NIPS’17 Workshop), arXiv:cs.DC/1711.04325, (2017).

[3] Xianyan Jia, Shutao Song, Wei He, Yangzihao Wang, Haidong Rong, Feihu Zhou, Liqiang Xie, Zhenyu Guo, Yuanzhou Yang, Liwei Yu, Tiegang Chen, Guangxiao Hu, Shaohuai Shi, Xiaowen Chu: Highly Scalable Deep Learning Training System with Mixed-Precision: Training ImageNet in Four Minutes, Workshop on Systems for ML and Open Source Software at NeurIPS 2018, arXiv:cs.CV/1807.11205, (2018).

[4] Chris Ying, Sameer Kumar, Dehao Chen, Tao Wang, Youlong Cheng: Image Classification at Supercomputer Scale, , Workshop on Systems for ML and Open Source Software at NeurIPS 2018, arXiv:cs.CV/1811.06992, (2018).