5L-08

ネットワーク経由GPU接続手法の特徴を考慮したGPU割り当て

岡崎　真博†　　森島　信†　　松谷　宏紀†

慶應義塾大学大学院　理工学研究科　〒223-8522　神奈川県横浜市港北区日吉　3-14-1†

Email: †{okazaki,morishima,matutani}@arc.ics.keio.ac.jp

1. 概要

GPU（Graphics Processing Unit）は、従来コンピュータゲームなどの動画像処理に特化して活用されてきた。近年は、GPUの並列計算能力の高さに注目したディープラーニングといった機械学習等、GPGPUとしても広く利用されている。

しかし、GPUを活用する際に従来のPCI Express（PCIe）に直接接続する手法では、使用できるGPUの数が計算機のPCIeスロットの数に制限されてしまう、1つの計算機が1つのGPUを占有するので、GPUリソースを余らせてしまうといった問題点がある。Ethernetネットワーク経由でGPUと接続するリモートGPU環境を用いることで上記の問題点を解決できるが、GPUを直接計算機に接続する場合と比べて、リモートGPUとの通信オーバヘッドを考慮する必要がある。

リモートGPUを実現するには、後述する通り、PCIe over 10GbEによる手法、クライアントサーバモデルによる手法がある。両者はCPU-GPU間の転送サイズが小さい場合の転送オーバヘッド等に差異があり、GPUのリモート化のオーバヘッドを極力減らすために各手法の特徴を考慮した使い分けが必要である。本論文はそのための方法を提案する。

2. 関連研究

Ethernet経由でリモートマシンのGPUを利用する手法としてクライアントサーバモデルによる手法、PCIe over 10GbEによる手法がある。

前者では、GPUを有するサーバマシンに対してGPUを利用したいクライアントマシンがネットワーク経由でサーバに接続し、サーバが有するGPUをネットワーク越しに利用する。前者の代表例としてrCUDA [1]がある。

後者では、PCIeパケットをEthernetフレームとしてカプセル化し、Ethernet上を転送する。リモートGPUとクライアントマシンが直接通信する。後者の代表例としてExpEther [2]がある。

GPU Assignment Based on Characteristics of Network-Attached GPU Methods

†Graduate School of Science and Technology, Keio University, Japan

3. 2つのリモートGPU手法の使い分け方法

3.1. 2つのリモートGPU手法の比較考察

PCIe over 10GbEによるリモートGPUでは、PCIeパケットがEthernetフレームとして送信される。一方、クライアントサーバモデルによるリモートGPUでは、クライアントとGPUの間にサーバプログラムが介在する。このため、通信遅延に関しては、PCIe over 10GbEによるリモートGPUが有利になりやすい。

クライアントサーバモデルによる手法では、リモートGPUの利用に際し、サーバプログラムのオーバヘッドが生じる。このため、スループットに関しては、CPU-GPU間の転送サイズが小さく転送回数が多いときにPCIe over 10GbEが有利になりやすい。一方、一度に送る転送量が増えるにしたがい両者の帯域はネットワーク帯域の上限に近づき、両者の差異は小さくなる。

上記のようにPCIe over 10GbEによる方法は通信オーバヘッドが小さいという利点があるが、PCIe over 10GbEのための専用カードが装着された計算機からの利用となる。一方で、クライアントサーバによる方法では、クライアント側のハードウェア環境に依存せずに、クライアント数を容易に拡張できるという利点がある。

3.2. システムモデル

上記2つのリモートGPU手法の使い分けを検討するため、ここでは、単純化された以下の仮定を行う。

- n台のクライアントマシンがあり、各マシンではGPUを利用するアプリケーションがm個ずつ動作する。
- n*m個のアプリケーションは同じ帯域でGPUと通信を行うものとするが、その平均転送サイズはアプリケーション毎に異なるものとする。アプリケーションiの単位時間毎の転送回数をPiとする。
- n台のマシンは、PCIe over 10GbEによる方法とクライアントサーバモデルによる方法の両方に対応しており、どちらか片方を選択できる。
- リモートから利用可能なGPUの数をgとする。g≦nとする。

なお、クライアントサーバモデルによる方法

では、単一 GPU を異なるクライアントマシン上で動作するアプリケーションで共有できる。

3.3. リモート GPU 手法の使い分け方針

上記のシステムモデルにおいては、直感的に以下の傾向が予想される。

- 転送回数 Pi が大きなアプリケーション（転送サイズが小さいアプリケーション）が集中してるクライアントマシンにおいては、PCIe over 10GbE を用いて、クライアントとリモート GPU の間に専用リンクを張ると、サーバプログラムによるオーバヘッドを軽減できる。
- 転送回数 Pi が小さなアプリケーション（転送サイズが大きい）を、クライアントサーバモデルによる手法で単一 GPU に複数収容してもスループットの劣化は目立たない。

上記の割り当て手法を実現するには、単位時間当たりの転送回数 P の値に応じて、PCIe over 10GbE による手法とクライアントサーバモデルによる手法でどの程度スループットに差が生じるかを調査する必要がある。

本論文では rCUDA と 10GbE 版の ExpEther の 2 つの手法について、4GB のデータを転送したときのスループットを実機測定した。具体的には、転送サイズ（つまり転送回数 P）を 4B から 40MB まで変化させながらスループット T を実測した。紙面の都合から測定結果は省略するが、P が大きいとき（転送サイズが小さいとき）は PCIe over 10GbE のスループット T が高くなった。

3.4. GPU 割り当て手法

GPU 数 g がクライアントマシン数 n と同じとき、すべてのクライアントマシンは PCIe over 10GbE 手法でリモート GPU を使用する。

GPU 数がクライアントマシン数 n より小さいとき、i 個のリモート GPU を PCIe over 10GbE 用として使い、g-i 個のリモート GPU をクライアントサーバモデル用として使うものとする。前者を Gi 群、後者を Gg-i 群と呼ぶ。ここで、

- 転送回数 P の合計値が多い i 個のクライアントマシンはそれぞれ Gi 群の GPU を 1 つ占有する。
- 残った n-i 個のクライアントマシンは、単一 GPU の負荷が均一になるように Gg-i 群の GPU を共有する。

上記を i の値を 0 から g まで変化させながらトータルスループットが最大化される i の値を探索する。

4. シミュレーションによる評価

提案した GPU 割り当て手法の動作を確認するために簡易的なシミュレータプログラムを作成した。ここでは転送サイズは 4B、40B、400B、4kB、40kB、400kB、4MB、40MB からランダムに決まるものとし、アプリケーション毎の転送回数 P を計算した。クライアントマシン数 n=16、アプリケーション数 m=4 とし、リモート GPU 数 g の値を 4 から 16 まで変化させた。PCIe over 10GbE、および、クライアントサーバモデルにおける P ごとの転送スループット T の値は、3.3 節で述べた実測値を使用した。

提案手法の比較相手は、i 個のクライアントマシンをランダムに選び、Gi 群の GPU を 1 つ占有させた場合とする。g 個すべての GPU をクライアントサーバモデルによる手法で用いた場合の性能を 1 とし、このうち i 個の GPU は PCIe over 10GbE による手法に切り替えたとして、提案手法のスループット向上率/比較相手のスループット向上率を下図に示す。g の数が大きくなるにしたがい両者の差は小さくなっているが、少数の g に対しては提案手法によって注意深く割り当てたほうがスループットの利得が大きくなった。

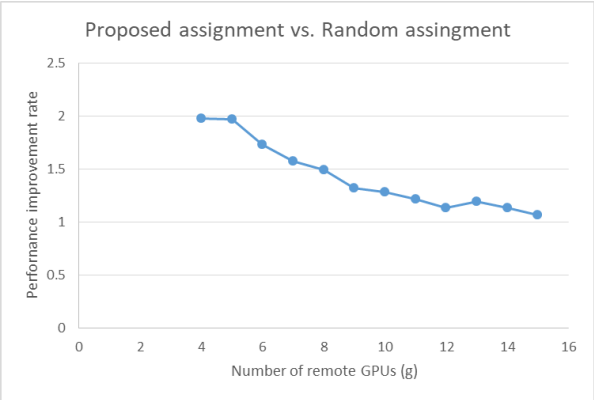

図 1. リモート GPU の数(g)を変化させたときの提案手法とランダムな割り当てを比較した性能向上率

5. 結論

リモート GPU を実現するには、PCIe over 10GbE による手法、クライアントサーバモデルによる手法がある。両者は転送サイズが小さい場合の転送オーバヘッド等に差異があり、GPU のリモート化のオーバヘッドを極力減らすために 2 つの手法の特徴を考慮した使い分けが必要である。本論文はそのための方法を提案した。

6. 参考文献

[1] J. Duato, et al., “rCUDA: Reducing the Number of GPU-based Accelerators in High Performance Clusters”, Proc. of HPCS 2010.

[2] J. Suzuki, et al., “ExpressEther - Ethernet-Based Virtualization Technology for Reconfigurable Hardware Platform”, Proc. of Hot Interconnect 2006.