7G-06

ExpEtherにおけるパケット圧縮手法の提案

志村英樹† 三石拓司‡ 天野英晴†
†慶應義塾大学 理工学部 情報工学科 ‡慶應義塾大学大学院理工学研究科

1 はじめに

近年 PCIe を利用するデバイスが増え、PCIe の拡張のため ExpEther[1] が開発された。ExpEther は、PCIe を拡張することを目的とした Ethernet を基盤とした仮想化技術である。この ExpEther は、Ethernet を利用しているため、PCIe と比較してバンド幅が小さい。そこで、送信パケットを圧縮した後データを転送し、転送先でデータを伸長することでデータ転送時間を削減する。よって、圧縮によりバンド幅を改善できるという提案である。本研究では、WAH という bitmap 形式のデータ圧縮の圧縮に利用される手法を利用して圧縮を行い、圧縮無の ExpEther の転送時間と圧縮有の転送時間を比較し評価を行う。

2 想定システム:GPU-Box

ExpEther により複数 GPU を 1 つのホストに接続したシステムで、高速化の要求に合わせて GPU 台数を増減することができる。本研究では GPU-BOX[3] に搭載される ExpEther に提案する圧縮機構を組み込むことを想定する。

3 ExpEther への圧縮機構の提案

ExpEther の通信バンド幅を向上させるために、データを圧縮しデータサイズを小さくすることでデータ転送時間を減らすことを提案とする。

WAH（Word-Aligned Hybrid）[2] は、31bit を一つのデータ長として bit 列を見ていく圧縮方式である。図 1 の WAH の圧縮例参照。

31bit がすべて 0 か 1 の場合はその出現回数を保持する。連続で同じパターンが出てきた場合保持している回数をインクリメントし、違うパターンが出現した場合は、新しいパターンの記録を 0 から開始する。31bit すべて 0 の場合、先頭 2bit を 00 として、残りの 30bit を連続して 31bit 全て 0 の場合をカウントする。31bit 全て 1 の場合は先頭 2bit を 01 として残り 30bit をカウント bit として扱う。合計 32bit を圧縮後の形式として扱う。31bit が 0 と 1 が混ざり合っている bit 列の場合は、先頭に 1 を付けた合計 32bit を圧縮した bit 列として格納する。

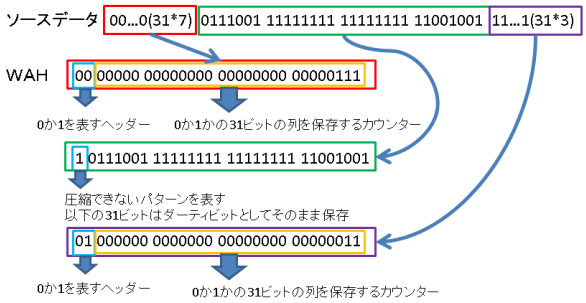

図 1: WAH の圧縮例

4 圧縮機構の実装

実装は、Xilinx Zynq-7000 を使用し Vivado HLS という高位合成ツールを用い IP コアを作成し、Vivado を使い実装を行った。シミュレーションはデータを DDR から読み出し、圧縮、伸長したデータを DDR へ書き込むことでシミュレーションを行う。図 2 は ExpEther の簡略図で、PCIe と ExpEther の間で PCIe パケットからのデータ部分の圧縮を行うように実装することを想定する。

WAH の圧縮は 31bit 毎、データの処理を行うので、convert31to32 モジュールで 32bit のデータを 31bit に変換する処理を行い、その後 wah_compress モジュールで圧縮を行う。対して、伸長は wah_uncompress モジュールでデータ伸長後、31bit で出力されたデータを 32bit に変換する convert31to32 モジュールで処理を行う。モジュールへのデータ入力出力は図 3 のように FIFO IP コアがストリーム処理を行う。FIFO IP コアは Vivado によって生成されたコアで、深さ 1024 の AXI Stream インターフェスのコアである。

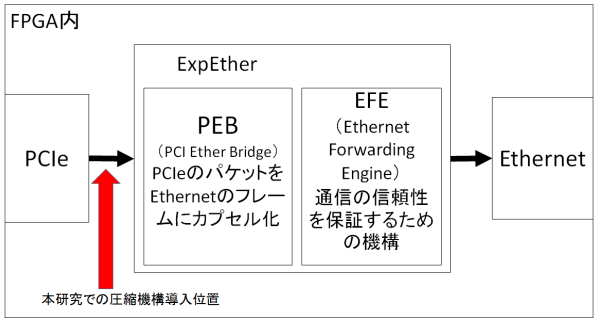

図 2: ExpEther の簡略図

Proposal of packet compression technique in ExpEther
†Hideki Shimura †Takuji Mitsuishi †Hideharu Amano
†Keio University

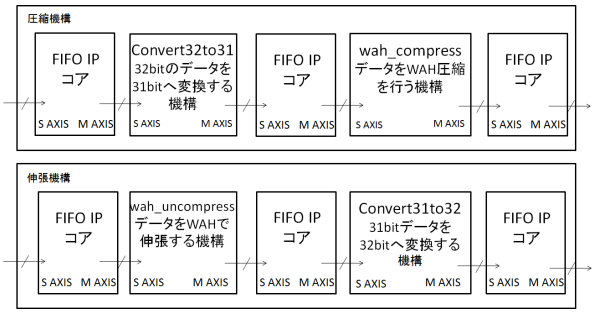

図 3: 圧縮および伸長機構

5 評価

5.1 シミュレーションの評価環境

評価アプリケーションとして、Graph500 ベンチマークで提供される幅優先探索 (BFS) のリファレンス実装 [5] を GPU 向けに修正したものを想定する。このアプリケーションではデータサイズ固定の bitmap 形式の中間データが GPU 間で通信される。中間データの通信は探索の深さ毎に行われ、探索の序盤, 終盤（data1,5,6）では中間データがほぼ bit-0 で構成され WAH での圧縮が行い易く、探索の中盤（data2,3,4）では bit-0,bit-1 が混載しエントロピーが大きいため圧縮が行いにくい。本稿ではサイズがそれぞれ 128KB の data1-6 を用いて評価を行う。

評価は ZedBoard の zynq-7000 を使い圧縮時間、圧縮率の評価を行った。上記の環境とデータを利用し圧縮率、圧縮時間を測定した。動作周波数は 100MHz としている。

5.2 圧縮効率の評価

各 6 つのデータに対して圧縮効率の評価を行った。評価の対象は、圧縮を行わず ExpEther を通して行った転送時間と圧縮を行った場合の転送時間で比較を行う。各データの圧縮効率を図 2 とした。この評価から data1,5,6

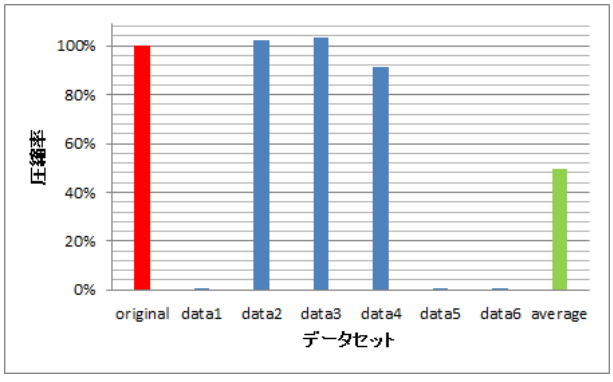

図 4: 各データの圧縮効率

は 0 の出現回数が多いため、圧縮率が 1 %以下となっている。対して data2,3,4 は 0 と 1 の混ざり合った出現パターンが多いため、data2,3 に関していうとヘッダーがついてしまい 100 %を超えた。ゆえに、この評価から圧縮サイズは平均で 49.7 %圧縮できることがわかる。

PCIe2.0 × 2 帯域幅は、2.00[GB/sec] なため 128KB データ転送時は 65.3 μ sec 時間がかかる。wah_compress モジュール一つでは、圧縮時間が約 683 μ sec(data1-6 全て) かかってしまっているので、PCIe からデータが転送されてくる速度に対応できず、圧縮機構で時間がかかってしまう。上記に対応するために動作周波数を 150MHz(1.5 倍) にし、この圧縮モジュールを 7 並列にすることにより 65.04 μ sec で転送が行えると考え、この問題を解決できると見込める。[4] に記載されている ExpEther で接続された GPU 間の転送スループットにより 128KB のデータを転送するには 137 μ sec かかることがわかる。図 4 のデータから最大 圧縮率 0.003%の圧縮率で 38.5 μ sec、最小 103.2%の圧縮率で 141 μ sec となり ExpEther のデータの転送速度は、圧縮機構を導入することで最大 3.6 倍、最小 0.97 倍の速度を提供することができる。

6 結論

本研究で、ExpEther は PCIe を拡張することを目的とした Ethernet を基盤とした仮想化技術であり、Ethernet を利用しているため、バンド幅が小さい。ゆえに、送信パケットを圧縮した後データを転送し、転送先でデータを伸長することでデータ転送時間を削減するという提案を行った。

ExpEther に圧縮機構を導入したことで、転送速度は最大 3.6 倍、最小 0.97 倍となった。しかし、圧縮率や転送速度がデータに依存してしまうため、バンド幅を改善したことにはならない。ゆえに、データの依存性を解決しバンド幅を改善することが今後の課題である。

参考文献

[1] Jun suzuki, et al. ExpressEther-Ethernet-Based Virtualization Technology for Reconfigurable Hardware Platform.
[2] Kesheng Wu, et al. Optimizing Bitmap Indices with Efficient Compression.
[3] Shimpei Nomura, et. al. Performance analysis of the multi-GPU System with ExpEther.
[4] Takuji Mitsuishi *Mitsuishi,Breadth First Search on Cost-efficient Multi-GPU Systems*.
[5] *Graph500, http://www.graph500.org/.*