C-041

ExpEther NIC 向けの圧縮手法 SECOMPAX の実装

志村英樹 †　三石拓司 †　菅真樹 ††　吉川隆士 ††　天野英晴 †
† 慶應義塾大学大学院理工学研究科 ††NEC システムプラットフォーム研究所

1 あらまし

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。ExpEther は Ethernet 部分のトラフィックが混みデータ転送時間に影響がでるという問題を抱えている。そのため、ExpEther に圧縮機構を設けることで Ethernet 部分のトラフィック削減することがこの論文の目的である。本研究では bitmap 形式のデータを利用し圧縮を行いデータの削減率の評価を行う。

2 ExpEther

ExpEther は、Ethernet を通じて PCIe の拡張のために開発された仮想化技術である。ExpEther には PEB(PCIe-to-Ethernet-Bridge) という PCIe のパケットを Ethernet のフレームへカプセル化を行う部分が存在する。PEB の機構により、恰も他のマシンに繋がっている PCIe デバイスをホストマシンに直接繋がっているように見せかけることで PCIe の拡張を行っている（図 1）。他に EFE（Ether-Forwarding-Engine）という再送や輻輳制御を行う部分があり、ExpEther は PEB と EFE の二つで構成されている。

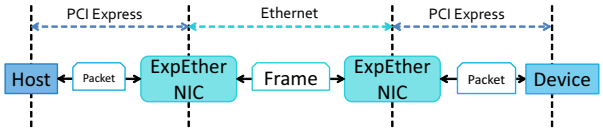

図 1: ExpEther の構成

3 ExpEther への圧縮機構の提案

ExpEther の Ethernet 部分は、トラフィックが混みデータ転送速度が落ちてしまう問題がある。ゆえに、ExpEther の通信時のトラフィックを削減するために、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量を削減することを提案とする。圧縮機構は PEB 部分に実装することを想定する。

データ削減には SECOMPAX(Scope-Extended COMPressed Adaptive index)[2] とよばれ、bipmap データの圧縮を行うアルゴリズムを使用した。

SECOMPAX は 31bit を一つのデータ長として bit 列をタイプ分けし圧縮を行う。31bit すべて 0 の場合を Fill-0、31bit すべて 1 の場合を Fill-1 とする。他に、31bit を 1byte ずつ区切ったときに 1byte のみ違う場合を NI(Nearly Identical) と呼び以下のように分けられる。

- 0000000 01001010 00000000 00000000 (NI-0)
- 1111111 11111111 01010101 11111111 (NI-1)

上記に当てはまらないものは L（Literal）として扱うが、NI は Literal として扱われる。そのため、入力 31bit は F か L に分類される。そこでタイプ分けを行った 1 つを 1 ワードとして扱い、SECOMPAX では 3 ワードずつ処理を行う。その圧縮処理を図 2 と図 3、図 4 とした。

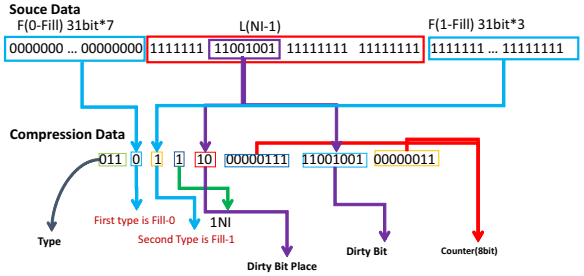

図 2: type FLF の場合

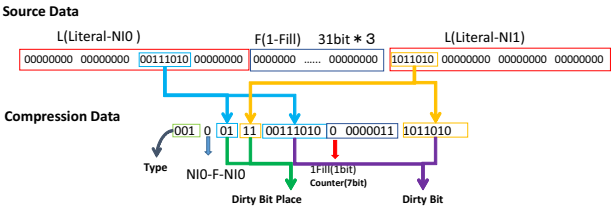

図 3: type LFL(L が同タイプ) 場合

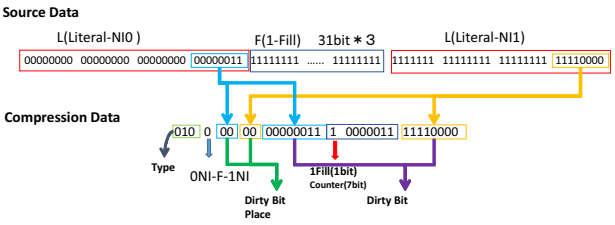

図 4: type LFL(L が違うタイプ) の場合

Data compression algorithm SECOMPAX for ExpEther NIC
†Hideki Shimura　†Takuji Mitsuishi　†Hideharu Amano
†Keio University

4　圧縮機構の実装

SECOMPAX はデータ入力を 32bit として実装を行ったため、PCIe のデータ入力 128bit/clock に合わせるため並列化も行った。図 5 のように圧縮および伸長機構の実装を行った。32bit を 4clock でデータを入力するため 16 並列の実装を行い図 6 とした。

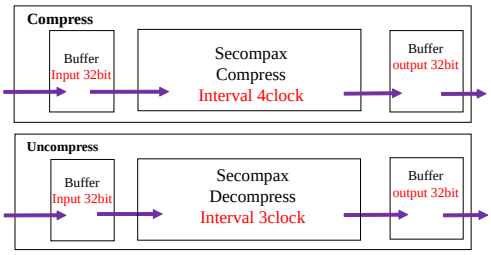

図 5: SECOMPAX の圧縮・伸長

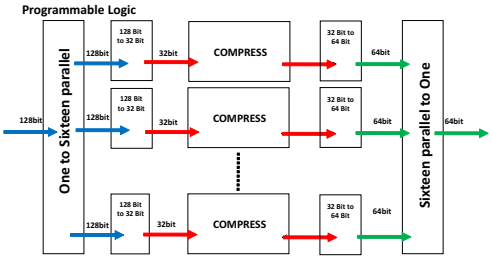

図 6: SECOMPAX の並列化

5　評価

5.1　対象アプリケーション

評価環境は表 1 の通りになっており、ExpEther の代わりに Zynq-7000APSoC を使用した。対象データは 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 [3] を GPU 向けに修正されたものを利用した。データサイズはスケールサイズによって変化し、スケールは 14-25 で測定を行った。スケールサイズはグラフの頂点数を示しており、スケール 14 の場合は 2 の 14 乗個の頂点数が存在すると定義されている。データの傾向としては、探索の序盤, 終盤では中間データがほぼ bit-0 で構成され、探索の中盤は bit-0,bit-1 をが混載したデータとなっている。

5.2　圧縮効率の評価

各スケールサイズごとに評価を行った。図 7 を SECOMPAX の各スケールサイズの圧縮後のデータサイズとした。上記で示したように探索の序盤、終盤のデータは bit-0 が多く構成されているため圧縮率が高い。反対に、探索の中盤のデータは bit-0 と bit-1 が混載しているので圧縮率が低い。スケールサイズ 14 の場合は平均で 56.4%、スケールサイズ 25 の場合は平均で 28.1%の圧縮率となった。

表 1: Implementation environment

FPGA	XC7Z020-1CSG484
HLS	Vivado HLS 2014.2
FPGA Design	Vivado 2014.2

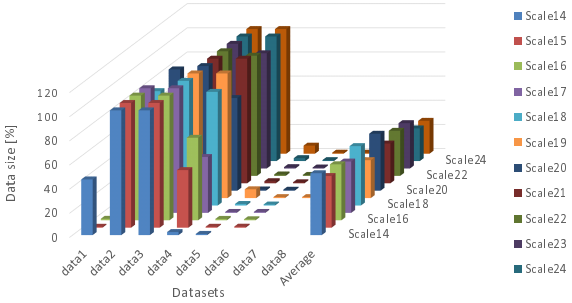

図 7: SECOMPAX を使った圧縮サイズ

6　結論

本研究では、ExpEther を利用し多くのパケット通信を行った場合、Ethernet 部分でトラフィックが混んでしまいデータ転送速度が遅くなる問題が生じた。ゆえに、パケットを圧縮することでデータ通信時のトラフィックを削減する提案を行った。そこで SECOMPAX という bitmap 形式のデータ圧縮に適したアルゴリズムをストリーム形式に改良し実装を行った。

結果、スケールサイズが 14 のときデータサイズは 56.4%、スケールサイズ 24 のときは平均で 28.1%という結果になった。今後の展望として、本研究では PCIe のデータ転送速度に合わせるため SECOMPAX のモジュールを並列化を行った。しかし、並列化を行うと使用する FPGA リソースが増えるため ExpEther の機能を圧迫する可能性があるため、今後の研究としては PCIe の入力データを直接圧縮するようなアルゴリズムを提唱する必要があると考える。

参考文献

[1] J. Suzuki, Y. Hidaka, J. Higuchi, T. Yoshikawa, and A. Iwata. Expressether - ethernet-based virtualization technology for reconfigurable hardware platform. In *High-Performance Interconnects, 14th IEEE Symposium on*, pp. 45–51, Aug 2006.

[2] Yuhao Wen, Zhen Chen, Ge Ma, Junwei Cao, Wenxun Zheng, Guodong Peng, Shiwei Li, and Wen-Liang Huang. Secompax: A bitmap index compression algorithm. In *Computer Communication and Networks (ICCCN), 2014 23rd International Conference on*, pp. 1–7, Aug 2014.

[3] Graph 500. http://www.graph500.org/.