

## FPGA, CPU, GPU を用いたヘテロジニアスコンピューティングの

### 研究

#### Study of heterogeneous computing using FPGA, CPU and GPU

工藤知宏

東京大学 情報基盤センター

#### 1. 研究目的

様々な処理が必要となる大きなジョブの実行を効率よく行うためには、処理の性質により CPU, GPU, FPGA などの様々な処理装置を組み合わせる必要がある。しかし、このようなヘテロジニアスな処理環境を組み合わせるクラウド的に使うためのフレームワークの研究開発はほとんど進められていない。そこには、異種の処理エンジンを組み合わせるジョブをどうプログラムするか、資源の割り付けをどう決めるか、多数の異種資源上の実行をどう管理するかなどの様々な新たな技術が必要となる。本研究では、CPU, GPU, FPGA を組み合わせる処理を行う際に、実行するジョブの内容に応じて CPU, GPU, FPGA を組み合わせ、これらの間のデータ転送を管理しつつジョブの実行を行うジョブ実行フレームワークである FlowOS の機能を MCRP-FPGA システムに実装し、その効果を評価することを目的とする。

#### 2. 研究成果の内容

FPGA と GPU を用いるヘテロジニアスコンピューティングの例として、FPGA と GPU の協調処理によるネットワーク型不正侵入検知システム (NIDS) を構築した。これは、広帯域の I/O を備え一定帯域でデータを欠落させることなく入力処理することができる FPGA の性質を生かし、FPGA 上に簡易なプレフィルタを実装して、100Gbps を超える大量のトラフィックから 10% 程度の怪しいパケットを抽出し、抽出されたパケットのみを GPU に転送して、様々な長さの文字列に対する完全なマッチングを行うものである。FPGA から GPU へのパケットの転送には CPU を介さない低レイテンシで広帯域の DMA による転送を活用している (図 1)



図 1 : FPGA と GPU を用いた NIDS システムにおけるデータの流れ

FPGA を用いたプリフィルタについて、OpenCL を用いて実装を行った。400Gbps の入力を処理することを想定し、図 2 に示すように、256Bytes の入力に対して同時に 1000 個の 3 Bytes シグネチャとのマッチングを行う設計とした。400Gbps の入力を処理するためには、195.3MHz で動作する必要がある。

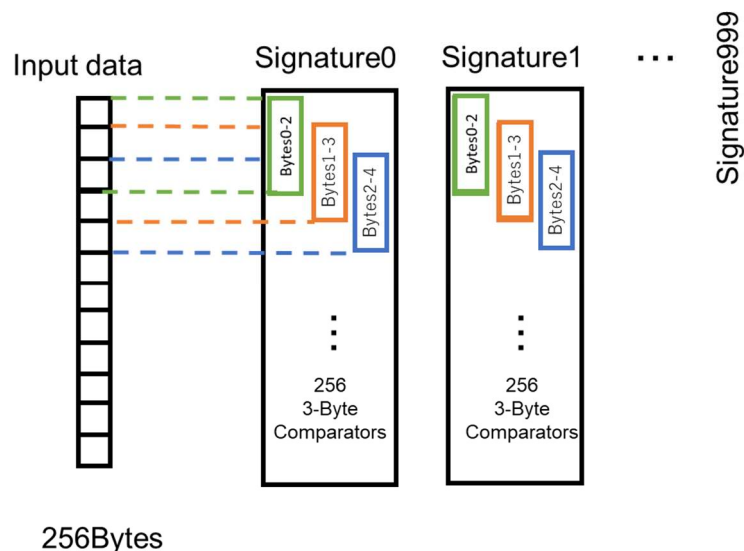


図 2 : FPGA を用いたプリフィルタにおけるマッチング

FPGA のプログラミングは OpenCL を用いて行った。シグネチャ数 30 では、300MHz で動作することが確認できた。しかし、リソースの使用率には十分な余裕があるにもかかわらず、シグネチャ数を増やすとなんらのレポートを生成することもなく回路合成に失敗するという問題が発生し、シグネチャ数 1000 個のマッチングを行う回路を生成することはできなかった。これは、合成ツールの問題であると考えられるが、それ以上研究を進めることはできなかった。

### 3. 学際共同利用が果たした役割と意義

FPGA と GPU 間の DMA による通信手法をはじめとした FPGA と GPU を協調して使用する手法および Cygnus で使用されている FPGA のプログラミングについて、筑波大学計算科学研究センターの研究者が持つ知見を得ることで研究の加速が図れた。

### 4. 今後の展望

本研究では Flow OS を実装する前段階として、FPGA と GPU を用いその間で大量のデータを転送するアプリケーションの実例として、NIDS の実装を試みた。OpenCL のような高位記述を用いたプログラミングは、本研究のような一定の帯域のデータを処理しなくてはならない用途にはあまり適していないこと、合成ツールにはまだ不十分な点があることが確認できた。今後は、本研究のようなアプリケーションを、性能を確保しつつ簡易に実装できる手段が必要になると考えられる。

5. 成果発表

- (1) 学術論文 なし
- (2) 学会発表 なし
- (3) その他

菊地駿太、池上 努、Akram ben Ahmed、工藤知宏、小林諒平、藤田典久、朴泰祐「FPGA/GPU 協調によるネットワーク型不正侵入検知システムの構築」信学技報, vol. 120, no. 338, CPSY2020-42, pp. 113-118, 2021 年 1 月

使用計算機	使用計算機 に○	配分リソース※	
		当初配分	追加配分
Cygnus	○	2000	0
Oakforest-PACS			
※配分リソースについてはノード時間積をご記入ください。			