

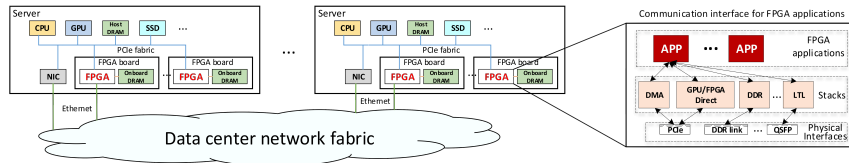
Communication Architectureが固定的

- 例えば、ToR越しに直接NICで接続された2台のFPGAの環境で、同じPCIeドメインにいる場合のみリモートで利用が可能
 - ただし帯域を食いつぶすことができず無駄になる
 - スケーラビリティにも難

リソースへのアクセス性とunified naming

- DCリソースという面でいえば潜在的にFPGAから利用できるリソース (computation/memory resource) は大規模である。
- しかし実際に利用を考えるとFPGAがリモートサーバのDRAM、SSD、GPUなどのリソースを利用するのは難しい。
 - これはnamespaceがlocal serverであることに起因している。

- unified naming schemeを定義し、リモートから特定のリソースにアクセス/利用できるようにしたい。



Resource Multiplexerが貧弱

- DCリソースをプールとして見るときにFPGAで考えるべきはマルチプレクサである。
 - ex. 同一リソースへの複数アクセス時のmux/demux
 - 実装に354行程度HDLコードが追加が必要
- 現時点で、一般的な物理インターフェスのmultiplexing schemeは存在しない。(先行研究はある)
 - 同じPCIeバスを利用する別リソースへの同時アクセスも難しい
 - ローカルホストDRAMと、SSDへの通信など
- 更に異なるインターフェース間、Stack間のmultiplexingは現在考えられていない。

通信インターフェースの実装が困難

- アプリケーション開発者に高いプログラミングスキルが求められる。
- 特にベンダー依存、実装依存が強い場所でもある。
 - 基本的に既存のStackは利用できる。
 - ただしベンダー同士で相互互換性がなかったりなどもする。
- 異なる通信インターフェースにかかる実装コスト一覧

Resource	Communication stack	LoC
Host DRAM	DMA	294
Host CPU	FPGA host stack	205
Onboard DRAM	DDR	517
Remote FPGA	LTL	1356

非効率なCommunication Stack

- コミュニケーションStackの実装・選択が効率に大きく影響を与える。
 - 例えば、PCIeを通したFPGA-to-FPGAの通信メカニズムでは通信開始前の同期処理が入る。
 - 例えば、リモートホストのDRAMへアクセスする際は、LTLを利用しないとCPUを中継する必要が発生し性能が下がる。