第 4 章

ExpEther を用いたマルチ GPU システム

本章では既存の複数ノード構成のマルチ GPU システムにおける問題点を提示するとともに，それに対する提案手法である ExpEther を用いて実現されるマルチ GPU システムについて示す．

4.1 複数ノード構成マルチ GPU システム

マルチ GPU システムについて，一定の規模を超えると複数のノードをネットワークにより相互接続して実現されるクラスタ構成が一般的である．クラスタ構成が用いられる要因はノードの筐体サイズや電源，またポート数から単一ノードあたりに搭載可能な GPU の台数が最大でも 4 台程度となるからである．複数ノードを接続するネットワークに拡張性を持たせることによって，クラスタ構成では利用できる GPU の数についてスケーラビリティがある．

複数ノード構成における問題点

複数ノード構成では，まずシステムを構築する際に手順が増すという問題が挙げられる．単純に 1 ノードで構成されるシステムと比較して，複数ノード構成では GPU を搭載した複数のノードを準備するために手間が増える．例えば複数のノードそれぞれに OS や CUDA, OpenCL といった開発環境を整えて，設定を行う必要がある．ノードを相互接続するネットワークに関する設定を行う必要もまた生じる．加えて必須ではないが複数のノードで共有されるようなストレージシステムなどを導入した方が開発は容易になる．GPU は物理，化学，経済などの様々な分野への問題解決へと用いられているため，利用者は情報科学の専門家に限らない．多様な利用者に対して，GPU に関する知識のみならずネットワークやストレージシステムといった情報科学分野に対する知識が要求することでマルチ GPU システムの利用の敷居を高いものとしている．

加えてプログラミングモデルが複雑化するという問題が挙げられる．複数ノードのマルチ GPU システムを制御するためには GPU を制御するための並列プログラミングと複数ノードに対する並列処理を記述するための並列プログラミング，2 つの並列プログラミングに関する知識と技術が必要となる．アプリケーションの中に存在するいくつかの粒度の並列性に対してどの階層の並列処理で扱うかを適切に選択して最適化を行う必要がある．処理の順序などを示すポリシーもまた混在する複数の並列プログラミングの双方のものを併用し，適切に記述を行うことが難しくなる．また，これらの困難を経て記述されたプログラムについてのデバッグを行うこともまた困難である．バグの要因が並列プログラム 2 階層分に広がることに加え，この 2 階層の並列プログラミングに対してそれぞれにデバッガが用意されるものの，組み合わせて扱えるような環境がないためである．結果としてアプリケーションの開発にかかる時間が増大し，マルチ GPU の利用を困難なものとしている．

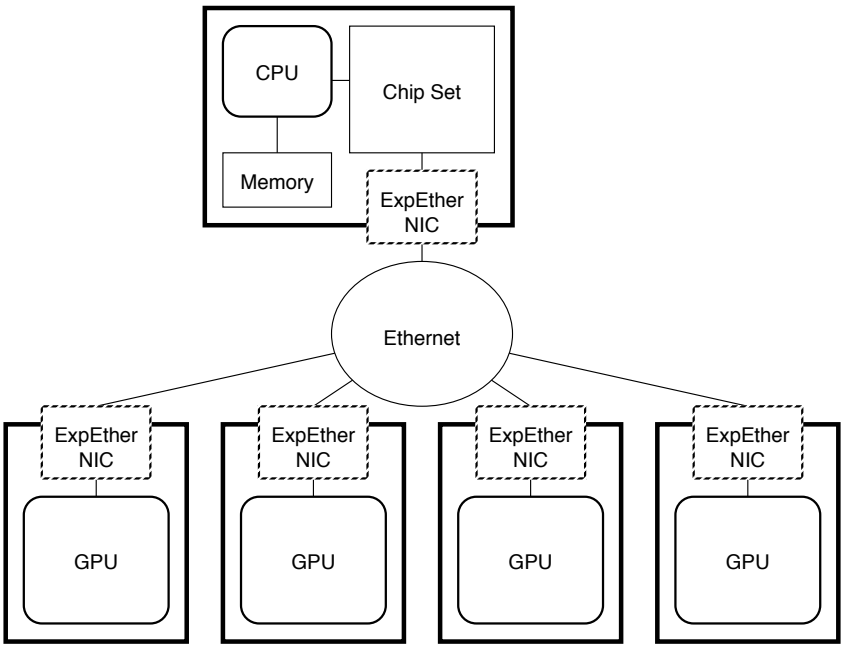

図 4.1: ExpEther を用いた単一ノード構成マルチ GPU システム

これらの問題点に対して第 2.5 節で示したような関連研究が存在するが，本研究では ExpEther を用いることで GPU の台数のスケーラビリティをもつ単一ノード構成マルチ GPU システムを提案する．

4.2 ExpEther を用いたマルチ GPU システム

複数ノード構成マルチ GPU システムでの問題点に対して，システムバス仮想化技術 ExpEther を用いることでスケーラビリティを維持しつつ単一ノード構成を実現する．また ExpEther の機能を生かして実現されうる応用的なマルチ GPU システムについて示す．

4.2.1 ExpEther を用いた単一ノード構成マルチ GPU システム

提案手法では単一ノード構成をとることにより複数ノード構成時に生じる問題が解消される．ExpEther を用いると図 4.1 に示すように単一ノードのみで多くの GPU デバイスの制御が可能なシステムが構築可能となる．これは ExpEther によってシステムバスに高拡張性を持たせることでき，単一のノードに対して接続可能な GPU デバイス数をスケーラブルにすることが出来るからである．結果として複数ノード構成マルチ GPU システムでのネットワークや複数のノードについての環境構築やプログラミングモデルの複雑化による開発コストの増加といった問題を避けることが出来る．

この構成を用いる場合，L2 スイッチを使ったネットワークの拡張によって PCIe の規格で定められるデバイスの上限である 256 台の接続を許容する．また Ethernet をより広帯域なものに変更することなどの手段で，リンクのバンド幅はスケーラビリティを持っている．現在は PCIe と 2 本の 10G Ethernet のブリッジが可能となっており，また今後 40 G Ethernet などの新たな Ethernet 規格への対応によって，より高いバンド幅の利用が可能になると考えられる．

図 4.2: GPU-BOX

提案システムでは複数ノード構成で挙げられたいくつかの問題点を回避することが出来るが，一方でノード内でのコミュニケーションであるホスト－デバイス間のコミュニケーションについては PCIe の規格で直接つないだ場合と比較してしまうと遅延が増加しバンド幅も小さくなる．そこで提案システムにおけるホスト－デバイス間コミュニケーションの性能低下の度合いやそれがアプリケーション実行時に全体の性能に与える影響について第 5 章で評価する．

4.2.1.1 GPU-BOX

本研究では提案システムの構築のために GPU-BOX と呼ばれる ExpEther I/O 拡張ユニット (図 4.2) の試作機を用いている．GPU-BOX は電源，PCIe スロット，また十分な容積をもち GPU を搭載することが可能となっている．また ExpEther の機能を備えて Ethernet ポートから PCIe パケットをカプセリングして転送することが出来る．

GPU-BOX の構成は図 4.3 に示すようなものとなっている．本研究で利用する GPU-BOX は最大で 8 台の GPU を搭載することができる．供給される電源は最大で 3000 W であり，Ethernet ポートとして 1 GPU あたり SFP+ が 2 つ割り当てられ，最大 20 Gbps の帯域幅が与えられる．また各スロットには FPGA 上に実装された ExpEther NIC によって ExpEther の各機能が提供されている．

4.2.2 応用的なマルチ GPU システム

最後に本研究で実現し評価を行う基本的な ExpEther を用いたマルチ GPU システム以外に，今後 ExpEther の機能を用いて実現可能だと考えられるいくつかのマルチ GPU システムの例を示していく．本研究における評価は今後開発されるであろうこれら応用的なシステムに対しての予備評価という側面も持っている．

ケース 1: 複数ユーザでの複数 GPU の共有

ExpEther を用いることで端末と GPU を離れた場所に配置することが可能になる．PCIe についてケーブルの仕様 [22] が定められており，ケーブル長は数メートル以内と非常に短い距離でないと接続することが出来ない．しかしながら ExpEther によって Ethernet を介することでシステムバスを離れた距離まで延ばすことが可能になる．GPU のような高性能計算機は廃熱や騒音，サイズ

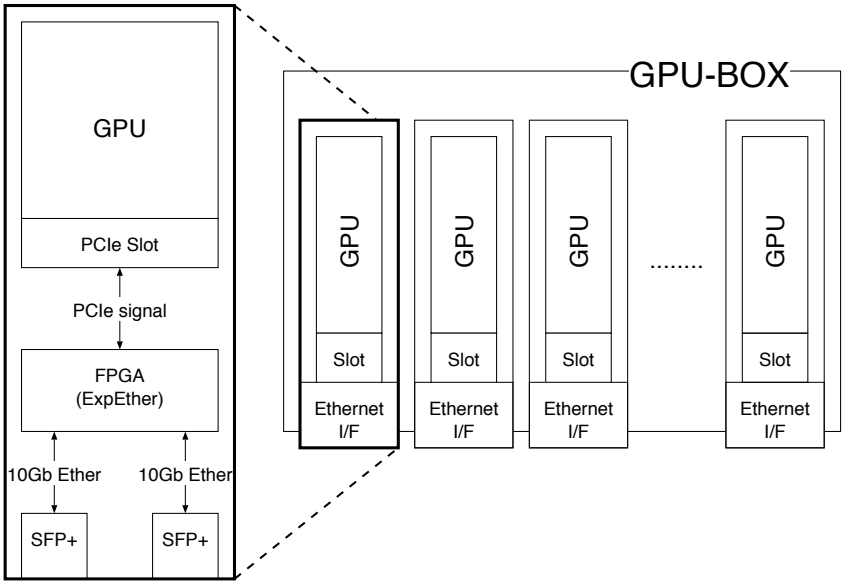

図 4.3: GPU-BOX の構成

の大きさなどから周囲への配慮が求められることもあるため，配置に関する制約からの解放されることの利点は小さくない．

GPU の配置の制約から解放されることで複数のユーザの GPU を一所に集めて管理コストを低減することも可能である．このときそれぞれの端丸が常に GPU を必要とするわけでなければ，端末に必要なときだけ GPU を割り当てることで，全体での GPU 数の減らし GPU の利用率の向上を図ることが出来る．ユーザは必要としたときのみ ExpEther のシステムマネージャを介して必要な数だけ GPU を割り当ててもらい，使い終わったら再度システムマネージャに問い合わせて GPU を解放することで実現できる．組織内でのリソース管理コストでの低減のためや IaaS (Infrastructure as a Service) での活用が考えられる．

ケース **2:** ミドルウェアによる **GPU** システムの抽象化

より発展したケースとしてユーザに実際に利用する GPU の数を指定させるのではなく，管理者はフレームワークなどの形でユーザがデバイスにアクセスするためのインタフェースを与えるということが考えられる．ユーザがリクエストを行った際に指定したパラメータから必要だと考えられる GPU を含むリソースを自動的に確保し，処理の結果のみを返すという形で実現される．PaaS (Platform as a Service) や SaaS (Software as a Service) といったサービス形態での ExpEther の活用となる．

このケースではユーザは GPU の存在を特に意識することなく利用できるため，アクセシビリティの向上へとつながる．またサービス側ではユーザが指定する処理内容やパラメータの値によって，システムマネージャを介して最適なシステム構成をとることが可能となる．