2A-05

ルートコンプレックスの仮想化によるCPUとI/Oデバイスツリーの PCI-Expressプロトコルレベルでの分離

辻　聡†　　鈴木　順†　　林　佑樹†　　菅　真樹†　　宮川　伸也†
馬場　裕司†　　木村　司‡
†日本電気株式会社　　‡日本電気通信システム株式会社

1　はじめに

PCI-Express(PCIe)はUSBコントローラやネットワークインタフェース(NIC)など，コンピュータ内部の様々なPCIeデバイスを接続するのに用いられている．また，ExpEther[1]のようにPCIeインターコネクトをコンピュータの筐体外に延長する技術の登場により，筐体の大きさやPCIe拡張スロットの数に依存せずにPCIeデバイスの追加を行うことが可能になってきた．

一方で，ルータなどの専用機器では機器内部ではPCIeが利用可能だが，PCIeデバイスの増設は難しいという課題がある．これは，PCIe拡張スロットを備えないなど，元々そのような拡張を想定していないハードウェア設計によるものである．また，ExpEtherのような技術で増設する場合，PCIeインターコネクトを筐体外に延長するための専用ハードウェアインタフェースを備える必要があり，PCIeデバイスの拡張を想定していない機器への導入は難しい．

本論文では，ソフトウェアベースでPCIeの仮想的なインターコネクト(PCIe仮想インターコネクト)を構築することで上記の課題を解決する方式を提案する．

PCIe仮想インターコネクトは，ソフトウェアで実現されたPCIeルートコンプレックス(仮想ルートコンプレックス(Virtual Root Complex:VRC))をルートとし，イーサネットなどの汎用ネットワーク上のPCIeデバイスをエンドポイントとするデバイスツリーである．このPCIeデバイスは，機器上で稼働するOSやアプリケーションからは，あたかも機器内部に設置されたPCIeデバイスのように使用可能である．

VRCはOSやデバイスドライバの下層に位置し，PCIeデバイスへのアクセス要求をトラップし，汎用ネットワークを通じてPCIeデバイスに転送する．これによって，OSやデバイスドライバの改変を最小限，あるいは全く行わずに，様々な機器にPCIe仮想インターコネクトを導入可能となる．

本論文では，本方式の実証のためにVRCを仮想マシンモニタ(Virtual Machine Monitor:VMM)に組み込む形で実装し，仮想マシン(Virtual Machine:VM)からネットワーク上のNICを通じたパケットの送受信を確認したことについて述べる．

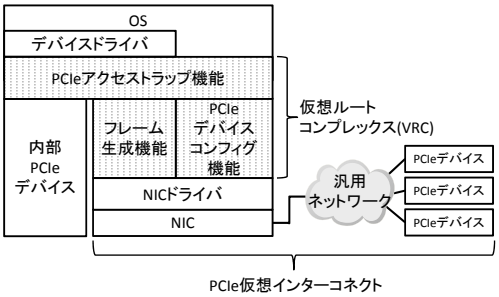

図1: システムアーキテクチャ

2　提案手法

図1は本提案のPCIe仮想インターコネクトを実現するシステムのアーキテクチャを示したものである．PCIe仮想インターコネクトはVRCをOSやデバイスドライバの下層に導入することで実現される．

VRCは，(1)OSやデバイスドライバからPCIeデバイスへのアクセス要求をトラップする機能(図1のPCIeアクセストラップ機能)，(2)トラップしたアクセス要求を汎用ネットワーク上のPCIeデバイスへ転送するためのフレームを生成する機能(図1のフレーム生成機能)，(3)PCIeデバイスのコンフィギュレーション機能(図1のコンフィグ機能)，から構成される．

PCIeアクセストラップ機能によって，OSやデバイスドライバからのPCIeデバイスへのアクセス要求を取得することができる．このアクセス要求にはアクセス先のPCIeデバイスを識別する情報などが含まれている．ネットワーク上のPCIeデバイスへのアクセスの場合に，この情報を元にフレーム生成機能がPCIeにおけるデータ送受信用のパケット(Transaction Layer Packet:TLP)を生成し，さらに汎用ネットワークのフレームでカプセル化を行う．このカプセル化されたフレームがNICを経由して送信される．これにより，NICさえ備えていれば汎用ネットワークを介してPCIeデバイスの増設

I/O Device Tree Separation from CPUs at PCI-Express Protocol Level by Root Complex Virtualization
†Akira TSUJI　†Jun SUZUKI　†Yuki HAYASHI　†Masaki KAN
†Shinya MIYAKAWA　†Hiroshi BABA　‡Tsukasa KIMURA
†NEC Corporation　‡NEC Communication Systems

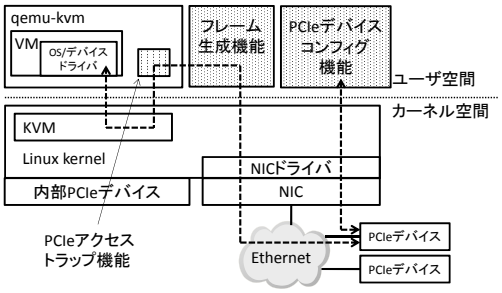

図 2: 実装構成

が可能な構成となっている．

コンフィグ機能はネットワーク上の PCIe デバイスの PCI コンフィギュレーションレジスタを設定する．本来，このレジスタは BIOS や OS が設定を行うが，本提案では VRC の起動後にネットワーク上の PCIe デバイスにアクセス可能になるため，VRC がネットワーク上の PCIe デバイスを設定する．この設定により PCIe デバイスに BDF 番号 (Bus, Device, Function 番号) が割り振られ，PCIe 仮想インターコネクトが形成される．

PCIe デバイスは BDF 番号もしくはアドレス値によって識別される．アクセス要求をカプセル化したフレームを PCIe デバイスに配送するために，コンフィグ機能はこれらの識別子と汎用ネットワークのアドレスの対応付けを行う．この対応付けを基にフレーム生成機能がカプセル化を行う．

3 実装

2 節で提案した PCIe 仮想インターコネクトの実現性検証のために，図 2 に示すように，KVM(Kernel-based Virtual Machine) 環境のユーザ空間で動くプログラムとして VRC を試作した．PCIe アクセストラップ機能は qemu-kvm の PCIe デバイスへアクセスを行う個所を改造することで実現し，VM 上で動く OS またはデバイスドライバから PCIe デバイスへのアクセスをトラップ可能にした．汎用ネットワークとしてイーサネットを用い，PCIe over ethernet 機能を提供するハードウェア [1] を利用して PCIe デバイスを接続した．

KVM のようなサーバ仮想化環境を用いた理由は，PCIe デバイスへのアクセスのトラップが容易なためである．OS を図 2 における VM 上の OS とみなすと，VRC は VMM 層 (qemu-kvm/KVM 層) と同じレイヤとなる．従って，VMM 層に VRC を組み込むことで，図 1 と同様のアーキテクチャを実現可能となる．

PCIe アクセストラップ機能は VM 毎に各 qemu-kvm 内に導入される．そのため，フレーム生成機能は PCIe デバイスの識別子だけでなく，qemu-kvm の識別子も基にしてフレームを生成する．

4 機能検証

本方式の実現性を確認するために，VM からネットワーク上の NIC を利用しての通信試験を行った．

KVM のホスト環境として Intel Xeon E3-1225v2，16GB RAM，CentOS 7(kernel-3.10.0-123)，Intel 82576 搭載 NIC，VM として仮想 CPU を 1 コア，2GB RAM，CentOS 7(kernel-3.10.21) を，対向ホストとして Intel Core2 Duo T8300，2GB RAM，Ubuntu 12.04(kernel-3.20-24)，Broadcom BCM 95906 搭載 NIC を用い，back-to-back 接続での ping による RTT を測定した．ペイロードサイズが 64 バイトの場合，VM が ping 送信側のときに 2.69ms，対向ホストが ping 送信側のときに 3.06ms であった．遅延が大きいため提案方式の性能改善は必要であるが，実環境で動くことを示したことで本方式の実現性を示せたといえる．

5 関連研究

ソフトウェア的にネットワーク越しのデバイスをローカル接続のデバイスとして扱う技術として，USB デバイスを IP ネットワーク越しに扱う USB/IP[2] が存在する．USB のデバイスドライバは階層構造となっており，USB/IP ではデバイスドライバ中の全 USB デバイスに共通するレイヤでローカル側とリモート側に分割し，間を IP ネットワークで接続する．

本方式はデバイスドライバや OS より下層のインターコネクトのレイヤでローカル側とリモート側を分割する．そのため，デバイス特有のソフトウェア構造などに依存しない方式である．従って，方式的には PCIe デバイスに限らず，より広い範囲のデバイスに対してネットワーク越しのデバイスを接続可能である．

6 まとめ

本論文では，PCIe デバイスの追加が難しい機器に PCIe デバイスの追加を可能にする PCIe 仮想インターコネクト構築方式を提案し，KVM を用いてその実現性を確認した．性能改善が今後の課題の 1 つである．

謝辞 本研究において多大な協力を頂いている，株式会社 iD に感謝致します．

参考文献

[1] Suzuki, J., Hidaka, Y., Higuchi, J., Yoshikawa, T., and Iwata, A. ExpressEther - Ethernet-Based Virtualization Technology for Reconfigurable Hardware Platform. In *Proceedings of the 14th IEEE Symposium on High-Performance Interconnects*, HOTI'06, pp. 45–51, 2006.

[2] Hirofuchi, T., Kawai, E., Fujikawa, K., and Sunahara, H. USB/IP - A Peripheral Bus Extension for Device Sharing over IP Network. In *Proceedings of the USENIX 2005 Annual Technical Conference, FREENIX Track*, ATC'05 FREENIX Track, pp. 47–60, 2005.