A Thesis for the Degree of Ph.D. in Engineering

Making GPUs First-Class Citizen Computing Resources in Multi-Tenant Cloud Environments

August 2018

Graduate School of Science and Technology
Keio University

Yusuke Suzuki

別表５
(3)

主　論　文　要　旨

No.1

報告番号	㊙ 乙 第　　号	氏 名	鈴木 勇介

主 論 文 題 名：
Making GPUs First-Class Citizen Computing Resources in Multi-Tenant Cloud Environments
(マルチテナントなクラウド環境下での GPU の第一級計算資源としての抽象化)

（内容の要旨）

Graphic processing unit (GPU) は高い並列性を持ち, GPU の汎目的計算での利用 (GPGPU) を促進している. GPGPU は様々な領域のアプリケーションにとって有用な手法となっており, その中にはサーバサイドワークロードも含まれる. GPGPU のサーバサイドワークロードでの適用や GPU の計算性能の向上は GPGPU アプリケーションのコンソリデーションにとって強いモチベーションとなっている. GPU を第一級の計算資源としてクラウドで抽象化することは, マルチテナントなクラウド環境において GPGPU アプリケーションのコンソリデーションを達成するために重要となっている. しかしながら, GPU の資源の仮想化に関する先行研究では, それぞれのアプローチのトレードオフは明らかではない. 様々なインターフェースのレベルにおける仮想化手法のトレードオフや技術的な困難さについて光を当てることで, より適切な GPU の資源仮想化手法の開発を促進することができる.

本研究では GPUvm と GLoop の 2 つの GPU 資源仮想化手法を提案する. GPUvm は Hypervisor レベルの GPU 仮想化手法で, 完全仮想化, 準仮想化そして高性能準仮想化の 3 種類の仮想化のモードを持つ. GPUvm は memory-mapped I/O や DRM APIs といった低, 高レベルなインターフェースをゲスト仮想マシン (VM) に対して提供する. 評価の結果 GPUvm はインターフェースのレベルによって異なったオーバーヘッドを示すということが明らかになった. GPUvm は GPU スケジューリングを用いて VM 間での粗粒度な GPU 利用の公平性を達成することができた.

本研究ではまた GLoop を提案する. これは先進的な GPGPU アプリケーションのコンソリデーションを可能とするソフトウェアランタイムである. 粗粒度の公平性はアプリケーション透過な手法によって達成することができたが, GPU eater とする近年の先進的な GPGPU アプリケーションは共有 GPU を占有してしまう. GLoop は GPU eater が存在する場合においてもコンソリデーションを可能とすべく, アプリケーションの変更を含む application-assisted な手法を取る. GLoop はイベント駆動型のプログラミングモデルを導入し GLoop アプリケーションの GPU eater の機能を維持したまま, プロポーショナルシェアポリシーを適用したスケジューリングを共有 GPU 上で可能とする. 本研究では GLoop のプロトタイプを実装し, 8 つの GPU eater を GLoop アプリケーションに移植した. 実験の結果, GLoop はコンソリデーションされた GPGPU アプリケーションをポリシーにそってスケジュールし, リソースアイソレーションを維持することが可能であることを示した.

本研究の貢献は次の 2 つにまとめられる. 第一に, GPU の完全仮想化のデザインと実装を示し, そのボトルネックを明らかにし, 仮想 GPU に対するより高いレベルのインターフェースがこのオーバーヘッドを削減することができることを示した. これはクラウドソフトウェア開発者が利用用途に沿った仮想化手法を選択するのに役立てることができる. また, GPU ハードウェアベンダが将来の GPU における仮想化向けの拡張機能のデザインを行うことに役立てることができる. 第二に, アプリケーション透過な手法の限界を示し, application-assisted なアプローチである GLoop が GPU eater を含む GPGPU アプリケーションのコンソリデーションを達成できることを示した. この手法はマルチテナントなクラウド環境がより幅広い GPGPU アプリケーション間で GPU を共有することを可能にする. また, GLoop はプリエンプションの機能を持たない GPU 以外のアクセラレータのクラウドでの共有に適用できる可能性を示している.

別表６
（３）

Keio University

Thesis Abstract

No.

Registration Number	☑"KOU" □ "OTSU" No. *Office use only	Name	Yusuke Suzuki

Thesis Title

Making GPUs First-Class Citizen Computing Resources in Multi-Tenant Cloud Environments

Thesis Summary

Graphic processing units (GPUs) provide massively parallel computational power and encourage the use of general-purpose computing on GPUs (GPGPU). GPGPU has become an attractive platform in various domains of applications including server-side workloads. Adaption of GPGPU in server-side workloads and scaling up of GPU computing capacity motivate the consolidation of GPGPU applications. Making GPUs first-class citizen computing resources in the cloud is a key to consolidation in multi-tenant cloud platforms. Despite the previous study on GPU resource virtualization, the tradeoffs between the approaches remain unclear. Shedding light on these tradeoffs and the technical requirements for the resource virtualization at various interface-levels would facilitate the development of an appropriate GPU resource virtualization solution.

This dissertation presents two approaches for GPU resource virtualization, GPUvm and GLoop. GPUvm is an architecture for hypervisor-level GPU virtualization. GPUvm offers three modes: the full-, naive para-, and high-performance para-virtualization. GPUvm exposes low- and high-level interfaces such as memory-mapped I/O and DRM APIs to the guest virtual machines (VMs). Our experiments show that GPUvm incurs different overheads as the level of the exposed interfaces is changed. The results also show that GPU scheduling can achieve a coarse-grained fairness among multiple VMs.

We also present GLoop, a software runtime that enables us to consolidate GPGPU applications including advanced GPU applications. While the coarse-grained fairness can be achieved by the application-transparent approaches, advanced GPGPU applications, referred to as GPU eaters, can monopolize a shared GPU. GLoop explores the way to achieve consolidation of GPU eaters by taking an application-assisted approach including modification of the applications. GLoop introduces an event-driven programming model to offer the GPU eaters' high functionality while scheduling them on a shared GPU with a proportional-share policy. We implement a prototype of GLoop and port eight GPU eaters on it. Our experiments show that our prototype successfully schedules the consolidated GPGPU applications on the basis of its scheduling policy and isolates resources among them.

The contribution of this dissertation is twofold. First, we show the design and implementation of full-virtualized GPUs, clarify the bottleneck, and show that the high-level interface for virtual GPUs can mitigate the overheads. This helps the cloud software developers to select an appropriate virtualization approach for their use cases, and helps GPU hardware vendors to design the future GPU hardware extension for virtualization. Second, we show the limitation of the application-transparent approaches, and show that the application-assisted approach can share a GPU even in the face of GPU eaters. This allows the multi-tenant clouds to share a GPUs with a wider range of applications. Moreover, GLoop envisions the clouds sharing not only GPUs but also other non-preemptive accelerators.