

# サーバレスコンピューティングにおける ハードウェアアクセラレータ仮想化機構の初期検討

小柴 篤史<sup>1,a)</sup> 佐野 健太郎<sup>1,b)</sup>

## 1. はじめに

近年のクラウドサービスにおいて、サーバレスコンピューティングが急速に普及している (AWS Lambda [1], Azure Functions [2]). サーバレスコンピューティングでは、仮想サーバのインスタンス化や環境構築、オートスケールといったサーバ管理を全てクラウドプロバイダが担う。クラウドユーザはサーバ管理の手間なくクラウドリソースを利用できることに加え、実際のアプリケーションの実行時間だけ費用がかかるため、レンタル中に常に費用が発生する従来の課金モデルよりもコストを抑えることができる。また、異なるユーザのワークロードに対するリソース割り当てやオートスケールをクラウドプロバイダが一括して行うため、リソース利用効率を向上できる。このように多くのメリットがあるサーバレスコンピューティングの市場規模は 2018 年より急速に増大しており [3], 次世代のクラウドサービスモデルとしての地位を確立しつつある。

一方、既存のサーバレスプラットフォームでは、ハードウェアアクセラレータを利用できない点が課題である。ビッグデータ解析や深層学習などの大規模演算を高速に処理するため、クラウドにおいてもハードウェアアクセラレータの需要が増加している。特定の演算に特化したアクセラレータへタスクの一部をオフロードすることで、汎用 CPU より数 100 倍の処理性能を達成できる。FPGA (Amazon EC2 F1) や GPU (Amazon EC2 P3), TPU (Google Cloud TPU) といった様々なアクセラレータを提供するクラウドサービスがあるが、これらは従来の Infrastructure as a Service (IaaS) モデルとして提供されている。IaaS モデルではサーバ管理や実行環境構築の手間がかかることや、スケールアップできない、高コスト等の課題がある。そのため、サーバレスコンピューティングにおけるハードウェアアクセラレータの利用が望まれている [4].

本研究では、サーバレスコンピューティングでアクセラ

レータを利用するための仮想化と資源管理を行うシステムソフトウェアについて検討する。その初期検討として、サーバレスコンピューティングのアクセラレータ制御に求められる要件と、既存技術の問題点を述べる。更にアクセラレータ仮想化機構のシステム設計と、FPGA を対象としたプロトタイプ実装の進捗について報告する。

## 2. サーバレスにおけるアクセラレータの要件

本研究では、サーバレスコンピューティングで実行される各ユーザのワークロードがホストアプリケーションとしてアクセラレータへ処理の一部をオフロードする場合を検討する。ホストアプリケーションはゲスト OS やコンテナ等の仮想実行環境で実行されるが、サーバレスコンピューティングの利点を損なわずアクセラレータを活用するには、サーバレスコンピューティングの要件とアクセラレータ制御に求められる要件を満たす仮想実行環境を提供する必要がある。

まず、サーバレスコンピューティングの要件としては、以下の 3 つが挙げられる [5].

*Isolation:* 異なるユーザのワークロードが同じ計算リソースを共有するため、ワークロード間のデータアクセス等が不可能なセキュアな仮想実行環境が求められる。

*Startup Latency:* サーバレスコンピューティングでは、ユーザのワークロードは指定されたイベントが起きた時に実行される (イベントドリブン)。そのため起動時のレイテンシ (cold start) は可能な限り短いことが望ましく、高い応答性が求められる。*Elasticity:* 各ワークロードのイベント発生や実行時の負荷変動に対応するため、柔軟な資源割り当てやオートスケール機能が求められる。

次に、アクセラレータをサーバレスコンピューティングへ適用する際、以下の要件が挙げられる。

*I/O overhead:* デバイスドライバを介した I/O 制御が頻繁に必要なために、コンテキストスイッチなどの OS オーバヘッドが性能低下を招く恐れがある。加えて、アクセラレータの命令メモリ初期化や入力データ転送などの初期化オーバヘッドは起動時間増加の原因となる。このよ

<sup>1</sup> 理化学研究所 計算科学研究センター, Hyogo 650-0047, Japan

<sup>a)</sup> atsushi.koshiba@riken.jp

<sup>b)</sup> kentaro.sano@riken.jp

うなアクセラレータ制御に伴うオーバヘッドの削減が必要である。Programmability: ホストアプリケーションは、OpenCL, CUDA, TensorFlowなどのライブラリを用いて記述されるが、多くの仮想化環境ではこれらのライブラリはサポートされていない。また、サーバレスコンピューティングで多用されるpythonやjavascriptなどの抽象度の高い言語とのギャップをどう埋めるかが課題となる。

クラウドにおける仮想実行環境は、主に(1)ゲストOS, (2)コンテナ, (3)ユニカーネルに分類される。ゲストOSは、KVM等のハイパーバイザ上で独立した仮想マシン(VM)として独立実行されるためセキュアだが、起動時間が長い、仮想化オーバヘッドが大きい等の課題がある。コンテナは通常のユーザプロセスと同等の仮想オーバヘッドで起動が速い等の利点があるが、コンテナ同士はOSを共有するために、セキュリティ面が課題である。これに対しユニカーネルはハイパーバイザ上で動作するライブラリOSで、フットプリントが小さく起動時間が短い(~10ms程度)。またVMとして実行されるためにセキュアである。更に、アプリケーションとOSが同一アドレス空間上で実行されるため、コンテキストスイッチのオーバヘッドが小さい。これらの特性はサーバレスコンピューティングとアクセラレータ制御の両方に適していると言える。そこで本研究では、ユニカーネルを用いたサーバレスコンピューティングに特化したアクセラレータ仮想化機構を提案する。

### 3. 提案機構

図1に提案機構の全体構成を示す。提案機構では、各ユーザのワークロードがアクセラレータのホストアプリケーションとして、ユニカーネル上で動作する。ハイパーバイザは各ホストアプリケーションの要求に応じてアクセラレータを割り当てる(allocator)。アクセラレータ上で動作するカーネル間のisolationを保証するため、各アクセラレータは単一のワークロードへ割り当てられ、ワークロードはパススルー方式でアクセラレータへアクセスすることでI/O overheadを削減する。これにより、ホストアプリケーションのisolation, startup latency, I/O overheadの要件を満たす。一方、ユニカーネルを用いるだけではelasticityとprogrammabilityの要件を満たすことはできないが、これを満たす新たな仮想化機能を提案する。ユニカーネルは、アクセラレータ制御用のデバイスドライバに加え、OpenCL/CUDA等の標準APIをサポートする仮想ライブラリを提供する。OpenCL/CUDA等で記述されたホストアプリケーションをユニカーネル上でそのまま実行可能にすることでprogrammabilityを保証する。また、利用可能なリソース量とアプリケーションの要求に応じた柔軟な資源管理を行うノード間スケジューラ(scheduler)、負荷分散機構(load balancer)をハイパーバイザに組み込むことでelasticityを保証する。

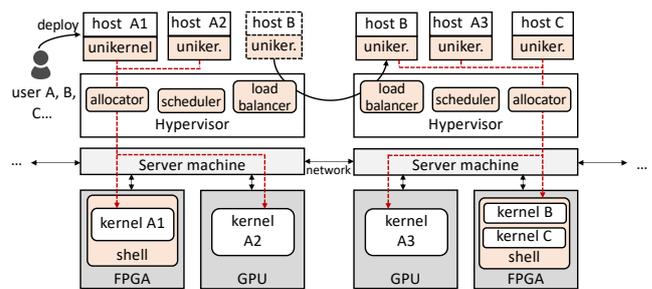


図1 An overview of the proposed system.

## 4. 実装の進捗

我々は現在、FPGAアクセラレータとOpenCLアプリケーションを対象としたプロトタイプの実装を進めている。実装環境としてZynq Ultrascale+ SoCを搭載したUltraZedボードを用いており、UltraZed上でハイパーバイザ(KVM)およびユニカーネル(solo5)が既に稼働している。現在はユニカーネルへのOpenCL仮想ライブラリと、ユニカーネルからのアクセラレータ制御部を実装しており、近日に動作確認と初期評価を行う予定である。また、本機構はAlveo U250などの高性能FPGAカードを搭載したクラウドサーバ環境への拡張実装も予定している。

## 5. おわりに

本研究では、サーバレスコンピューティングにおけるユニカーネルを用いたアクセラレータ仮想化機構の概要と進捗を述べた。今後の課題として、提案機構の実装と実機検証、複数アクセラレータの制御方式の検討が挙げられる。

**謝辞** 本研究は、JSPS 科研費 JP19K24360 の助成を受けたものである。

## 参考文献

- [1] : AWS Lambda, Amazon Web Services (online), available from <https://aws.amazon.com/lambda/> (accessed 2019-12-2).
- [2] : Azure Functions, Microsoft (online), available from <https://azure.microsoft.com/ja-jp/services/functions/> (accessed 2019-12-2).
- [3] Papadopoulou, E.-E.: The state of serverless computing: Current trends and future prospects, JAXenter (online), available from <https://jaxenter.com/state-serverless-trends-future-147183.html> (accessed 2019-12-2).
- [4] Hellerstein, J. M., Faleiro, J. M., Gonzalez, J. E., Schleier-Smith, J., Sreekanti, V., Tumanov, A. and Wu, C.: Serverless Computing: One Step Forward, Two Steps Back, *CoRR*, Vol. abs/1812.03651 (online), available from <http://arxiv.org/abs/1812.03651> (2018).
- [5] Fingler, H., Akshintala, A. and Rossbach, C. J.: USETL: Unikernels for Serverless Extract Transform and Load Why Should You Settle for Less?, *Proceedings of the 10th ACM SIGOPS Asia-Pacific Workshop on Systems, AP-Sys '19*, New York, NY, USA, ACM, pp. 23–30 (online), DOI: 10.1145/3343737.3343750 (2019).