公開

研 究 終 了 報 告 書

「新世代デバイスを用いた密結合型マイクロサービス実行基盤」

研究期間: 2019 年 10 月～2023 年 3 月
研 究 者: 坂本 龍一

1. 研究のねらい

Web アプリケーションの開発現場では開発を容易に行い、効率的、かつ、継続的にサービスを展開することが求められている。マイクロサービス開発は 1 つのアプリケーションを複数の小さなサービスに分割し、それらのサービスを疎に接続し、多数のサービスの集合体として 1 つのアプリケーションを開発する開発手法である。これにより 1 つのチームは 1 つのサービスの開発に注力することができ、開発の効率化が実現できる。また、サービスの数を増やすことのよって、容易にスケールする Web サービスを展開することが可能になる利点がある。

一方でマイクロサービスは実行効率が非常に悪いという課題がある。それぞれのサービスは Linux のコンテナ仮想化技術を用いている。そのため、サービスの数だけコンテナが必要になる。また、大量の Linux プロセスの生成、コンテナ生成が必要となる。これらの、各種サービスは TCP/IP を用いて通信を行っており、通信部分のオーバーヘッドが問題となっている。このように、実行時間の大半がプロセスの生成や TCP/IP の通信、インタプリタ実行などに費やされている。

そこで、これらの問題を解決するために新世代デバイスを用いた密結合マイクロサービス実行基盤の実現を目指す。これまで疎に接続されていたマイクロサービスを密に結合し、高い実行効率を有するマイクロサービスの実行基盤を実現する。マイクロサービスの高いモジュラリティを失うことなく新世代デバイスを有用に利用するために、①新しいハードウェアの抽象化を提供するハードウェアアブストラクションレーヤーの実現を目指す。さらに、②ハードウェアアブストラクションレーヤーを有効に活用するリソースマネージャを提案し、デバイスの特性を生かした適材適所の資源管理を実現する。

①では既存のマイクロサービスの高い利便性を損なうことなくネットワーク部分の密結合化を実現する。様々な Web アプリケーションを改変することなく通信部の高速化を実現する。具体的には SmartNIC によるトラフィック制御の高速化を実現する。さらに、コントロールプレーンと連携し、SmartNIC の詳細を隠ぺいするミドルウェアのライブラリの開発を行う。

②では、マイクロサービスの配置を最適化することにより、通信の削減、CPU 利用効率の改善によって性能向上を目指す。特に、サービス間の通信を事前に解析しサービスの配置を最適化する静的最適化と、実行時に最適なサービスを選択する動的最適化を行う。

2. 研究成果

(1) 概要

本研究では、マイクロサービスのネットワーク処理を効率的に高速化するためのアブストラクションレーヤーの探求、ネットワーク処理やトラフィック処理を高速化する専用アクセラレータの開発、アブストラクションレーヤーを実現するシステムソフトウェアの構築を行った。さらに、サービスの配置や選択を効率的に行う資源最適化を行うことによって資源利用率や応答性能の改善を行った。

マイクロサービスではコンテナ環境を用いているため、公開された既存のコンテナイメージを利用できることが望ましい。専用アクセラレータ向けのライブラリや API を提供する方法ではアプリの再コンパイルやコンテナイメージの再構築が必要となる。そのため、専用 API や専用ライブラリを提供する方法ではなく、アプリを改変しなくても済むアブストラクションレーヤーを提供・実現することが重要となる。また、スケーリングや分散環境のためのトラフィック処理がマイクロサービスでは多く行われており、これらのトラフィック制御をサポートする必要があった。

そこで、マイクロサービスで用いられているサービスメッシュ部分に着目し、FPGA SmartNIC をサービスメッシュの中に隠ぺいすることとした。さらに、eBPF を用いてソケット通信をハイジャックすることで既存のコンテナ環境を生かしたままアクセラレータを利用するネットワークミドルウェアを開発し、専用アクセラレータを隠ぺいするアブストラクションレーヤーを実現した。これにより、既存のコンテナイメージを編集することなく直接、提案する環境にコンテナを導入することができる環境を実現した。しかしながら、サービスメッシュ内のトラフィック処理の高速化には FPGA の深い理解が必要であり汎用性に問題があった。そこで、SmartNIC 上で動作する WebAssembly 環境を構築し、サービスメッシュにおけるトラフィック処理をハードウェアを意識することなく様々な言語から制御できる環境を実現した。

ネットワーク部の効率化では TCP オフローディングと合わせてサービスメッシュ機能を複数の SmartNIC にオフロードする手法を提案し、マイクロサービスの応答性能を改善した。また、サービスメッシュ向けのフィードバック型資源管理を提案し、データの収集・フィードバック処理を高速化するためのデータ収集分散基盤アクセラレータを開発した。

マイクロサービスではサービス間で多くの通信が発生するため、サービスの配置最適化が重要となる。そこで、サービス間の通信頻度や通信量を基に複数サーバーに適切にサービス配置を行う配置最適化によって応答性能の改善を達成した。また、多数のレプリカ環境において、リクエスト到着時に最適なレプリカを選択する動的レプリカ選択手法によって応答性能を改善する手法を実現した。

(2)詳細

SmartNIC を隠ぺいするマイクロサービス向けネットワークミドルウェア

本研究では、マイクロサービスで多用されるサービス間ネットワークを専用ハードウェア化することにより応答性能の改善を行う提案ネットワークミドルウェアが、これらの専用ハードウェアをマイクロサービスに対して隠ぺいすることが非常に重要である。多くの専用アクセラレータを用いる研究ではアクセラレータを利用するための専用 API やライブラリを用

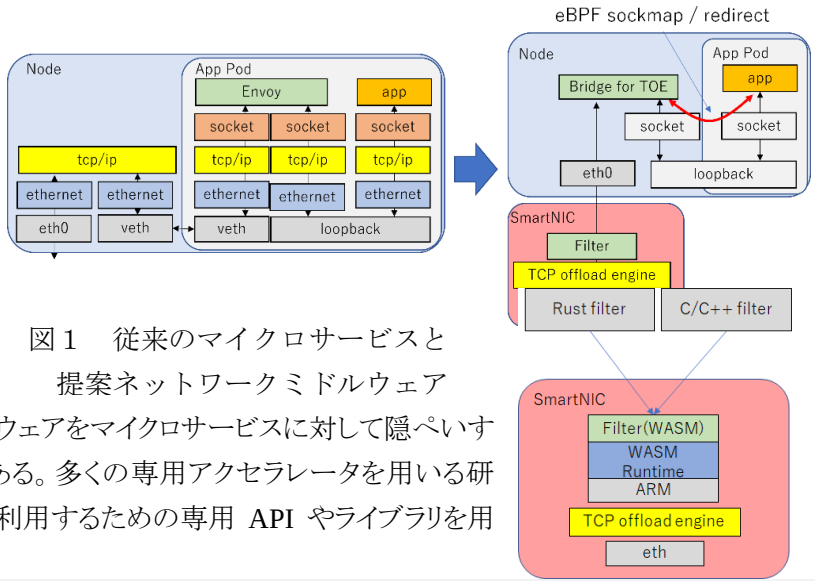


図１　従来のマイクロサービスと提案ネットワークミドルウェア

いてアクセラレータのプログラミングを行う。一方でマイクロサービスでは広く公開されたレジストリサーバーからコンテナイメージをコピーしローカル環境に展開する。コンテナイメージにはコンパイル済みの実行形式のバイナリが含まれており、専用 API やライブラリを用いる方法では実行ファイルの再コンパイルが必要となり、広く公開されたコンテナイメージを利用することができない。そのため、専用ハードウェアを隠ぺいするアブストラクションレーヤーを実現することが重要であった。また、マイクロサービスはアプリケーションと独立してトラフィック処理を実現するサービスメッシュと呼ばれる制御ネットワークがあり、分散環境におけるデバッグやオートスケールといった機能を提供するレイヤがあり、これらの機能も含めてアブストラクションレーヤーを選定する必要があった。そこで、本研究ではサービスメッシュを境界にアブストラクションレーヤーを構築した。さらに、サービスメッシュ処理とホストプロセッサ内で行われているネットワークスタック内の TCP スタックも合わせて SmartNIC にオフロードする方式を実現した。この際、eBPF を用いてソケット通信をハイジャックすることで従来の Socket API を用いたコンテナを変更なしに動作させることに成功した。従来のサービスメッシュと提案手法を図 1 に示す。Envoy はサービスメッシュを実現する Proxy サーバーである。これを、Filter として TCP スタックと合わせて SmartNIC 内にオフロードした。また、Linux カーネルが提供するカーネルをプログラマブルにする eBPF の機能を用いることで、TCP 通信をハイジャックしコンテナの改変なしに SmartNIC を利用できるようにした。これらにより従来ポットが通信する際、4 回 TCP スタックを経由していたが、これを 0 回に抑えることができた。

提案する SmartNIC 環境ではサービスメッシュのプログラミングに関して FPGA に対する深い知識が必要となる。従来のサービスメッシュの Envoy ではサービスメッシュのプロキシ内で WebAssembly を動作させることによって様々な言語でサービスメッシュのプログラミングを可能とする環境を提供している。そこで、SmartNIC 内での L4 以上の処理を簡単化するために、SmartNIC 内の ARM プロセッサ上で WebAssembly ランタイムを実行させ様々な言語から SmartNIC をプログラミングできる環境を実現した。これらを図 2 に示す。

図 2 SmartNIC 内 WebAssembly 機構

SmartNIC を用いたマイクロサービス向けのアクセラレーション

マイクロサービスでは、アプリと独立してネットワークをサポートするサービスメッシュと呼ばれるプログラマブルなネットワークが提供されている。サービスメッシュは分散環境でのトレースや性能評価・A/B テストなどに用いられているが多数のプログラマブルなプロキシやソフトウェアスイッチから構成されており、ネットワーク遅延の増加、CPU 負荷の増加の要因となっている。そこで、これらのサービスメッシュ処理をノード内の TCP スタック処理と合わせて SmartNIC にオフロード（図 3）した。サービスメッシュで多用される分散トレース機能を提案する SmartNIC にオフロードし、Web サーバーコン

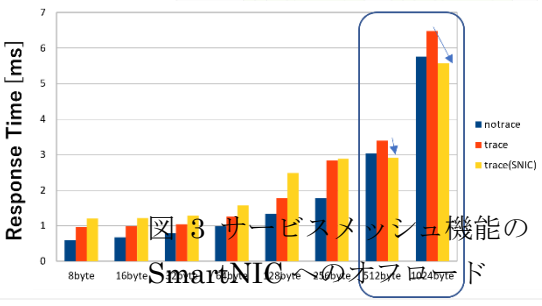


図 3 サービスメッシュ機能の SmartNIC へのオフロード

テナを前述のネットワークミドルウェアを用いて SmartNIC 環境にデプロイし、ペイロードサイズを変えた場合の応答性能の比較を行った。この結果(図 4)、データサイズが 512 バイト以上の際に応答性能を改善することができた。

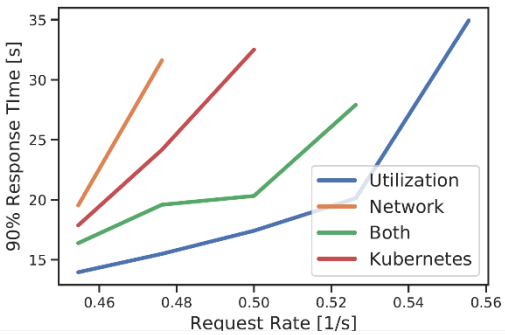


図 4 分散トレース時の応答時間平均

マイクロサービスや FaaS 等のシステムではリクエストの応答分布がロングテールとなることが問題視されている。ロングテールな応答分布はノード内で発生する非同期処理によるノイズやノードごとのリクエスト偏りによって生じるため、全ノードの状況をモニタリングし、リクエストの到着のたびに適切なノード上のサービスを選択するモニタリング・フィードバック型資源管理を行うことによって応答を改善できると考えられる。しかし、従来のソフトウェアによるモニタリングではモニタリング自体のノイズやスケーリング、遅延が問題となる。そこで、大規模ノード環境において低遅延かつ、スケール可能な MQTT broker の開発を行った。MQTT broker の機能を SmartNIC 内に内蔵することでソフトウェアノイズの削減を行った。図 5 は MQTT を用いてデータを収集・分散した際のフィードバック時間の分布を示したものであり、収集から分配までに要した時間を昇順にソートしたものである。この結果、応答性能を大きく改善し、応答時間のばらつきも抑えることができた。他にも非同期通信の高速化[1]を行う研究を行った。

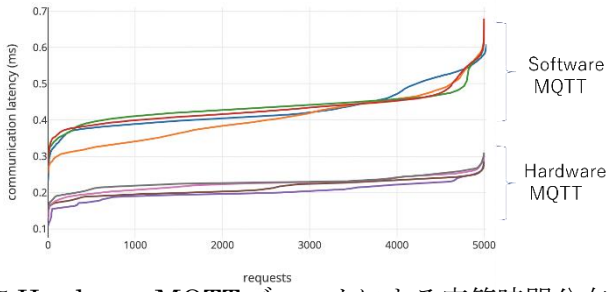
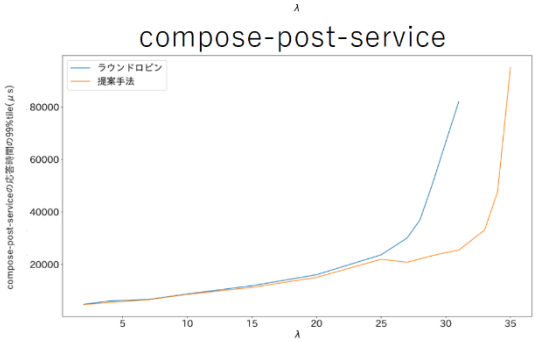


図 5 Hardware MQTT ブローカによる応答時間分布

資源配置最適化によるマイクロサービスの効率化

複数の物理マシンにまたがって動作するマイクロサービスによるアプリケーションでは通信オーバーヘッドが問題となる。物理マシン間通信削減のためマイクロサービスを少数マシンに集約配置すると、リソース使用量に偏りが生じ一部のマシンがボトルネックとなりスループットが低下しうるというトレードオフが存在する。そこで、リソース使用量の複数マシンへの配分と物理マシン間通信の削減を静的に解析した情報を基にアプリケーションに合わせたバランスで行うことのできる配置最適化手法を提案した。本静的な手法により Kubernetes スケジューラと比較して、通信時間の大幅な削減を果たし、スループット・レイテンシの改善に成功した。図 6 では Microservice ベンチマークに対する応答時間を示す。横軸はリクエスト送信頻度を示しており、右に行くほどリクエストが多いことを示している。また、縦軸は応答時間を示している。この結果既存の Kubernetes と比較



図

し、提案手法(Utilization 等)はよりよい応答性能を達成することができている。

前述の静的な解析を基に配置を決定する手法ではトラフィック特性が大きく変動する場合、一部のサービスに負荷が集中し性能が低下する課題があった。また、アプリのデプロイ時に分析を行うオーバーヘッドが生じる課題があった。そこで、デプロイ時にあらかじめサービスのコピーであるレプリカを多数物理ノード上に分散配置し、リクエスト到着時に各物理ノード上の混雑状況を考慮したうえで最適なレプリカを選択する動的レプリカ選択手法を提案した。この結果、提案手法ではより高負荷な場合においても多くのリクエストを処理できることが確認できた。図 7 はリクエスト数（横軸）を増やした場合の平均応答時間(縦軸)の変化を示しており、Kubernetes で用いられる Round-robin 方式と比較して高負荷な場合において応答性能を大きく改善できることができた。

研究領域内外の研究者や産業界との連携

本研究を通して国内クラウド事業者との意見交換を行い、共同研究の開催に向けた準備を進めている。

3. 今後の展開

論文執筆

プリミティブな評価のみで論文を執筆できていないものが多いため、論文の投稿を目指す。実用的な評価を示すことが重要視されているが実用的な評価のためには実装の完成度や評価内容の追加が必要である。そのため、実装の完成度向上、実アプリでの追加評価を進め、論文執筆を行う。

社会実装へ向けた取り組み

実運用されているクラウドシステムにおいて提案した成果物がシステムに組み込まれ有効利用されることを目指す。クラウドシステムは社会に欠かせないものであり、多くの人への貢献につながると考える。実運用されているシステムに導入されるためには 2 種の方法が考えられる。1 つ目は SmartNIC を販売開発しているベンターと協力し製品に組み込んでもらう方法である。もう 1 つはマネージドなクラウドシステムを提供しているクラウドベンダに直接利用してもらう方法である。1 つ目は、Intel や AMD, NVIDIA 等の企業が考えられる。2 つ目は、アマゾン、マイクロソフト、国内であればさくらインターネットなどが考えられる。

いずれの働きかけにおいても成果物を周知してもらうこと、有効な成果物であり容易に導入可能であること、持続的に更新され最低限のサポートが期待できることが重要である。そのために、論文を発表し周知を行う。さらに、それらを OSS として公開し学会併設のチュートリアル等で発表し成果を多くの人に知ってもらう。実装の完成度についても大幅に上げる必要がある。現在は Kubernetes 等との連携が十分でないため連携機能を強化する実装、高機能化、信頼性向上、ツールとしての使いやすさ向上を行う必要がある。さらに、OSS をベースにコミュニティを巻きこむ必要があり Cloud Native Computing Foundation(CNCF)にホスティングされるようなプロジェクトにすることを目指したい。CNCF にホスティングされるようなフレームワークになれば、SmartNIC ベンダやクラウドベンダとの協力も容易になると考えられる。

展開までのスパンは論文公開・追加実装・OSS としての公開がプロジェクト終了 1 年以内、

OSS 界隈での発表が 2 年以内を予定する。2 年目以降は何らかのアクセラレーションプログラムによって資金を調達し開発の大規模化、広告活動が必要である。3 年目以内にマネージドなシステムを提供するクラウドベンダや SmartNIC ベンダにアプローチし製品に組み込まれ活用されることを目指したい。

4. 自己評価

期間を通して多くの論文を投稿することができなかった。理由はコロナや半導体不足による遅れもあるが、シミュレーションと実機の差が大きい。ソフトウェアシミュレータではうまくいくものの、実機環境ではなかなかうまくいかないことが多かった。シミュレータを用いた研究もあるものの、実機を用いた評価が本分野では重要視されているため実機環境での評価を目指した。分散環境の場合、パラメータ数が多く初期実験パラメータを適切に設定することに非常に多くの労力が必要であった。また、実機のための実装もそれなりの実装量、包括的な知識等が必要であり、分散環境特有の再現性の低いデバッグ等にも多くのコストがかかってしまった。また、ソフトウェアが有する最大セッション数等のソフトウェアリミットも問題となった。

一方で、実装した成果物は実機上で正しく動作しおり、プロダクションレベルには至らないものの一定以上の完成度を有している。論文公開後、これらは OSS として公開する予定であり、社会・経済への波及効果は見込まれると考えている。特に、マイクロサービス向けネットワークミドルウェアと SmartNIC によるサービスメッシュのアクセラレーション研究に関しては国内の大手クラウドベンダとの共同での研究を視野に入れた協力体制を構築することができた。前述のクラウドベンダは多くの国内ユーザーが利用しており、提案手法や実装物がデータセンタにて利用されれば、社会や経済への貢献につながる。

5. 主な研究成果リスト

(1) 代表的な論文(原著論文)発表

研究期間累積件数:●件

1. 1. R. Sakamoto, Y. Ezaki, M. Kondo, Hash Distributed A* on an FPGA, International Symposium on Highly-Efficient Accelerators and Reconfigurable Technologies (HEART2022), June, 2022.
マイクロサービスでは多数の物理ノード上のサービス間で非同期な通信が多く行われる。非同期処理の実装はアプリケーションと並行して通信を行う通信処理機構、コールバックによるアプリとの同期機構、多数のバッファ処理から構成されており、非同期処理自体はオーバーヘッドが大きい処理である。従来ソフトウェアで行っていた非同期通信処理をハードウェアによって軽量に行うことで、分散処理を高速に行うための非同期通信機構について検討した。

(2) 特許出願

該当なし

(3) その他の成果(主要な学会発表、受賞、著作物、プレスリリース等)

1. 坂本　龍一, : FPGA SmartNIC によるサービスメッシュの効率化, ACRi ウェビナー, Dec, 2022.
1. 中島創太, 坂本龍一, 中村宏, : 動的スケジューリングによるマイクロサービスの応答性能改善, コンピュータシステム・シンポジウム(ComSys2022), Dec. 2022.
2. 仮屋 郷佑, 坂本 龍一, 中村 宏, :動的スケジューリングによるマイクロサービスの実行最適化, コンピュータシステム・シンポジウム(ComSys2021), Dec. 2021.
3. 横山 遼, 坂本 龍一, 中村 宏 : 通信経路を考慮したマイクロサービスの高速化検討, 研究報告システムソフトウェアとオペレーティング・システム（OS）, 2020-OS-148(9), 1-9 (2020-02-20)
4. 江崎 ゆり子, 坂本 龍一, 近藤 正章 : 経路探索処理向け専用ハードウェアの検討, 研究報告システム・アーキテクチャ（ARC）, 2020-ARC-240(19), 1-8 (2020-02-20)