

## サイドカープロキシ環境下におけるマイクロサービスの遅延時間分析と FPGA SmartNIC による性能改善の検討

河野竜也<sup>1</sup> 坂本龍一<sup>1</sup>

**概要：**マイクロサービスアーキテクチャにおけるサイドカープロキシの導入は、セキュリティやトラフィック管理などの点で重要な役割を果たす。しかし、その一方でプロキシの経由によりアプリケーションの応答時間に遅延が発生する可能性がある。そこで本研究では、サイドカープロキシの有無におけるアプリケーションの性能を比較・分析し、その影響を評価した。さらに、FPGA SmartNIC の導入により性能がどの程度向上するのかを探求する。

**キーワード：**マイクロサービス、サービスメッシュ、SmartNIC

### 1. はじめに

近年、マイクロサービスアーキテクチャは、Netflix や Uber をはじめとする多くの企業で導入が進んでおり、システムの拡張性や保守性などの向上に貢献している。しかしながら、マイクロサービスアーキテクチャは、サービス間の通信が複雑になるなどの課題を抱えている。この課題を解決するための技術として、サービスメッシュが注目されている。サービスメッシュは、サービス間の通信を効率化し、負荷分散、トレーシングなどの機能を提供している。サービスメッシュについてディープラーニングを用いた異常検知[1]など、現在でもさまざまな研究がなされている。

しかしながら、サービスメッシュの導入も良いことばかりではない。サービスメッシュが使用するサイドカーパターンは、通信の経由が増えることから遅延が発生する可能性がある。本研究では、この遅延増加に焦点を当て、マイクロサービスアプリケーションにおけるサイドカープロキシの影響を調査する。具体的には、Bookinfo と呼ばれるマイクロサービスアプリケーションに対し、サイドカープロキシ等をいっさい入れない理想の場合と Envoy プロキシを入れた場合の実行時間を測り、遅延時間を評価する。

また、この遅延問題の解決策として、SmartNIC の活用可能性についても考察する。SmartNIC はネットワーク処理のオフロードで利用される技術であり、これまでも推薦システムの効率化[2]や eBPF を SmartNIC にオフロードする[3]などさまざまな研究でその有効性が示されている。このような効果が認められていながら、これまで SmartNIC をサイドカープロキシのオフロードに適用させるという研究はされてこなかった。そこで本研究では、SmartNIC をマイクロサービスの領域に応用し、サイドカープロキシの処理をオフロードする方法を探求する。これにより、サービスメッシュにおける通信の遅延時間がどれだけ改善されるのかを定量的に評価することを目指す。

### 2. Bookinfo アプリケーション

本研究では、マイクロサービスアーキテクチャの理解と評価のために、具体的なサンプルとして Bookinfo アプリケーションを使用する。図 1 のように Bookinfo は、複数のマイクロサービスから構成されるアプリケーションであり、各サービスは異なるプログラミング言語で実装されている。具体的に Bookinfo アプリケーションは、以下の 4 つのマイクロサービスで構成されている。

- 1.Productpage: 本の詳細やレビューをユーザーに表示するサービス。
- 2.Details: 本の詳細情報を提供するサービス。
- 3.Reviews: 本のレビューを提供するサービス。このサービスは、3つのバージョンが存在し、異なるレビュー結果を返すことができ、サイドカーによってロードバランシングされる。
- 4.Ratings: 製品の評価を提供するサービス。

図 1 には、各サービスに Envoy プロキシが挟まれており、これにより通信の経由が増え、遅延が発生する可能性がある。この Bookinfo アプリケーションを用いてセクション 3 では Envoy を入れた場合と入れない場合の遅延時間の分析を行い、セクション 4 では SmartNIC を入れた場合の性能改善の考察を行う。

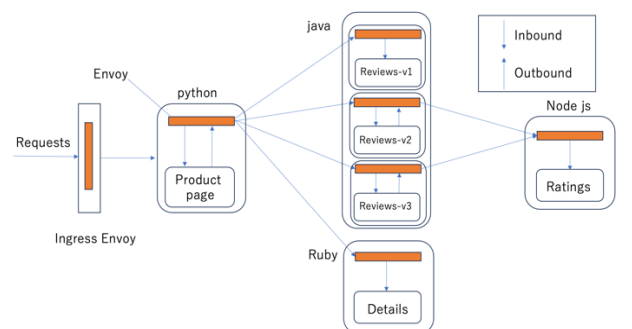


図 1 Envoy を入れた Bookinfo アプリケーション

<sup>1</sup> 東京工業大学  
Tokyo Institute of Technology

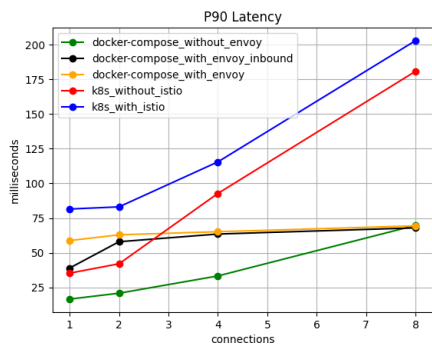


図2 コネクション数を変えた時の90%レイテンシー

### 3. Envoy プロキシによる遅延時間の分析

Bookinfo アプリケーションを用いて Envoy プロキシを入れた場合と入れない場合、また kubernetes で istio を入れた場合と入れない場合の実行時間の差を測定した。図2はその結果である。以下は各要素の説明である。

- 1.docker-compose\_without\_envoy(緑線): docker-compose に envoy を入れない真にアプリのみの状態
- 2.docker-compose\_with\_envoy\_inbound(黒線):docker-compose に inbound だけ envoy を入れたもの
- 3.docker-compose\_with\_envoy(黄線): docker-compose に対して inbound と outbound の両方に envoy を入れたもの
- 4.k8s\_without\_istio(赤線): kubernetes に istio を入れないもの
- 5.k8s\_with\_istio(青線): kubernetes に istio を入れたもの

1 コネクションにおいて理想的な場合の docker-compose (緑線)は16.51msであった。それに対して Envoy を inbound だけ入れたもの (黒線)は38.11ms となり2倍以上の実行時間となり、outbound も入れたもの (黄線)においては58.75ms となり3倍以上の実行時間となった。一方で kubernetes に istio を入れないもの(赤線)では35.24ms、istio を入れたもの(青線)は81.5ms となりこちらもサービスメッシュを入れることによって2倍以上の実行時間となった。コネクション数を8に増やすと理想的な場合(緑線)と inbound に Envoy を入れたもの(黒線)や outbound にも Envoy を入れたもの(黄線)がほぼ同じ実行時間となっているが、これは理想的な場合(緑線)ではロードバランシングができず、一つの Reviews サービス(今回は Reviews-v3 のみ)にリクエストが集中しているからだと考えられる。以上のことから Bookinfo アプリケーションにおいては、Envoy プロキシを入れることによってアプリケーションの性能を大きく下げることが判明した。したがってサイドカーによる遅延を減らすことは、アプリケーションの性能を大幅に向上させることにつながる。

### 4. SmartNIC による性能改善の可能性

サイドカープロキシの遅延を減らすための SmartNIC の可能性について考察する。SmartNIC はネットワーク処理を

オフロードするためによく用いられ、それは Envoy プロキシの処理をオフロードするためにも活用できる可能性がある。本研究では、初期評価として単純にパスを通すだけの SmartNIC のプロキシを用意し、SmartNIC を通さない時と比べた時の性能の差を測定する。具体的には、python の httpserver を用意して SmartNIC を通した時と通さない時の時間の差について curl を用いて測定する。また今回の実験では本研究室で開発を行っている SmartNIC を使用し、この SmartNIC は単一のサービスの Envoy プロキシをオフロードする[4]などの有用性が示されている。

結果は次のようになった。SmartNIC を通さないものは、4.0ms,SmartNIC を通したものは4.1ms となり、0.1ms の差になった。これは、図2で見た Envoy プロキシによる遅延に比べれば微量なものである。したがって、Bookinfo などのマイクロサービスアプリケーションにおいてプロキシのオフロードができれば、相当な実行時間の短縮になることが期待される。今後の研究では、実際にマイクロサービスアプリケーションに SmartNIC を入れて時間を測定することで、性能改善ができることを実証していきたい。

### 5. おわりに

マイクロサービスにおけるサイドカープロキシの遅延について調査した。マイクロサービスのサンプルアプリである Bookinfo において、1 コネクションでは Envoy プロキシを入れることによって2倍以上の実行時間がかかってしまい、大きく性能悪化を招くことがわかった。この遅延を減らすことはアプリケーションのユーザーエクスペリエンスを向上させる上で非常に重要である。また SmartNIC にプロキシの処理をオフロードさせることによってこの遅延を減らせないかについて考察した。

**謝辞** この成果は、国立研究開発法人新エネルギー・産業技術総合開発機構(NEDO)の「ポスト5G情報通信システム基盤強化研究開発事業」(JPNP20017)の委託事業の結果得られたものです。

### 参考文献

- [1] ZHANG, Chenxi, et al. DeepTraLog: Trace-log. combined microservice anomaly detection through graph-based deep learning. In: *Proceedings of the 44th International Conference on Software Engineering*. 2022. p. 623-634.
- [2] JIANG, Wenqi, et al. Fleetrec: Large-scale. Recommendation inference on hybrid gpu-fpga clusters. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2021. p. 3097-3105.
- [3] BRUNELLA, Marco Spaziani, et al. hXDP: Efficient software. packet processing on FPGA NICs. *Communications of the ACM*, 2022, 65.8: 92-100.
- [4] Gongyu Deng et al. Run-time pluggable packet processing on SmartNIC with WebAssembly. SWoPP 2023