

エッジコンピューティングプラットフォームにおける通信遅延を考慮したコンテナデスケジューラ

2019SE077 上野空

指導教員：宮澤元

1 はじめに

IoTの普及によるインターネットに接続するデバイスの増加に伴い、エッジコンピューティングが注目されている。エッジコンピューティングとは、クラウドコンピューティング(クラウド)のサービスや機能をユーザが利用するデバイスの近くのサーバを用いて提供する技術である。エッジコンピューティングを活用することで、多くのデバイスからのサービス処理要求がクラウドへ集中することによる負荷増大を抑制したり、サービス接続時の通信遅延を低減したりすることができる。

エッジコンピューティングを活用するためには、サービスを提供するコンテナをデバイスの近くの計算ノードで動作させるように適切に配置する必要がある。また、デバイスが地理的に移動することによって、デバイスとコンテナが配置された計算ノードとのネットワーク距離が離れてしまう可能性があるため、デバイスに対して低遅延でサービスを提供し続けるためには、一度配置したコンテナを再配置するためにデスケジュールする必要性も生じる。

従来のクラウド環境では、デバイスが移動したとしても計算ノードごとにデバイスとの接続性が変化することはないので、計算ノード構成やコンテナの配置制約が変更されたときだけコンテナを再配置すればよい。一方、エッジコンピューティングでは、デバイスと各計算ノード間との位置関係を常に把握して、適切なタイミングでコンテナ再配置を行う必要がある。特に、デバイスと計算ノード間の通信遅延を基準としてデスケジュールリングを行う場合、複数のデバイスのそれぞれに対して、どの程度の通信遅延によってデスケジュールリングを行うかを定めなければならないが、最適な基準を定めることは難しい。

本研究の目的はエッジコンピューティングプラットフォームにおいて、デバイスと計算ノード間の通信遅延を考慮した適切なデスケジュールリングを行うことである。Kubernetesでは一度ポッドがノードに配置されると、ノード障害などで稼働しているポッドの数が減らない限り、ポッドの再配置は行われず、通信遅延を考慮したスケジューラを実装するには、通信遅延を考慮してポッドの削除を行う仕組みが必要となる。

本研究ではデバイスとアクセス先の計算ノード間の通信遅延を最適化するデスケジュールリングポリシーを提案する。デバイスおよびプラットフォーム上の各計算ノード間の通信遅延を定期的に測定し、デバイスとクラウド上の計算ノード間の通信遅延を基準としてデスケジュールリングの判断を行う。研究課題は、通信遅延を基準にコンテナ削除を

行うデスケジュールリングポリシーの提案と実装の方針についての検討である。

2 関連研究

エッジコンピューティング環境でのスケジューリングには、緊急時の対応やセキュリティ、ユーザのモビリティについてなど様々な課題が挙げられており、さらなる研究の必要がある。本章では、エッジコンピューティングプラットフォームで通信遅延について考慮したスケジューラの関連研究について述べる。

七野は従来のKubernetesでは、通信遅延を考慮してコンテナ配置を行うことができない点に着目し、ノードから削除されたポッドに対して、デバイスと計算ノード間の通信遅延を基にノードへの再スケジュールを行うスケジューラを提案している[1]。この研究では、ポッドとノード間の通信遅延の具体的な測定方法や、配置されているポッドを削除するための方法については示されていない。

古澤らは遅延値に基づく車両向けコンテナアプリケーションのオフローディング先を、車載機と各計算ノード間の応答時間に基づき車載機、エッジサーバ、クラウドの中から動的に切り替えるフレームワークを提案している[2]。この研究では想定されている車載機とエッジサーバ、クラウドの数はそれぞれ一つだけである。また、車載機が複数である場合などについて考慮されていない。

3 提案手法

本研究ではエッジコンピューティングプラットフォームにおける通信遅延を考慮したコンテナデスケジューラを提案する。提案するコンテナデスケジューラの構成図を図1に示す。

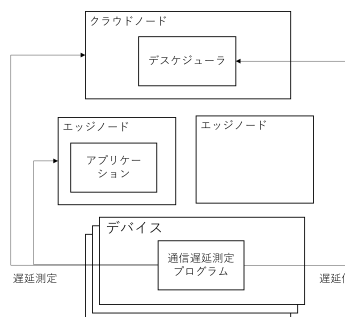


図1 デスケジューラの構成図

クラウドノードにはデスケジューラが配置され、デバイスから送信された通信遅延の測定値を基にコンテナの削除を行う。エッジノードは複数存在し、その内の一つにアプ

リケーションが配置される。デバイスは複数である場合が考えられるので、アプリケーションにアクセスするすべてのデバイスで通信遅延測定プログラムを実行する。

通信遅延測定プログラムは、クラウドノードとアプリケーションが配置されたエッジノードに対して遅延測定を行い、その遅延値をクラウドノードに配置されたデスクレギュラに送信する。デスクレギュラは受信した測定値の平均値を計算し、比較を行うことでより通信遅延の少ないエッジノードにポッドを配置する。比較方法としては、クラウドノードの測定値の平均値が、エッジノードの測定値の平均値を下回る場合に、ポッドを削除し、同じエッジノードに再びアプリケーションが配置されることを防ぐ処理を行う。

4 実装の方針

提案したコンテナデスクレギュラの実現方法について検討する。通信遅延測定プログラムの処理のフローチャートを図2に示す。クラウドノードとアプリケーションの配置されたエッジノードのIPアドレスに向けて、Linuxのユーティリティプログラム ping を実行する処理を繰り返すことで通信遅延の測定を行う。実行結果から遅延値のみをテキストファイルに測定値を書き込み、デスクレギュラに送信する。

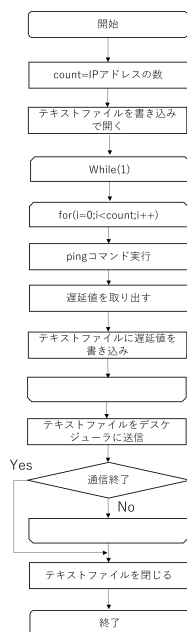


図2 通信遅延測定処理のフローチャート

次にデスクレギュラの処理のフローチャートを図3に示す。提案したデスクレギュラを Descheduler for Kubernetes を拡張して実装する [3]。デスクレギュリングの方法としては、ポッドを削除するための基準である Strategy の RemovePodsViolatingNodeTaints を用いる。この Strategy は有効化することで、ノードにポッドをスケジューリングされないようにする Taint を確認し、それに違反するポッドを削除する。デスクレギュラの処理としては、クラ

ウドノードの測定値の平均値が、エッジノードの測定値の平均値を下回る場合、アプリケーションの配置されたエッジノードに対して Taint を付与する。次に Taint が付与されたノードをクラスタから探索し、ノードに Taint に違反するポッドが存在する場合、ポッドの削除を行う。これにより同じノードに再びポッドが配置されることを防ぐ。

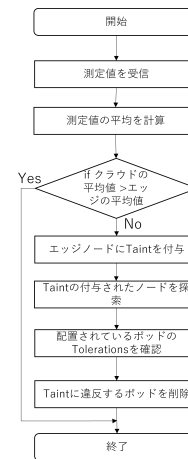


図3 デスクレギュラの処理のフローチャート

5 おわりに

本研究では、エッジコンピューティングプラットフォームにおいて、デバイスと計算ノード間の通信遅延を考慮した適切なデスクレギュリングを行うことを目的に、デバイスとアクセス先計算ノード間の通信遅延を最適化するデスクレギュリングポリシーを提案した。具体的には、デバイスに配置された通信遅延測定ポッドがエッジノードとクラウドノードに対して遅延測定を行う処理の概要と、測定値を基準にポッドの削除を行うデスクレギュリングアルゴリズムを示した。提案したデスクレギュリングポリシーの実装の方針として、Descheduler for Kubernetes を拡張して実現する方法を示した。

今後の展望としては、提案したデスクレギュラを実装し、デスクレギュリングを行う間隔、通信遅延の測定値を適切に利用する Strategy の設定など今回考慮できなかった点について、実験を通して決定する。

6 参考文献

- [1] 七野宏紀: “エッジコンピューティングにおける通信遅延を考慮したコンテナスケジューラ,” 2021 年度南山大学理工学部卒業論文, 2021.
- [2] 古澤 徹 他: “遅延値に基づく車両向けコンテナ型アプリケーションの動的エッジオフローディングの実装と評価”, 情報処理学会論文誌デジタルプラクティス (TDP), Vol.3, No.3, pp.32-43, 2022
- [3] kubernetes-sigs/descheduler, <https://github.com/kubernetes-sigs/descheduler>, (2023/1/13 access).