2D-01

低遅延リアルタイムコミュニケーション向け エッジコンピューティングシステムにおける負荷分散手法の検討

戴 競択† 吉見 真聡‡ 岡田 浩希*

TIS† TIS‡ TIS* & 電気通信大学*

1 はじめに

近年、物体の遠隔操作や高画質 Web 会議などのリアルタイムコミュニケーションの需要が高まっている。特に、ストリーミングデータをクラウド上にあるメディアサーバを通させて、コンピュータービジョンや生配信などの高付加価値のあるサービスを提供する形態が増えている。一方、データがパブリッククラウドを経由することで、遅延の増大をはじめとして様々の問題が生じる。本研究では、メディアサーバをデータソースに近いエッジ側に設置し、コンテナ技術を用いて接続数やトラヒックに応じてメディアサーバを動的に配置するという負荷分散手法を検討する。

2 エッジ側におけるメディア処理

AR/VR(Augmented Reality/Virtual Reality)、クラウドゲーミング、多視点映像配信、１対多／多対多ライブ配信など、サービスプロバイダーはパブリッククラウドでメディア処理サーバを設置し、様々な高付加価値なリアルタイムコミュニケーションサービスを提供する。図１の示すように、動画像、音声をはじめとして、多数のユーザ間の大量なデータがクラウドで処理される必要がある。ユーザ数の増加、サービス種類の増加、データ量の増加に応じて、クラウドでの通信負荷及び計算負荷が指数関数的に増加する。これによって、クラウドでの情報処理速度が遅くなり、レスポンスまでの遅延時間が増大する。

一方、高速・大容量化・多接続を実現する 5G ネットワークの商用化と、NFV（Network Functions Virtualization）技術の応用に伴い、MEC（Multi Access Edge Computing）[1]をはじめとして、RAN（Radio Access Network）などの

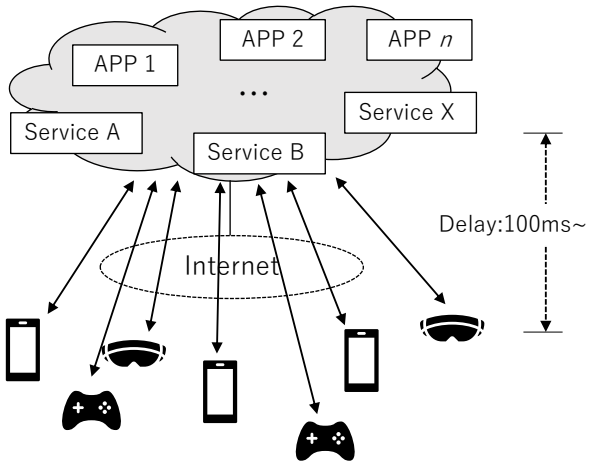

図１ アプリケーション／サービスはクラウド上で実装されたため、アクセスが集中する。

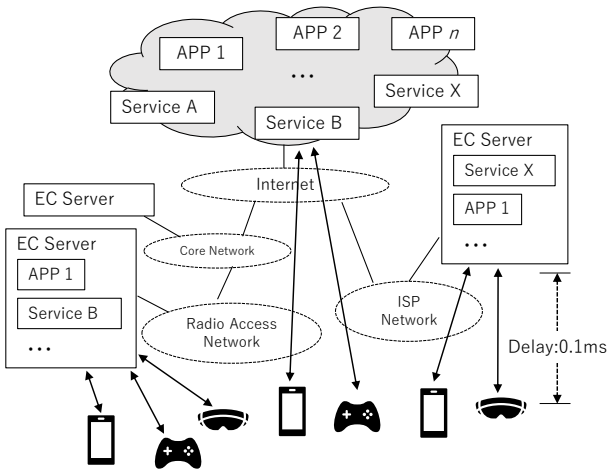

図２ アプリケーション／サービスはEdge Computing Server (EC Server)上でも実装されたため、アクセスが分散する。

A Study on Load Balancing Method in Edge Computing System for Low Latency Real-Time Communications

† DAI Jingze, TIS Inc.

‡ YOSHIMI Masato, TIS Inc.

* OKADA Hiroki, TIS Inc. & The University of Electro-Communications

ユーザに近い場所に設置する汎用サーバ（エッジコンピューティングサーバ）をサービスプロバイダーに提供するというエッジコンピューティングシステムが注目を集めている。図 2 の示すように、エッジコンピューティングの導入により、汎用サーバで実装されたアプリケーション・サービスは低遅延でユーザとコミュニケーションすることができる。そして、クラウドでの通信負荷及び計算負荷も軽減される。

3　コンテナによるエッジでの負荷分散

エッジサーバの計算リソースは一つのサービス・アプリケーションに占用されるわけではなく、オンデマンドで分配するのは一般的である。即ち、サービスリクエストの数やトラヒックの量に応じて、アプリケーション実行環境のスケールアウトを行う。リアルタイムコミュニケーションのようなタイムクリティカルなサービスに対して、必要最小限なリソースを用いて高速に起動し確実に実行できる環境を構築するのは望ましい。本報告では、コンテナ技術を用いてこれを実現する。

エッジでメディア処理を行うビデオ通話サービスを例としてコンテナ技術の活用について説明する。メディア処理として、映像のトランスコーデング、音声のミキシング、そしてコンピュータービジョンも挙げられる。本報告では、Kurento Media Server(KMS)[2]という OSS の WebRTC サーバを用いてメディア処理を行う。

図 3 の示すように、Application Server（AS）は通話（セッション）の初期化に必要なペアリング処理を行うもので、Client A と B を含めて数多くのクライアントが AS でレジストレーションしている。Client A が AS を通して Client B に繋ぐとする。AS が Client A のコールを受信して、既存の KMS（KMS1）がこの新規セッションをハンドリングできるかどうかを既定のロジックで判断する。具体的には、ローカルの CPU・メモリ・ネットワークの使用量、既存セッション数、あるいは既存セッションの WebRTC ストリームの平均遅延、ジッタ、パケットロスなどの統計情報を用いて、新規セッションの通信品質が条件満たせるかどうかを予測する。新

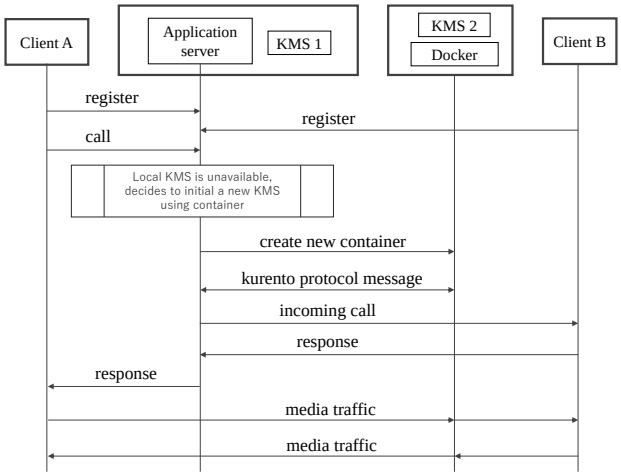

図3　メディア処理サーバのコンテナを使用することで負荷分散を実現する。

規セッションをハンドリングできないと判定したら、AS は他のコンピュータで新しい KMS のコンテナ（KMS2）を起動する。そして、Kurento Protocol Message を使って KMS2 での新規セッションの設定を行う。設定が完了した後で AS は Client B に着信通知を送る。Client B からの承諾を受けた後で Client A に通知することで、セッションの初期化を完了する。続いて、Client A と B がストリーミングデータを KMS2 に送信し、KMS2 から処理済みのデータを受信することで両者間のリアルタイムコミュニケーションを実現する。

4　まとめ

本報告では下記 2 点について検討した。

1．メディア処理サーバをエッジ側で導入することで、通信の低遅延性を実現する。

2．負荷に応じてコンテナ化されたメディア処理サーバをエッジで増減することで、通信品質の安定性及リソース利用の効率性も得られる。

参考文献

[1] ETSI - Multi-access Edge Computing - Standards for MEC, https://www.etsi.org/technologies/multi-access-edge-computing

[2] What's Kurento - Kurento https://www.kurento.org/whats-kurento