OSAKA UNIVERSITY

エッジコンピューティング環境における サービス機能の配置がユーザの通信品質に 与える効果の評価

大阪大学 大学院情報科学研究科
金田 純一

IN 研究会

2017/9/8

研究の背景

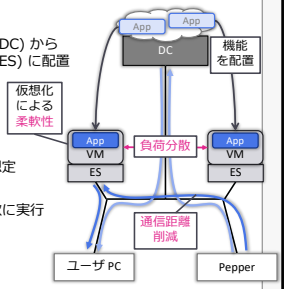
- **端末技術の発展**
  - 通信機能を持つエンド端末がセンサーやカメラを搭載
- **新たなアプリケーションの増加**
  - エンド端末のセンサーやカメラが取得した情報を遠隔地のデータセンタで処理
  - 結果をエンド端末へ転送しユーザへ提示
  - AR (Augment Reality) に代表されるリアルタイム性の高いサービスの増加

↓

- **データセンタを利用するサービスの遅延が増大・顕著化**
  - 距離による遅延
    - 遠隔地に位置するデータセンタまでの通信遅延
  - 負荷の集中による処理遅延
    - 多数のエンド端末からデータセンタへ負荷が集中
  - 遅延の顕著化
    - リアルタイム性の高いサービスの増加

エッジコンピューティング (EC) への期待

- **応答性の向上**
  - サービス機能の一部を、データセンタ (DC) から エンド端末に近い多数のエッジサーバ (ES) に配置
    - 通信距離の削減
    - 負荷の分散
- **柔軟なサービス提供**
  - 仮想化環境上でサービス機能の実行を想定
  - サービス機能の処理拠点を柔軟に配置
  - 通信距離の削減や処理の分散をより柔軟に実行
  - 要求に応じた資源の増強が可能

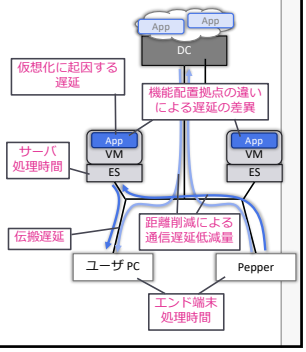


EC: Mobile Edge Computing
VM: Virtual Machine

EC 環境におけるユーザの通信品質

- **応答性に対する期待と懸念**
  - 通信距離削減による通信遅延の低減
  - 負荷分散による遅延の低減
  - 仮想マシンのソフトウェア動作に起因する遅延の増大
- **機能を配置する拠点の違いによる応答性への影響が未知**
  - 遅延の発生要因や発生量
  - 拠点の違いによる遅延の差異

今後の EC の導入や展開に向け EC 環境で生じる遅延の **発生要因**や**発生量**の理解が重要



研究の目的とアプローチ

EC 環境におけるサービス機能の配置がユーザの通信品質に与える効果の解明

↑ アプローチ

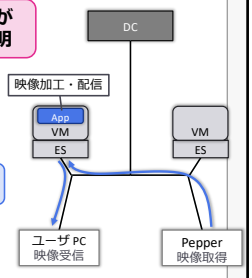
実験室内で実機を用いて EC 環境を構築

＋

EC 環境で生じるアプリケーション遅延を測定

手順

1. OpenStack を用いた EC 環境の構築
2. Pepper を用いた EC サービスの作成
3. サービス機能の配置拠点を変更しアプリケーション遅延を測定



作成した EC サービスの概要と サービス機能の配置例の 1 つ

エッジコンピューティング (EC) 環境の構築手順

EC 環境

↑

ネットワーク機能仮想化環境 (NFV)

↑

OpenStack

ネットワーク機能仮想化 (NFV) を応用して EC 環境を実現
- ネットワーク機能ではなく、サービスアプリケーションを仮想化し実行
- EC を想定しネットワークを構成

ネットワーク機能仮想化 (NFV) 環境の構築に OpenStack を利用
- OpenStack は NFV フレームワークにおける NFVI、VIM として機能

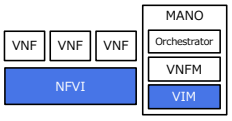
NFV: Network Functions Virtualization
NFVI: NFV Infrastructure
VIM: Virtualized Infrastructure Manager

OSAKA UNIVERSITY 7

OpenStack を利用した NFV 環境の構築

- **OpenStack による仮想化環境を NFVI として利用**
  - NFVI: NFV を実現するインフラ部分
  - 実機ではコンピュートノードとして実現
- **OpenStack の仮想化環境管理機能を VIM として利用**
  - VIM: NFV 全体の管理・調整を行う MANO の一部で、NFVI を管理
  - 実機ではコントローラノードとして実現

OpenStack
- IaaS 型のクラウド環境を構築するオープンソースソフトウェア群
- 様々な機能を有するモジュールを組み合わせてシステムを構築 (モジュラーアーキテクチャ)
- 近年様々な NFV オープンソース実装が OpenStack を NFVI、VIM として採用



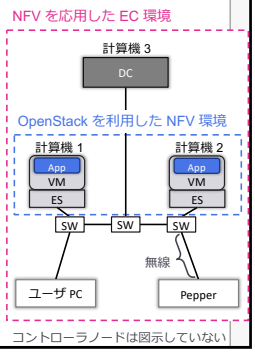
NFV Framework

MANO: Management and Orchestration
VNF: Virtual Network Function
VNFM: VNF Manager

OSAKA UNIVERSITY 8

NFV を応用した EC 環境の構築

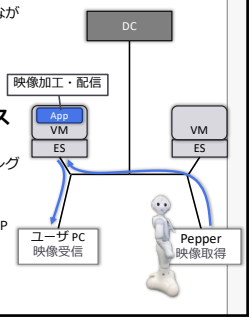
- **サービスアプリケーションを仮想化**
  - 仮想マシン上で、ネットワーク機能ではなくサービスアプリケーションを実行
- **EC を想定しネットワークを構成**
  - ユーザ PC、Pepper および 3 台の計算機をスイッチ (SW) で接続
  - 計算機1、計算機2
    - エッジサーバ (ES) として動作
    - ユーザ側と Pepper 側、両拠点に用意
    - OpenStack コンピュートノードで実現
    - 仮想化されたサーバ
  - 計算機3
    - データセンタ (DC) として動作
    - 非仮想化サーバ
    - 実世界における通信距離は再現していない



OSAKA UNIVERSITY 9

Pepper を用いた EC サービスの作成

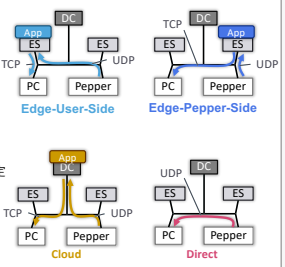
- **買い物代行サービスを想定**
  - Pepper が実店舗に赴き、ユーザは自宅にいながら買い物が楽しめるサービス
  - センシング情報を活用
  - AR 技術を用いて映像に商品情報を表示
- **映像のライブストリーミングサービスのみを実装**
  - Pepper が取得した映像をライブストリーミング
  - エッジサーバで映像に文字列を挿入
  - 映像配信サーバアプリケーションで配信
    - 映像配信サーバアプリケーションは映像を UDP で受信し TCP で送信
  - ユーザ PC で受信



OSAKA UNIVERSITY 10

サービス提供形態とサービス機能の配置

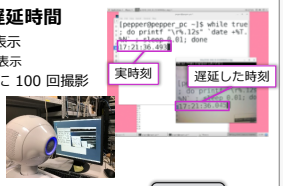
- **EC シナリオ**
  - **シナリオ Edge-User-Side**
    - ES の利用を想定
    - TCP 通信の通信距離が 2
  - **シナリオ Edge-Pepper-Side**
    - ES の利用を想定
    - TCP 通信の通信距離が 4
- **非 EC シナリオ**
  - **シナリオ Cloud**
    - クラウドを提供する DC の利用を想定
    - 非仮想化サーバを利用
    - TCP 通信の通信距離が 3
- **比較シナリオ**
  - **シナリオ Direct**
    - Pepper から PC へ直接映像を配信
    - エンド端末の処理時間を計測



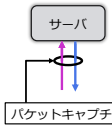
OSAKA UNIVERSITY 11

遅延測定方法

- **ライブストリーミング映像の遅延時間**
  1. Pepper の前でミリ秒単位の時計を表示
     - 時刻同期の観点からユーザ PC 上に表示
  2. 時計と配信映像を並べて、1 秒おきに 100 回撮影
  3. 実時刻と遅延した時刻の差を算出
  4. 平均値を算出

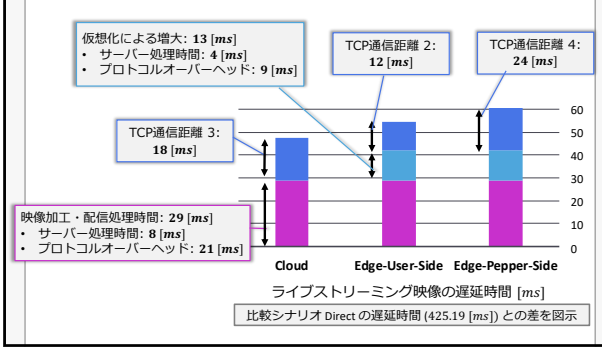


- **サーバ処理時間**
  1. 計算機の出入口となる地点でパケットをキャプチャ
  2. ストリーミングの開始・停止操作を 10 回実行
  3. 各ストリーミングにおける最初と最後のパケットについて、キャプチャ地点の通過時刻の差を算出
     - データはサーバで処理されるため、パケットの運ぶデータの同一性は通過順以外で特定不可能
  4. 平均値を算出



OSAKA UNIVERSITY 12

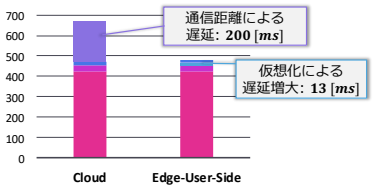
ライブストリーミング映像の遅延時間と内訳



OSAKA UNIVERSITY 13

クラウド環境における通信遅延

- **データセンタまでの通信遅延は 100 $[ms]$ 単位で発生**
  - 通信距離による最大遅延: 200 $[ms]$ と仮定
  - 仮想化による遅延増大: 13 $[ms]$



データセンタまでの通信遅延 [1]
- 日本国内: 100 $[ms]$ 以下
- 米国まで: 100 $[ms]$
- 欧州まで: 200 $[ms]$
- ping の実行結果 (ICMP パケット 1000 回送信の平均)
  - amazon.com: 207.041 $[ms]$
  - dropbox.com: 123.616 $[ms]$

データセンタからエッジサーバへサービス機能を配置することで遅延を**最大約 30%** 低減可能

[1] 田中裕之, 高橋紀之, 川村龍太郎, "IoT 時代を拓くエッジコンピューティングの研究開発," NTT 技術ジャーナル, pp. 59-63, Aug. 2015.

OSAKA UNIVERSITY 14

まとめと今後の課題

- **エッジサーバを利用したサービス提供の有効性を定量的に確認**
  - 仮想化による遅延増大: 13 $[ms]$
  - データセンタまでの通信距離削減による遅延低減: 最大 30%
  - ただし各シナリオにおけるアプリケーション遅延に占める割合は低い
    - 遅延の大部分 (400 $[ms]$ 程度) は Pepper のエンド端末処理時間
    - 現行 Pepper の搭載 CPU の FLOPS は Intel Core i7 の約 10 分の 1
    - 同性能の CPU が搭載されると、エンド端末処理時間は約 40 $[ms]$ まで低減
  - 端末性能の改善を前提とすれば、エッジサーバを利用したサービス提供は効果がある
- **今後の課題**
  - 仮想マシンのライブマイグレーションによるサービス機能の**動的な再配置**がユーザの通信品質に与える影響・効果の調査