

# エッジコンピューティング環境における局所的情報を用いた モバイル端末へのリソース割当て手法

谷玲治<sup>1</sup> 福永昂輝<sup>1</sup> 杉村侑起<sup>1</sup> 横山和俊<sup>1</sup>

概要：本稿では、モバイルエッジコンピューティング環境において、各 MEC サーバが局所的情報を用いてモバイル端末の割当てを決定する分散型のアルゴリズムを提案する。提案手法について、岡山駅周辺を対象に交通シミュレータ SUMO で走らせた車両をモバイル端末と扱い、OpencellId から取得した岡山駅周辺の KDDI 基地局配置情報を元に MEC サーバを配置して、リソース割当てシミュレーションを行った。その結果、最近傍・最小リソース量による割当ては、局所的情報のみで全モバイル端末を割当て可能であることが示された。

## 1. はじめに

モバイルエッジコンピューティングにおいて一部のエッジサーバ（以下 MEC サーバと略す）に負荷が集中するとリソース不足が発生する可能性がある。この課題に過去研究では、モバイル端末と通信している MEC サーバの近傍にある MEC サーバを利用する集中アルゴリズムを提案した [1]。しかし、全てのモバイル端末のリソース割当てを一括管理するのは現実的ではない。本研究では、各 MEC サーバが局所的情報を用いて割当てを決定する分散型のアルゴリズムを提案し、有効性を評価する。

## 2. システムモデル

システムモデルを図 1 に示す。図 1 では、基地局とその基地局の情報を集約する集約局、集約局の情報をデータセンタへ送信する中間局が仲介となり、基地局間やデータセンタ（以下 DC と略す）との通信が実現される。以下、システムモデルについて説明する。

### (1) MEC サーバ

図 1 の各基地局に MEC サーバを配置し、 $m_i$  で表す。各 MEC サーバ  $m_i$  には割当て可能リソース量  $R_i$  が与えられる。自身の  $R_i$  を超えない限り、モバイル端末のリソース要求を処理できる。

### (2) モバイル端末

モバイル端末は道路ネットワーク上を走行する。モバイル端末を  $d_j$  で表し、各モバイル端末  $d_j$  には通信している MEC サーバに要求するリソース量が設定され

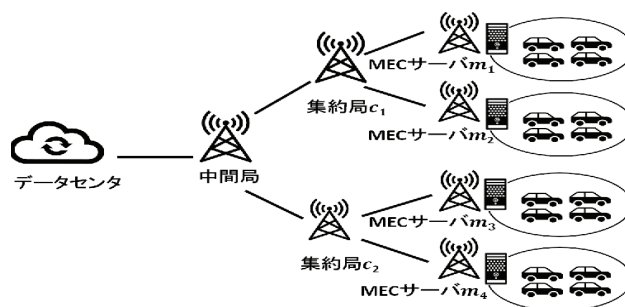


図 1 システムモデル

ている。モバイル端末は道路ネットワーク上で、最も近い基地局と通信する。また、モバイル端末は走行履歴を持っており、直近  $t$  秒間の移動距離  $s$ 、平均方位角  $\alpha$  がわかるとする。

### (3) 周辺 MEC サーバ

各基地局の MEC サーバ  $m_i$  は、半径  $\ell$ .km 以内の周辺の MEC サーバ  $m_n$  の情報を得ることができる（以下周辺 MEC サーバ  $m_n$  と呼ぶ）。また、各 MEC サーバ  $m_i$  は通信しているモバイル端末の全ての要求リソース量  $S_i$  がわかり、問い合わせることで半径  $\ell$ .km 以内の周辺 MEC サーバ  $m_n$  の  $S_n$  を知ることができる。また、MEC サーバと周辺 MEC サーバの距離は近いいため、情報伝達にかかる遅延がないものとする。

### (4) 割当て

割当ては一定時間ごとに各 MEC サーバ  $m_i$  が判断する。また MEC サーバ  $m_i$  が周辺 MEC サーバ  $m_n$  に対して割り当てたい場合は、対象の周辺 MEC サーバ  $m_n$  に対して割当て依頼を送信する。割当て依頼を受

<sup>1</sup> 高知工科大学 情報学群

表 1 実験パラメータ

シミュレーション時間	1000 sec
基地局の通信範囲	500 m
モバイル端末の要求リソース	50 ~ 500 のランダム
割当て可能リソース量	2000, 3000, 4000
割当て時間	10 sec

信した周辺 MEC サーバ  $m_n$  は自身の  $R_n$  を超えない場合は割り当てるが、超えた場合に割当て失敗として DC に割当て依頼を送信する。

(5) 遅延時間

システムモデルでの遅延時間をホップ数で表す。モバイル端末のリソース要求を割り当てる場合、様々な転送・中継設備を経由するほど通信遅延が発生する。本稿では、1 つの転送・中継設備を経由することを 1 ホップとする。

3. リソース割り当て手法

モバイル端末のリソース要求をどの MEC サーバに割り当てるべきかを考える。割当て手法は平均ホップ数を抑えることを目的とする。平均ホップ数はホップ数の総数に対して割当ての総回数を割ったものとする。以下では、評価する 2 つの手法について述べる。

(1) 最近傍割当て

各 MEC サーバ  $m_i$  は通信しているモバイル端末が自身に割当て可能か判断し、割当て不可の場合に物理的な距離が最も近い周辺 MEC サーバ  $m_n$  に対して割当て依頼を送信する。

(2) 最近傍・最小リソース量による割当て

各 MEC サーバ  $m_i$  は通信しているモバイル端末が自身に割当て可能か判断し、割当て不可の場合、周辺 MEC サーバ  $m_n$  で  $S_n$  と  $m_i$  から  $m_n$  へ割当て依頼済みリソース量  $Ar_{in}$  の和が最小の MEC サーバに対して割当て依頼を送信する。

4. 評価

4.1 評価内容

OpenCellID で岡山駅周辺の KDDI の基地局配置を取得し、各基地局に MEC サーバを配置する。また OpenStreetMap で岡山駅周辺の道路ネットワークを取得し、交通シミュレータ SUMO で取得した道路ネットワーク上を走行する車両を生成し、交通シミュレーションを実行した。本研究では、交通シミュレータ SUMO で走らせた車両を取得し、モバイル端末として扱った。表 1 にシミュレータでの実験パラメータを示す。評価項目としては、平均ホップ数と割当て失敗率とする。割当て失敗率は、割当て失敗回数を全ての割当て回数で割ったものとする。各手法で割当て可能リソース量を 2000, 3000, 4000 と変化させ評価を行う。

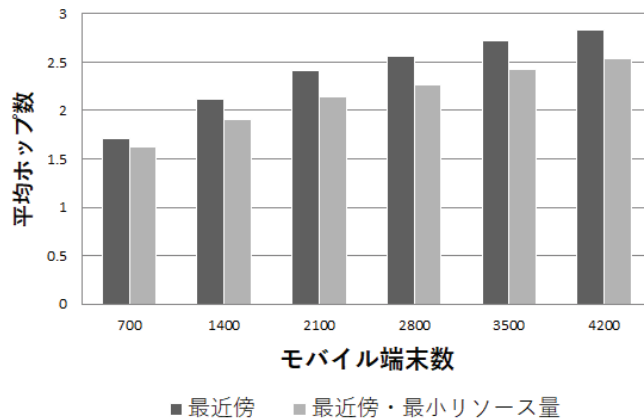


図 2 平均ホップ数 ( $R_n = 2000$  のとき)

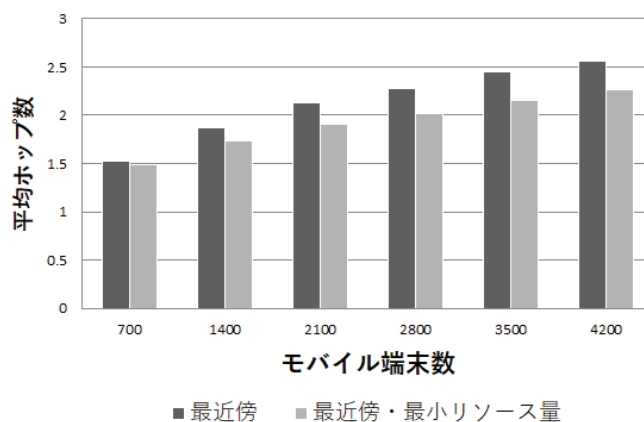


図 3 平均ホップ数 ( $R_n = 3000$  のとき)

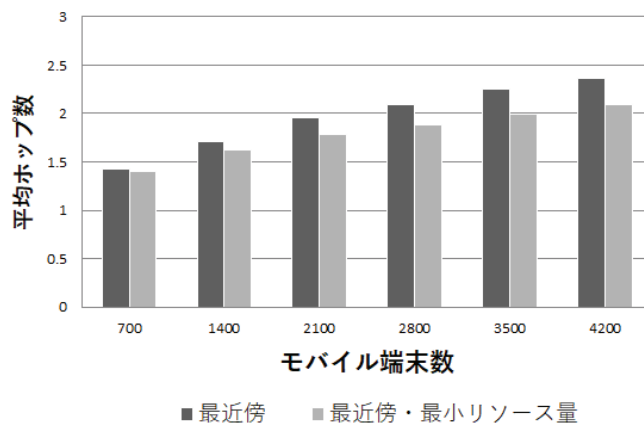


図 4 平均ホップ数 ( $R_n = 4000$  のとき)

4.2 評価結果

図 2,3,4 に各割当て手法の平均ホップ数の結果を、図 5,6,7 に割当て失敗率を示す。平均ホップ数については、両手法ともに割当て可能リソース量を増やすことによって減少した。両手法を比較した場合、常に最近傍・最小リソース量による割当ての平均ホップ数が低いものとなった。またモバイル端末数が少ない場合はその差は小さいがモバイル端末

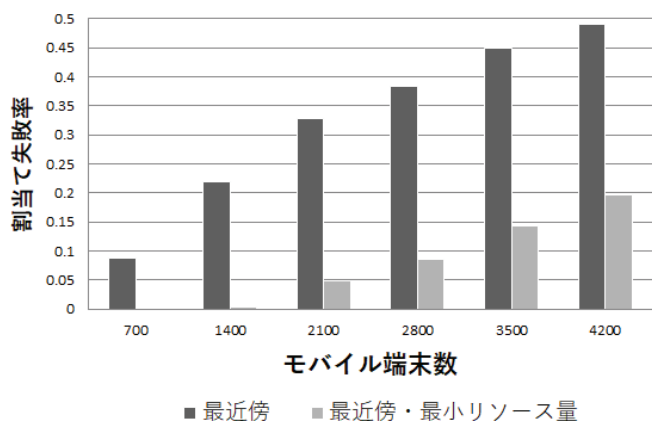


図 5 割当て失敗率 ( $R_n = 2000$  のとき)

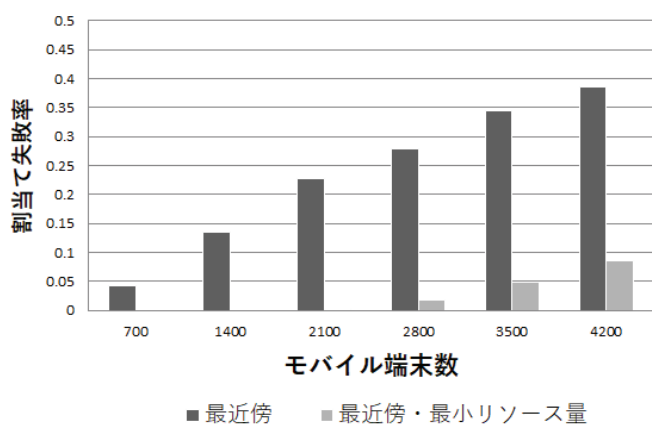


図 6 割当て失敗率 ( $R_n = 3000$  のとき)

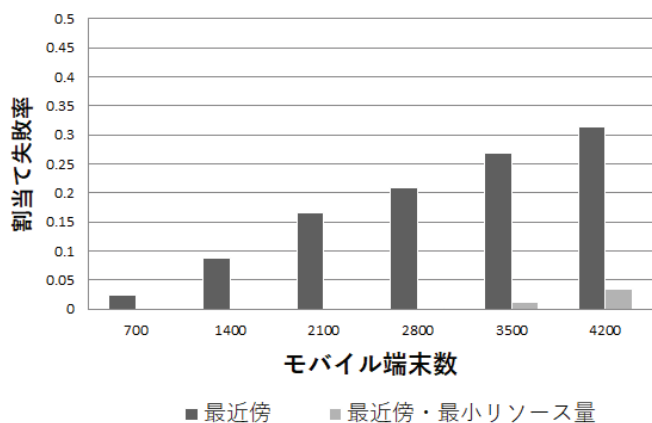


図 7 割当て失敗率 ( $R_n = 4000$  のとき)

が増えていくと差が大きくなり、モバイル端末が一定数を超えると差の増減がなくなった。割当て失敗率については、両手法ともに割当て可能リソース量を増やすことによって減少した。両手法を比較した場合、常に最近傍・最小リソース量による割当ての割当て失敗率が低いものとなった。また最近傍・最小リソース量において、割当て可能リソース量が 2000 の場合、モバイル端末が 1400 以下、3000 の場合、

2100 以下、4000 の場合、2800 以下で割当て失敗率が 0 となった。これは局所的情報のみで全モバイル端末を割当て可能であったことを表している。

## 5. おわりに

本稿では、モバイル端末のリソース要求を分散アルゴリズムで割当てを決定する手法を提案し、シミュレーションにより有効性を示した。

## 参考文献

- [1] 大崎康平, 福永昂輝, 横山和俊, Cloutlet 環境における移動経路計画を用いたリソース割り当て手法の検討, 情報処理学会研究報告, Vol.2017-DPS-173, No.8, pp1-8(2017).