

AP 動作環境の制約を考慮した Cloudlet 環境における リソース割り当て手法

福永 昂輝[†] 大崎 康平[†] 横山 和俊[†]

高知工科大学 情報学群[†]

1. はじめに

近年、クラウド環境の普及に伴い、車載端末や IoT 機器のようなモバイル機器の処理をデータセンタのサーバで実行する試みが行われている。しかし、モバイル機器が要求する処理はリアルタイム性が求められるものがあり、その場合、モバイル機器とサーバ間の通信遅延が問題となる。そこでネットワーク上に Cloudlet と呼ばれる計算機を配置し、サーバ負荷の軽減や応答時間の改善を行う研究がなされている。

しかし、Cloudlet は、データセンタのように豊富なリソースを持たないため、負荷が集中するとリソース不足になる問題がある。本稿では、この問題に対し、モバイル機器が利用するアプリケーション（以降、AP と略す）がどの Cloudlet 環境で動作可能かを考慮した上で、モバイル機器の位置から許される距離の近傍 Cloudlet を利用し、割り当てることでリソースの枯渇を抑えること、モバイル機器と Cloudlet 間の通信遅延を小さくすることを目的とするリソース割り当てアルゴリズムの提案を行う。

2. 環境モデル

Cloudlet 環境は、インターネット上にあるデータセンタと Cloudlet サーバ群、モバイル機器群の三層で構成されており、Cloudlet はそれぞれ実行プラットフォームをもつ。各 Cloudlet サーバは実行環境をデータセンタからダウンロードし、モバイル機器は無線通信を用いて通信を行う。

(1) Cloudlet サーバとモバイル機器

$n \times m$ の格子状のセルの 1 つ 1 つに Cloudlet サーバが配置される。それぞれの Cloudlet サーバを c_n で表す。Cloudlet サーバには実行可能 AP、割り当て可能リソース量が与えられ、モバイル機の要求により、様々な AP が動作する。しかし、モバイル機器が要求する AP が実行できない Cloudlet サーバがある。

モバイル機器を d_n で表し、AP 実行に必要なリソース量が定義される。これはモバイル機器の固有値として定義される。またモバイル機器は 1 単位時間に 1 回、上下左右に移動する。

(2) 最大割り当て距離(最大通信遅延)

モバイル機器と Cloudlet サーバ間の通信において、通信遅延は距離に比例するため、代替指標とし

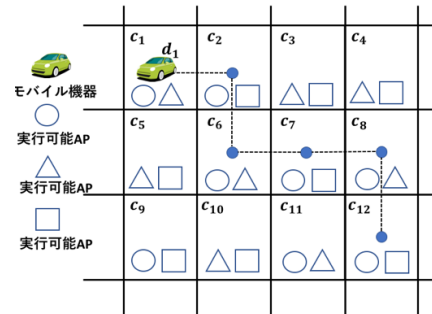


図1 Cloudlet サーバモデル

で割り当て距離を利用する。本稿では、モバイル機器と同じセルの Cloudlet サーバを割り当て距離 0 (最寄り Cloudlet サーバと呼ぶ) とすると、上下左右の Cloudlet サーバの割り当て距離は 1、斜めに位置する Cloudlet サーバの割り当て距離は 2 となる。また、最大割り当て距離とは、モバイル機器と割り当てられた Cloudlet サーバの間の最大距離のことを示す。

3. リソース割り当て手法

本稿では、モバイル機器の割り当て順序を変えることで、リソースの枯渇と遅延時間を小さくする。

3.1. 従来手法

従来手法として、参考文献[1]より混雑度順割り当て手法を使用する。混雑度順は、各 Cloudlet サーバにおける混雑度という指標を算出し、混雑度が大きい順にモバイル機器をソートする割り当て手法である。混雑度とは、指定した距離以内のすべてのモバイル機器の要求リソース量の和をその Cloudlet サーバの所有リソースで割ることで求められる。手法の実行例として表 1 に入力するモバイル機器の例、図 2 に Cloudlet サーバの配置例とリソース状況を示す。この時、調べる距離を 1、各 Cloudlet サーバのリソースを 5 とした場合の c_2 の混雑度を求める。 c_2 から距離 1 以内の Cloudlet サーバ c_1, c_2, c_3, c_5 のうち、要求しているモバイル機器は d_1, d_2 で総要求リソースは 6 となる。これを Cloudlet サーバの所有リソース 5 で割ると 1.2 となり、この数値が混雑度となる。

本稿では、Cloudlet サーバの実行プラットフォームを考慮する。従来手法では、AP を実行することができる Cloudlet サーバの制約を考慮していないため、遅延時間が増加することがある。このことを例を使用して説明する。従来手法では、混雑度が高く、モバイル機器周辺 Cloudlet サーバに AP1 を実行できるプラットフォームが少ない場合、AP1 を

Resource allocation method in Cloudlet environment considering constraint of AP execution environment

Koki Fukunaga[†], Kohei Osaki[†], kazutoshi Yokoyama[†],

[†]School of Information, Kochi University Technology

モバイル機器名	最寄 Cloudlet	要求リソース	使用 AP
d_1	c_2	3	AP1
d_2	c_5	3	AP1
d_3	c_6	5	AP3

表 1 入力データの例

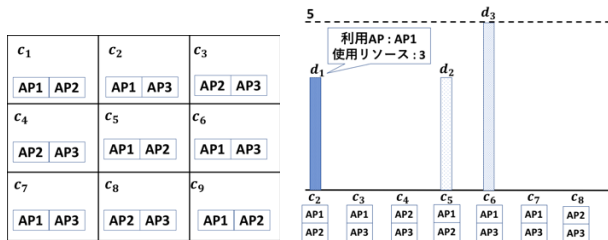


図 2 Cloudlet サーバの配置例とリソース状況

割り当てるより前に AP2, AP3 を使用するモバイル機器が割り当てられてしまう可能性がある。AP1 はより遠くの Cloudlet に割り当られることになるため、遅延時間が増えてしまう。

3.2. 提案手法

従来手法の問題を解決するために、AP 動作環境を考慮した優先度をモバイル機器に設定し、優先度が高い順に処理していくことで、効率的な割り当てを実現する。優先度はモバイル機器が利用する AP ごとの混雑度を、処理するモバイル機器の最寄り Cloudlet から指定した距離内の実行可能 Cloudlet サーバの数で割ったもので求める。モバイル機器が利用する AP ごとの混雑度は、指定した距離内の同じ AP を持つモバイル機器の要求リソース量の和で求められる。この優先度はモバイル機器が利用する AP ごとの混雑度に対して周辺に対応した実行環境を持つ Cloudlet サーバが少ないほど高い数値になるため、3.1. 従来手法 で述べたある AP を利用するモバイル機器がより遠くに割り当てられるという問題点を改善することができる。表 1, 図 2 を例に、調べる距離を 1 とした場合の d_2 の優先度を示す。 d_2 の最寄り Cloudlet サーバ c_5 から距離 1 以内の Cloudlet サーバの内 d_2 が使用する AP1 が割り当て可能であるのは c_2, c_5, c_6 なので実行可能 Cloudlet サーバ数は 3, また、モバイル機器が利用する AP ごとの混雑度は AP1 を利用するモバイル機器 d_1, d_2 の要求リソースの総数なので 6 となる。これらよりモバイル機器に設定する優先度は、AP ごとの混雑度 6 を実行可能 Cloudlet サーバ数 3 で割るので 2 となる。

4. 評価と考察

本稿では、入力データとして、 30×30 の Cloudlet の集合体を用意し、Cloudlet がもつリソース量は一律 200 とした。入力するモバイル機器は 30×30 の正方形の各辺の中央付近の 3 点から移動を

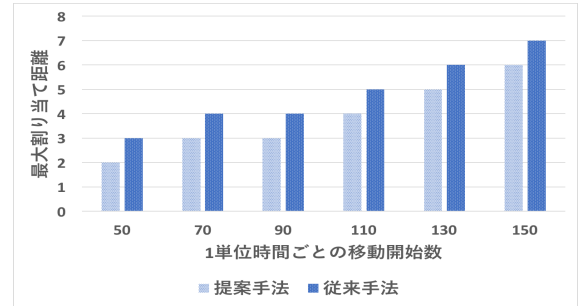


図 3 シミュレーション結果

開始し、対辺の 3 点に直進する。モバイル機器は、AP1, AP2, AP3 のどれかを利用する。AP は 1 から 5 の乱数で決定された数が AP のリソース量となる。偏りを設けるため、AP の比率は AP1:AP2:AP3 = 1:5:7 とする。モバイル機器数は 1 単位時間あたりの移動開始数を変え、30 単位時間でのシミュレーションを行い、6 つのデータを評価する。

従来手法、提案手法に関してシミュレーションを用いて、それぞれの最大割り当て距離を計測した。シミュレーション結果を図 3 に示す。グラフ縦軸は各手法での最も大きかった割り当て距離を示している。グラフ横軸は、1 単位時間でモバイル機器が移動開始する数を示している。結果より、モバイル機器が 1 単位時間に移動開始する数が増加するとともに、最大割り当て距離が増加しているのがわかる。また、グラフ中の提案手法、従来手法を比較すると、いずれのパターンにしても最大割り当て距離が 1 少なくなっていることがわかる。これは、モバイル機器が集中している場所において、実行可能 Cloudlet サーバが少ない AP から割り当てるため、混雑した状況においても実行環境がなく、割り当てられないという最悪を防げたためと考えられる。また、1 単位時間ごとの移動開始数が 50, 70 の時、同じ結果であるのは、AP の要求リソース量が乱数であるため 50 の時よりも 70 の時の方が AP の要求リソース量が小さいためと考えられる。

5. まとめ

ネットワーク上に小規模リソース Cloudlet を配置し、モバイル機器と通信させることで、通信遅延問題の改善することが求められている。しかし、Cloudlet はリソース不足の問題があり、改善方法として、近傍の Cloudlet を利用し、リソース不足の改善や通信遅延を小さくするリソース割り当てアルゴリズムの提案を行い、シミュレーションを行った。

【参考文献】

[1] 大崎康平, 福永昂輝, 横山和俊, “Cloudlet 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, pp. 1-8 (2018).