

平成 29 年度

修士学位論文

Cloudlet 環境における
移動経路計画を用いたリソース割当て手法

Resource Allocation Algorithm
using The Moving Path Planning in Cloudlet
Environment

1205078 大崎 康平

指導教員 横山 和俊

平成 30 年 2 月 28 日

高知工科大学大学院 工学研究科 基盤工学専攻
情報システム工学コース

要旨

Cloudlet 環境における 移動経路計画を用いたリソース割当て手法

大崎 康平

近年, Cloudlet と呼ばれる小規模なコンピューティングリソースをネットワーク上に配置し, 分散処理を行うことが提案されている. しかし, Cloudlet はデータセンタと比較すると大きなリソースを持たせることが難しい. そのため, 頻繁に移動を繰り返すモバイル機器を対象とした環境では, なんらかの理由でモバイル機器が密集し, 一部の Cloudlet に大きな負荷がかかるとリソースが枯渇する. そこで, 本研究ではモバイル機器の位置から許される距離にある近傍 Cloudlet のリソースを利用することで, リソースの枯渇を解消する. 提案する手法では, Cloudlet の混雑度を考慮し, 多くのモバイル機器が集まっている場所から割当て処理を行い, モバイル機器と Cloudlet 間の通信遅延が大きくなることを実現している. また, モバイル機器の移動によってアプリケーションの起動が頻発すると Cloudlet に負担がかかるため, 可能な限り同じ Cloudlet を割当ててアプリケーションの起動回数を削減する手法の提案も行う. 評価は, 提案手法と従来手法を用いてシミュレーションによって比較を行う. 評価指標には, 通信遅延を考慮するため最大割当て距離と, アプリケーション起動処理による負荷を考慮するためアプリケーション起動処理発生率を用いる. このシミュレーション評価によって, 混雑度順によって最大割当て距離が削減でき, 移動経路計画を用いることでアプリケーション起動処理発生率を削減できることを確認した.

キーワード 分散処理, クラウドコンピューティング, エッジコンピューティング, モバイル機器, 移動経路計画

Abstract

Resource Allocation Algorithm using The Moving Path Planning in Cloudlet Environment

Kouhei Oosaki

In recent years, it has been proposed to distribute processing with small computers called Cloudlet that are placed on the network. However, Cloudlet is harder to have rich computational resources compared with computers in the data center. Therefore, in an environment where mobile devices frequently move, mobile devices are crowded, some Cloudlets are heavily loaded, and resources may be exhausted.

In this research, we solve the exhaustion of the resource by using neighborhood Cloudlets in the allowable distance from the position of the mobile device. In the proposed method, considering congestion degree of Cloudlet, allocation is done from the place where many mobile devices exist. In addition, we propose a method to allocate the same Cloudlet as much as possible in order to avoid that rebooting of applications frequently occurs and the Cloudlet is over loaded. In the proposed method, we use the Cloudlet at the location planned to be moved using the moving path plan. We evaluate proposed methods by simulation. The evaluation result shows that the proposed method can reduce the maximum allocated distance according to the order of congestion degree, and can reduce rebooting of applications by using moving path planning.

key words Distributed processing system, Cloud computing, Edge computing, Mobile device, Moving path planning

目次

第 1 章	はじめに	1
第 2 章	関連研究	3
2.1	クラウド環境でのリソース割当てアルゴリズム	3
2.2	モバイル機器の移動を考慮したリソース割当てアルゴリズム	4
第 3 章	Cloudlet 環境モデル	5
第 4 章	リソース割当てアルゴリズムの設計方針	10
第 5 章	リソース割当てアルゴリズム	12
5.1	モバイル機器の割当て順序	12
5.1.1	従来手法と問題点	12
5.1.2	提案手法	13
5.2	Cloudlet サーバの選択方法	15
第 6 章	評価	20
6.1	モバイル機器の割当て順序	20
6.1.1	評価方法	20
6.1.2	評価	23
	(1) パターン 1 の結果	23
	(2) パターン 2 の結果	26
	(3) パターン 3 の結果	28
	(4) パターン 4 の結果	30
6.2	Cloudlet サーバの選択方法	31
6.2.1	評価方法	31

目次

6.2.2	評価	32
第 7 章	おわりに	35
7.1	まとめ	35
7.2	今後の課題	35
	謝辞	37
	参考文献	38

目次

3.1	Cloudlet 環境のモデル	5
3.2	Cloudlet の配置	6
3.3	モバイル機器の移動	7
5.1	従来手法の問題点	13
5.2	Cloudlet サーバの配置例	15
5.3	継続割当ての例	17
5.4	移動経路優先による優先順位	19
6.1	入力するモバイル機器の移動経路 (パターン 1)	20
6.2	入力するモバイル機器の移動経路 (パターン 2)	21
6.3	入力するモバイル機器の移動経路 (パターン 3)	22
6.4	入力するモバイル機器の移動経路 (パターン 4)	22
6.5	パターン 1 での最大割当て距離	23
6.6	パターン 1 での平均割当て距離	24
6.7	平均割当て距離が大きくなる例	25
6.8	パターン 2 での最大割当て距離	27
6.9	パターン 2 での平均割当て距離	27
6.10	パターン 3 での最大割当て距離	28
6.11	パターン 3 での平均割当て距離	29
6.12	パターン 4 での最大割当て距離	30
6.13	パターン 4 での平均割当て距離	31
6.14	Cloudlet 選択方法の評価でのアプリケーション起動発生率	32
6.15	Cloudlet 選択方法の評価での最大割当て距離	33

図目次

6.16 Cloudlet 選択方法の評価での平均割当て距離 34

表目次

3.1	移動経路計画の例	8
3.2	Cloudlet 環境モデルで用いる記号の定義	9
5.1	入力されるモバイル機器の例	15
5.2	モバイル機器の割当て順序で定義された記号の定義	16
5.3	Cloudlet サーバの選択方法で定義された記号の定義	19

第 1 章

はじめに

近年，スマートフォンや自動車などのモバイル機器に加え，インターネット接続を行うセンサや家電などの IoT 機器が増加している．これらをクラウド環境で，遠隔操作を行ったり，モバイル機器で行うには負荷が高い処理をサーバで実行することが行われている．このようなサービスでは実時間処理が重要であるが，クラウド環境ではインターネットを介して通信するため，通信遅延が問題となる．そのため，モバイル機器が稼働している地域に，データセンタと比べて小規模な計算資源を持つ Cloudlet を配置し，そこでアプリケーションの処理を行うことによって，ネットワーク遅延やネットワーク負荷を軽減する構成が研究されている [1, 2]．また，Cloudlet と同じように，インターネットとローカルネットワークの境界にエッジサーバを配置し，そこで処理を行うエッジコンピューティングと呼ばれる手法の研究も行われている．

しかし，Cloudlet やエッジサーバは，データセンタと比べて大きなリソースを持たせることが難しい．そのため，モバイル機器が一箇所に集中した場合，その場所で処理を行っている Cloudlet やエッジサーバが持つリソースを超えてしまい，サービスの提供ができない場合がある．本稿では，この問題を改善するため，モバイル機器が利用する Cloudlet やエッジサーバを最寄りのものに限定せず，少し遠くなるが許される遅延時間以内で通信可能なものを利用することを検討する．具体的には，モバイル機器と Cloudlet 間の通信遅延が小さくなる割当て方法を提案する．

本稿の構成を以下に示す．2 章で，自動車などの移動体を対象とした Cloudlet 環境でのモデルの定義とリソース割当てアルゴリズムに関する従来研究を示す．3 章では，リソース割当てアルゴリズムを適用する Cloudlet 環境のモデルを定義する．4 章では，リソース割

当てアルゴリズムの設計方針を述べ，5章で提案するリソース割当てアルゴリズムを示す．
6章でリソース割当てアルゴリズムの性能評価を行うため，シミュレーション実験の方法と
その結果を述べ，7章でまとめと今後の課題について述べる．

第 2 章

関連研究

2.1 クラウド環境でのリソース割当てアルゴリズム

クラウド環境を対象としたリソース割当てに関する研究として, [3, 4, 5, 6] がある.

文献 [3] では, クラウドの物理サーバの想定 CPU 使用率が閾値を超えた場合, 想定 CPU 使用率を閾値以内にするために, 最も少ない仮想マシンの移動方法を求める厳密アルゴリズムの提案を行っている.

文献 [4] では, 仮想マシンの集約効率をベクトルパッキング問題としてモデル化し, 古典的なパッキングアルゴリズムである First-Fit, Best-Fit, First-Fit-Decreasing などのアルゴリズムを改良したヒューリスティックアルゴリズムを提案している.

文献 [6] では, 広域にデータセンタが分散配置されたクラウド環境において, 端末がクラウド上のアプリケーションを実行する上での応答時間を短縮するシステムの提案を行っている. ここでは, アプリケーションの構成要素の配置構成を応答時間を実測しつつ繰り返し配置し直すことで応答時間を短縮している.

文献 [3, 4, 5] は, データセンタ内のクラウド環境を対象に, 物理サーバの CPU 利用率などが閾値を超えないように制御するアルゴリズムである. 一方, 本研究が対象としている環境では, モバイル機器の移動を考慮する必要があることと, モバイル機器を Cloudlet 間の通信遅延を短くすることが目的のため, 直接は適用できない. また, 文献 [6] では通信遅延が考慮されているがモバイル機器の移動が考慮されていない.

2.2 モバイル機器の移動を考慮したリソース割当てアルゴリズム

モバイル機器の移動を考慮したリソース割当て方法として文献 [7, 8] がある。

文献 [7] では、スマートフォンやタブレットなどのモバイル機器におけるリソース面での制約を考慮し、モバイル機器に近いネットワーク上にプロキシサーバを設け、モバイル機器が効率よくクラウドサービスを利用できることで応答時間の短縮やモバイル機器の消費電力の削減を実現するシステムを提案していた。また、はプロキシサーバのキャッシュヒット率を評価していた。文献 [8] の研究では、Cloudlet 環境でのモバイル機器のリソース割当てアルゴリズムを取り扱った研究である。この研究で提案されているモデルでは、モバイル機器の移動に伴って移動先の Cloudlet に対し、AP などの実行環境を移送しなければならないため、移動先の Cloudlet でリソースが不足した場合、実行環境が移送できずサービスの継続ができない場合がある。この問題を改善するため、予め Cloudlet に移送用のリソースを予約することを提案している。

文献 [7] ではプロキシサーバに対して、文献 [8] では Cloudlet に対して、それらが持つリソースを大きく超えるような負荷を考慮したリソース割当てが行われていない。そのため、そのような負荷が発生するとサービスが継続できないという問題が発生する。本稿では、上記問題を解決するため、モバイル機器の移動を考慮した Cloudlet のリソース割当てアルゴリズムを提案する。提案するアルゴリズムは、最も近い Cloudlet だけでなく、通信遅延が許容される範囲内の Cloudlet を割り当てる。その際、モバイル機器と Cloudlet 間の通信遅延が短くなることを考慮した割り当てを行う。

第 3 章

Cloudlet 環境モデル

Cloudlet 環境のモデルを図 3.1 に示す。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、そしてモバイル機器群の三層で構成されていることを前提とする。各 Cloudlet サーバはネットワークを介して相互にデータ通信を行うことができるものとする。以下で図 3.1 のモデルを説明する。

(1) Cloudlet サーバ

各 Cloudlet サーバは図 3.2 のように二次元の格子状に配置された各エリアに 1 つずつ配置されているものとする。

座標 (x, y) における Cloudlet サーバを $c_{x,y}$ で表す。各 Cloudlet サーバには、それぞれ所有リソース量が設定されており、Cloudlet サーバ $c_{x,y}$ の所有リソースが s とする

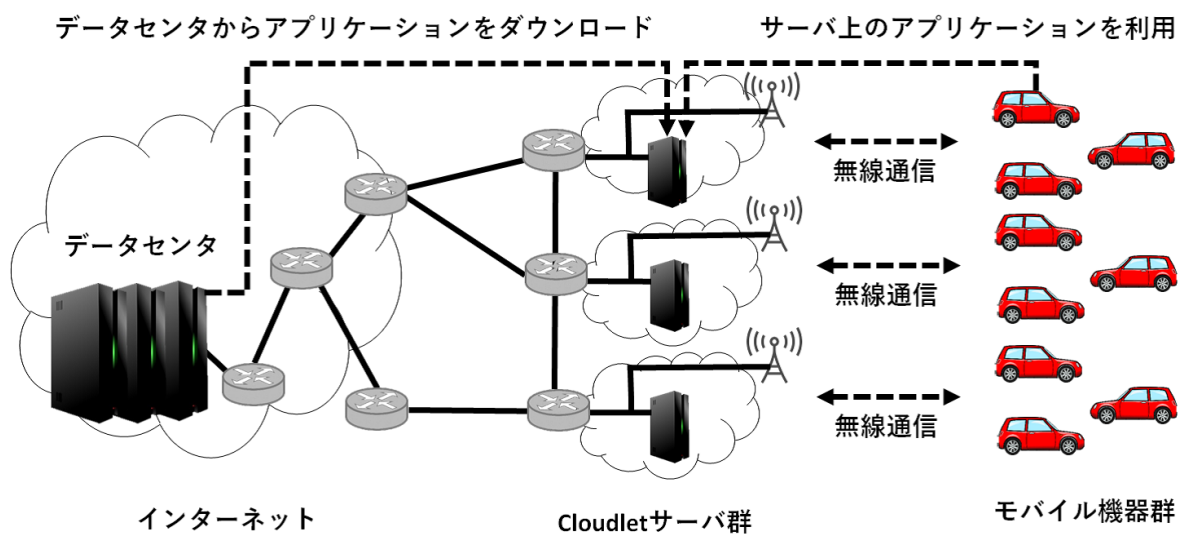


図 3.1 Cloudlet 環境のモデル

Cloudletサーバ

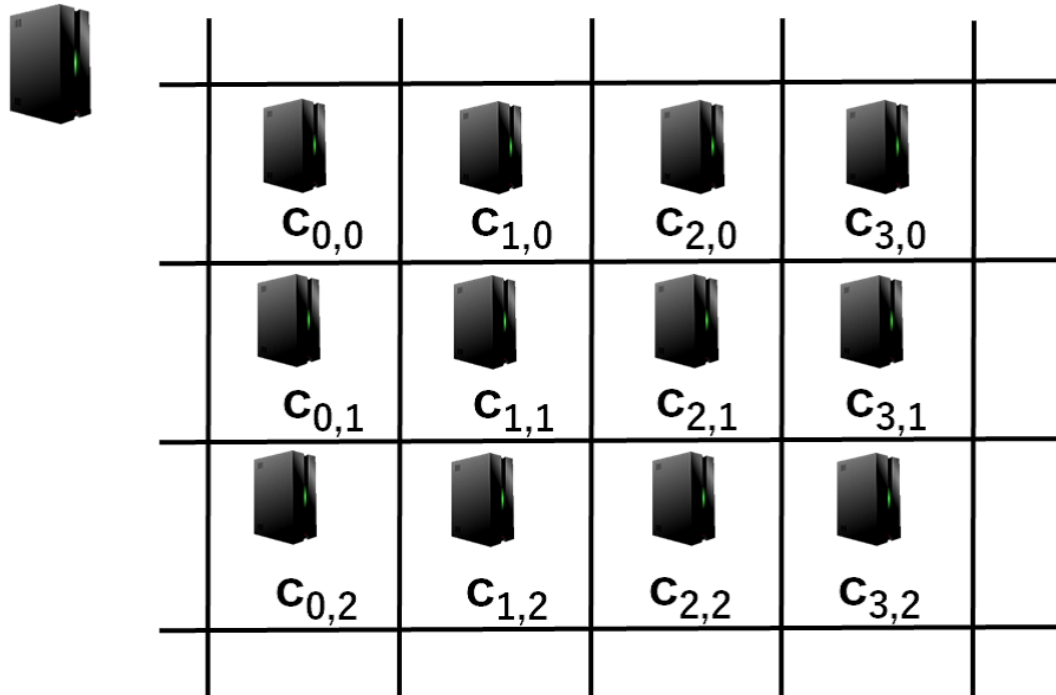


図 3.2 Cloudlet の配置

と, $S(c_{x,y})$ で求められる..

$$S(c_{x,y}) = s$$

実際の環境では, 所有リソースは記憶容量や, CPU コア数など様々な種類のリソースがあるが, 複数種類を考慮すると問題が複雑になるため, ここではリソースを1種類に限定する.

時刻を t で表す. 時刻 t において, 座標 (x,y) の Cloudlet サーバがモバイル機器に対し既に割当てを行った割当て済みリソースを v と表す. v は $V(c_{x,y}, t)$ で求められる.

$$V(c_{x,y}, t) = v$$

(2) モバイル機器

モバイル機器を m_i で表す. モバイル機器にはそれぞれ, アプリケーションの実行に必要な要求リソース量が設定される. あるモバイル機器 m_i の要求リソース量を r と表すと, $R(m_i)$ で求められる.

$$R(m_i) = r$$

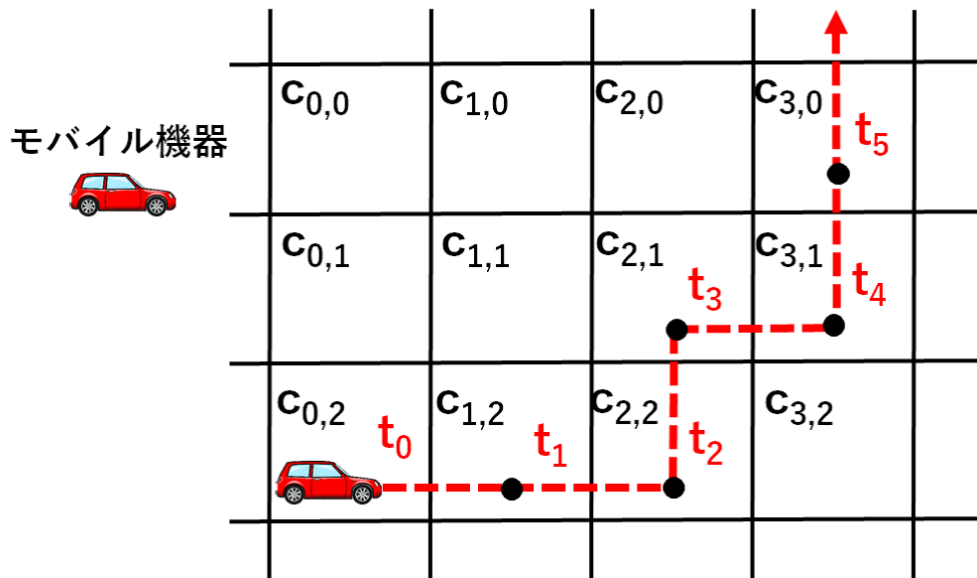


図 3.3 モバイル機器の移動

この値は，時間の経過によって変化することはない，生成された後は常に固定の値を持つ．

(3) モバイル機器の移動モデル

図 3.3 に示すように，モバイル機器は時間の経過によって単位時間あたりに必ず 1 回だけ移動する．モバイル機器は上下左右のみにしか移動することができず，斜めなどの方向には 1 回で移動できない．このため 2 つの座標 (x, y) と (a, b) の距離は $L((x, y), (a, b))$ で求められる．

$$L((x, y), (a, b)) = |x - a| + |y - b|$$

(4) 移動経路計画

モバイル機器は移動経路計画という情報をもっている．移動経路計画は，ある時刻とその時刻にモバイル機器がどのエリアにいるかを示した情報の組からなる．例として図 3.3 に示した移動の移動経路計画を表 3.1 に示す．この情報を用いることで，ある時刻にモバイル機器がどの座標にいるかがわかる．時刻 t におけるモバイル機器 m_i の座標

表 3.1 移動経路計画の例

時刻	モバイル機器の位置
t_0	$c_{0,2}$
t_1	$c_{1,2}$
t_2	$c_{2,2}$
t_3	$c_{2,1}$
t_4	$c_{3,1}$
t_5	$c_{3,0}$

(x, y) は $M(m_i, t)$ で求められる。

$$M(m_i, t) = (x, y)$$

この情報は現実では、カーナビなどから得られる情報をもとに、移動経路計画を得られるものと想定している。

(5) 通信遅延

Cloudlet サーバ間の通信は、モバイル機器の移動と同じように上下左右にしかできないものとする。遅延時間は距離に比例するため、距離を代替指標とする。また、あるモバイル機器と Cloudlet サーバとの距離が 0 の場合、その Cloudlet サーバを最寄 Cloudlet サーバと呼ぶ。

Cloudlet 環境モデルで用いた記号の定義を表 3.2 にまとめる。

表 3.2 Cloudlet 環境モデルで用いる記号の定義

記号	意味
$c_{x,y}$	座標 (x, y) に位置する Cloudlet サーバ
t	時刻
m_i	モバイル機器
$R(m_i)$	モバイル機器 m_i の要求リソース量
$S(c_{x,y})$	座標 (x, y) における Cloudlet サーバの所有リソース量
$V(c_{x,y}, t)$	時刻 t , 座標 (x, y) における Cloudlet サーバの割当て済みリソース量
$L((x, y), (a, b))$	座標 (x, y) と座標 (a, b) の距離
$M(m_i, t)$	時刻 t におけるモバイル機器 m_i の座標

第 4 章

リソース割当てアルゴリズムの設計方針

以下に、リソース割当てアルゴリズムの設計方針を示す。

(1) 近傍 Cloudlet サーバの利用

本稿で検討するリソース割当てアルゴリズムでは、モバイル機器が利用できる Cloudlet サーバは、最寄 Cloudlet サーバに限らず、通信遅延が許容する範囲内の Cloudlet サーバならば利用できる。モバイル機器の最寄 Cloudlet サーバと利用する Cloudlet サーバの距離を割当て距離と呼ぶ。

(2) 最大遅延時間の最小化

上下左右に隣接したエリアの Cloudlet サーバを利用する場合、割当て距離は 1 となり、また、斜めに隣接したエリアの Cloudlet サーバを利用する場合は割当て距離は 2 となる。現実では、この割当て距離は、ルータのホップ数などに相当し、これが大きくなると通信遅延が大きくなったりネットワークの負荷を増加させることに繋がる。実時間処理では、最大遅延時間が大きい場合、処理が破綻することがある。そのため、検討するリソース割当てアルゴリズムでは、全てのモバイル端末の割当て距離の中で最大のもの(以降、最大割当て距離と呼ぶ)を可能な限り小さくする。

(3) Cloudlet 切替の減少

モバイル機器が利用するアプリケーションは、Cloudlet サーバ上で実行される。そのため、モバイル機器が利用するアプリケーションが割当て Cloudlet サーバ上に起動さ

れていない場合、Cloudlet サーバはアプリケーションを起動する必要がある。アプリケーションは1度起動するとその後も動作し続けるわけではなく、利用するモバイル機器が別のエリアに移動するとアプリケーションを終了させる。アプリケーションの起動処理は、Cloudlet サーバに負荷がかかる。これが頻発すると大きな負荷がかかると予想できるため、検討するリソース割当てアルゴリズムでは、このアプリケーション起動発生数を少なく抑える。

(4) アルゴリズムの計算量

移動体が多い環境では、頻繁に割当て計画を修正する必要がある。そのため単位時間ごとにリソース割当て計画を算出する。計算時間の観点から、厳密に最も効率のよいリソース割当てを行うことは難しいため、短時間で効率のよい割当て計画を出力するヒューリスティックアルゴリズムを検討する。

第 5 章

リソース割当てアルゴリズム

リソース割当てアルゴリズムは，各単位時間に実行され，各モバイル機器に対して Cloudlet サーバを割り当てる．その際，どのようなアルゴリズムでモバイル機器のリソース割当て順序を決めるかという点と，近傍 Cloudlet サーバのうち，どの Cloudlet サーバを選択するかを決めるアルゴリズムが重要になる．

5.1 モバイル機器の割当て順序

5.1.1 従来手法と問題点

モバイル機器の割当て順序として，従来手法を単純に適用する 2 つの方法がある．

(1) 到着順

到着順は，単純に入力されたモバイル機器の順番に割当て処理を行う．

(2) 要求リソース順

要求リソース順は，入力されたモバイル機器が要求するリソースで，降順にソートした順番に割当て処理を行う．

要求リソース順は，参考文献で紹介した First-Fit-Decreasing を単純に適用したものである．この手法の良いところは，最初に小さな要求リソース量のモバイル機器を割り当てて，遅いタイミングで大きな要求リソース量のモバイル機器を割り当てようとする時，モバイル機器の近傍には，小さなリソースが断片的に残されているだけの状態になっている可能性がある．それを回避するために，最初のほうに大きなリソースを要求するモバイル機器の割り

5.1 モバイル機器の割当て順序

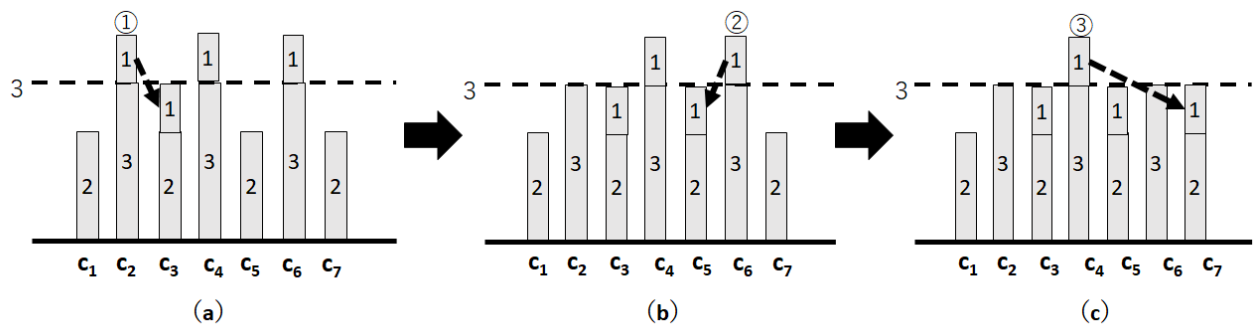


図 5.1 従来手法の問題点

当てを行う。

到着順と要求リソース順の問題点を図 5.1 を用いて説明する。ここでは、到着順を例にする。実際に割当てアルゴリズムを考えるモデルでは、Cloudlet サーバは 2 次元に配置されているが、ここでは説明のため 1 次元に配置されているものとする。各 Cloudlet サーバの割当て可能リソース量は 3 であり点線で示している。図中に各モバイル機器からの Cloudlet サーバに対する要求リソース量を示している。図 5.1(a) では、モバイル機器が c_2 へ、要求リソース量 1 を要求したが、 c_2 には、割当てられるリソースが残っていないため、 c_3 を割当てている。次に到着したモバイル機器は、最寄 Cloudlet サーバである c_4 に割り当てられず、また、隣の c_3 と c_5 も割り当てることができないため、距離的に離れた c_7 へ割り当てることになる。つまり、モバイル機器の通信遅延が大きくなる。

5.1.2 提案手法

この問題を回避するため、先にリソース要求の多いエリアとその周辺にあるモバイル機器から Cloudlet を割り当てるという考えで割当て順序を決める混雑度順を提案する。リソース要求の多いエリアだけでなく、その周辺にあるモバイル機器も先に割当てを行う理由は、周辺に混雑エリアが複数あると、そのエリアのモバイル機器によって周辺の Cloudlet サーバのリソースが取られるからである。

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、最寄 Cloudlet サー

5.1 モバイル機器の割当て順序

バの混雑度が大きい順にモバイル機器をソートして割り当てる手法である。混雑度は、事前に指定した加算距離と呼ぶ距離以内にある全てのモバイル機器の要求リソース量の総和である。そして、この混雑度が高い Cloudlet サーバを最寄 Cloudlet サーバとするモバイル機器から割当てを行う。時刻 t 、座標 (x, y) における全モバイル機器の要求リソース量の総和を $R(x, y, t)$ で表す。

$$R(x, y, t) = \sum_{M(m_i, t)=(x, y)} R(m_i)$$

加算距離を ℓ_a と表す。また、時刻 t における座標 (x, y) の Cloudlet サーバの混雑度 $G(x, y, t)$ で表す。

$$G(x, y, t) = \sum_{i=-\ell_a}^{\ell_a} \sum_{j=-(\ell_a-|i|)}^{\ell_a-|i|} R(x+i, y+j, t)$$

入力するモバイル機器を表 5.1、Cloudlet サーバの配置を図 5.2 とし、加算距離を 1 とした場合の混雑度順の例を示す。まず m_1 と m_2 の最寄 Cloudlet サーバである $c_{1,0}$ の混雑度を求める。 $c_{1,0}$ から距離 1 以内にある Cloudlet サーバ $c_{0,0}$, $c_{1,0}$, $c_{2,0}$, $c_{1,1}$ のうち、モバイル機器に最寄 Cloudlet サーバとされている Cloudlet サーバは $c_{1,0}$ と $c_{1,1}$ である。また、Cloudlet サーバ $c_{1,0}$ を最寄 Cloudlet サーバとするモバイル機器は、 m_1 と m_2 であり、その要求リソースの和は 3 である。Cloudlet サーバ $c_{1,1}$ を最寄 Cloudlet とするモバイル機器は m_3 のみであり、その要求リソースは 3 である。これらの総和である 6 が c_2 の混雑度となる。次に m_3 の最寄 Cloudlet サーバである $c_{1,1}$ の混雑度を求める。 $c_{1,1}$ から距離 1 以内にある Cloudlet サーバは $c_{1,0}$, $c_{0,1}$, $c_{1,1}$, $c_{2,1}$, $c_{1,2}$ である。これらのうち、モバイル機器に最寄 Cloudlet サーバとされている Cloudlet サーバは、 $c_{1,0}$, $c_{1,1}$, $c_{2,1}$ のみとなる。Cloudlet サーバ $c_{1,0}$ と $c_{1,1}$ を最寄 Cloudlet サーバとするモバイル機器の総要求リソースはそれぞれ 3 である。 $c_{2,1}$ を最寄 Cloudlet サーバとするモバイル機器は m_4 であり、その要求リソースは 5 である。これらの総和は 11 であり、これが $c_{1,1}$ における混雑度となる。最後に d_3 の最寄 Cloudlet である $c_{2,1}$ の混雑度を求める。Cloudlet サーバ $c_{2,1}$ から距離 1 以内にある Cloudlet サーバは $c_{2,0}$, $c_{1,1}$, $c_{2,1}$, $c_{2,2}$ である。これらのうち、モバイル機器に最寄 Cloudlet サーバとされている Cloudlet サーバは、 $c_{1,1}$ と $c_{2,1}$ のみである。 $c_{1,1}$

5.2 Cloudlet サーバの選択方法

表 5.1 入力されるモバイル機器の例

モバイル機器名	最寄 Cloudlet	要求リソース量
m_1	$c_{1,0}$	2
m_2	$c_{1,0}$	1
m_3	$c_{1,1}$	3
m_4	$c_{2,1}$	5

C_{0,0}	C_{1,0}	C_{2,0}
C_{0,1}	C_{1,1}	C_{2,1}
C_{0,2}	C_{1,2}	C_{2,2}

図 5.2 Cloudlet サーバの配置例

を最寄 Cloudlet サーバとするモバイル機器の要求リソースの総和は 3 であり, $c_{2,1}$ を最寄 Cloudlet サーバとするモバイル機器の要求リソースの総和は 5 である. これらの総和は 8 であり, これが $c_{2,1}$ の混雑度となる. 各モバイル機器の最寄 Cloudlet サーバの混雑度は, $c_{1,1} > c_{2,1} > c_{1,0}$ となり, 大きい順にモバイル機器をソートすると, 割当て順序は, m_3, m_4, m_1, m_2 となる.

モバイル機器の割当て順序で定義された記号の定義を表 5.2 にまとめる.

5.2 Cloudlet サーバの選択方法

各モバイル機器に対し, 割当てる Cloudlet サーバの選択は以下の 3 種類の方法を検討した.

5.2 Cloudlet サーバの選択方法

表 5.2 モバイル機器の割当て順序で定義された記号の定義

記号	説明
ℓ_a	加算距離
$R(x, y, t)$	時刻 t , 座標 (x, y) に位置する全モバイル機器の要求リソースの総和
$G(x, y, t)$	時刻 t , 座標 (x, y) における Cloudlet サーバの混雑度

(1) 最近傍選択

最近傍選択は、単純に最も近い Cloudlet から順番に割当て可能かを判定し、割当て可能なものを選択する方法である。時刻 t において、モバイル機器 m_i に対し、割当て可能な Cloudlet サーバは次式を満たす場合である。

$$S(c_{x,y}) - V(c_{x,y}, t) \geq R(m_i)$$

(2) 継続割当て・最近傍選択

継続割当てとは、前の時刻に割当てられた Cloudlet サーバが一定距離以内であれば、その Cloudlet サーバを継続して割当てるという手法である。継続して割当てることができる距離を継続割当て距離と呼ぶ。時刻 t において、モバイル機器 m_i に割当てられた Cloudlet サーバの座標 (x, y) は $A(m_i, t)$ で得られるものとする。

$$A(m_i, t) = (x, y)$$

これを利用すると、モバイル機器 m_i に前の時刻で割当てられた Cloudlet サーバの座標 (a, b) は次式で得られる。

$$(a, b) = A(m_i, t - 1)$$

このとき、継続割当て距離を ℓ_c と表すと、継続割当ての対象となる Cloudlet サーバは次式を満たすものである。

$$\ell_c \geq L(A(m_i, t - 1), M_i(t))$$

また、継続割当て距離内であっても、Cloudlet サーバが割当てに必要なリソースを持っていない場合は継続割当ては行えない。この継続割当てができなければ、最近傍選

5.2 Cloudlet サーバの選択方法

継続割当て距離=1

t_0 : 距離1の $c_{1,0}$ へ割り当て t_2 : 距離1の $c_{1,0}$ へ割り当て

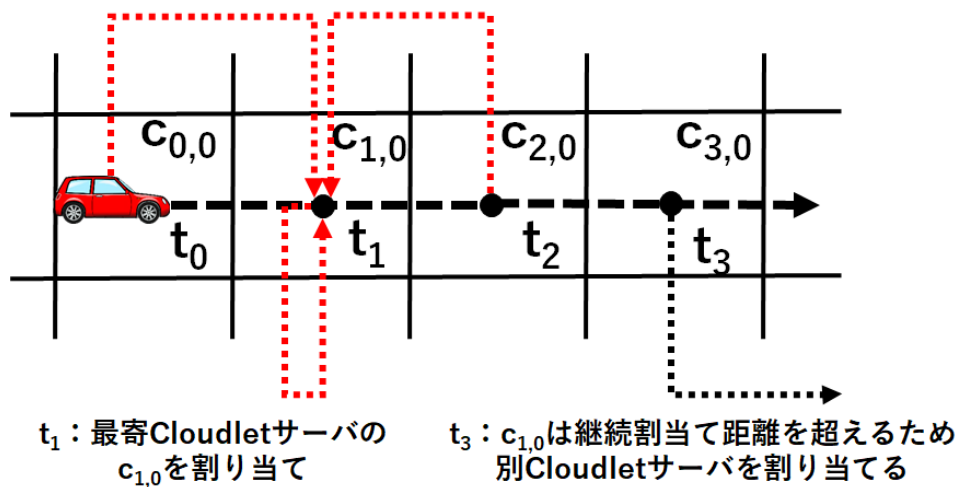


図 5.3 継続割当ての例

択を行う。

継続割当ての例を図 5.3 に示す。この例では継続割当て距離を 1 としている。時刻 t_0 で Cloudlet サーバ $c_{0,0}$ のエリアにあるモバイル機器に対し、 $c_{1,0}$ が割り当てられたとする。このあとモバイル機器は、時刻 t_1 で $c_{1,0}$ に進む。この時、一つ前の時刻 t_0 で割り当てられた Cloudlet サーバ $c_{1,0}$ が距離 1 以内にあるため、このモバイル機器に再び Cloudlet サーバ $c_{1,0}$ を割り当てる。時刻 t_2 では、モバイル機器は Cloudlet サーバ $c_{2,0}$ の場所に進む。このとき、Cloudlet サーバ $c_{1,0}$ は距離 1 以内にあるため、再び Cloudlet サーバ $c_{1,0}$ を割り当てる。時刻 t_3 では、モバイル機器は Cloudlet サーバ $c_{3,0}$ に移動し、Cloudlet サーバ $c_{1,0}$ との距離は 2 となる。この時、継続割当て距離を超えるため、継続割当ては行わず、最近傍選択によって割り当てる Cloudlet サーバを新たに選択する。

(3) 継続割当て・移動経路優先選択

この手法は、継続割当てに加えて、移動経路計画を用いて、未来に通過するエリアの Cloudlet サーバとその周辺 Cloudlet サーバを優先して割り当てる手法である。まず

5.2 Cloudlet サーバの選択方法

一定時間後のモバイル機器の最寄 Cloudlet サーバを移動経路計画から取得する。この Cloudlet サーバを移動予定 Cloudlet サーバと呼ぶ。また、どれくらい先の時刻をみるかを先読み時間と呼び、 t_f と表す。これより、時刻 t において、 t_f 時間後にモバイル機器 m_i が通過するエリアの座標 (a, b) は、次式で求められる。

$$(a, b) = M(m_i, t + t_f)$$

次に、モバイル機器の位置から継続割当て距離以内にある Cloudlet サーバをに対し優先度付けを行う。時刻 t におけるモバイル機器 m_i の場所が座標 (x, y) のとき、Cloudlet サーバの優先度を $P(x, y, t)$ で表す。

$$P(x, y, t) = L(M(m_i, t + t_f), (x, y))$$

継続割当て距離以内に割当て可能な Cloudlet サーバがあれば、もっとも優先度の高いものを割り当てる。割当て可能な Cloudlet サーバなければ、調べる距離を 1 ずつ増やして、割当て可能なものが見つかるまで繰り返す。

継続割当て距離を 2、先読み時間を 2 とした場合の優先順位の例を図 5.4 に示す。図では Cloudlet サーバは省略しているが、各セルに Cloudlet サーバがあるものとする。図中の矢印線が、モバイル機器の移動経路を示している。先読み時間は 2 であるため、時刻 t_0 から 2 進んだ時刻である時刻 t_2 のときの最寄 Cloudlet サーバの場所を移動予定位置とする。移動予定 Cloudlet サーバの位置を図中の * で示している。継続割当て距離以内エリアは色付きで示されている。エリアに示されている数字は優先度を示している。まず、* で示された移動予定 Cloudlet サーバへの割り当てを試みる。これができなければ、次に優先度 1 の Cloudlet サーバへ割り当てを試みる。これも割り当てができなければ、優先度 2 の Cloudlet サーバに試みる。このようにして、優先度 4 のエリアまで割り当てできなければ、調べる距離を 1 だけ増やし、もう一度優先度付けを行う。これを割り当て可能な Cloudlet サーバが見つかるまで繰り返す。

(2), (3) の手法は、継続割当てを行うことによって、可能な限り同じ Cloudlet サーバを利用し、モバイル機器が利用する Cloudlet サーバの変更回数が少なくする。これによって

5.2 Cloudlet サーバの選択方法

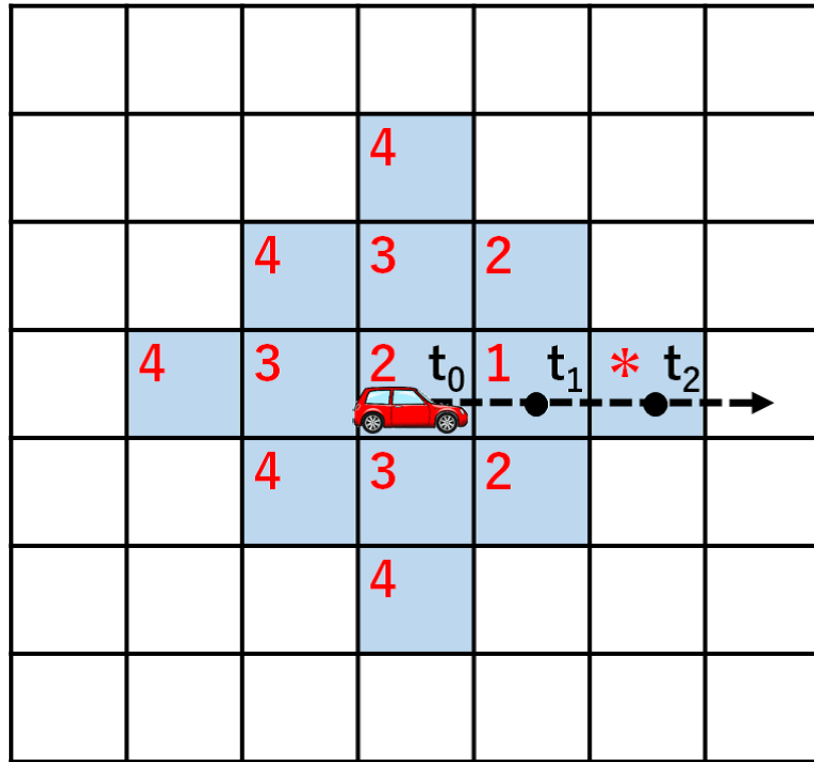


図 5.4 移動経路優先による優先順位

アプリケーションの起動処理発生回数を低減させることを狙う。また、(3) の手法により、モバイル機器に割当てられる Cloudlet サーバが、モバイル機器がこれから先の時刻に通過するエリアに近くなり、継続割当てが成立しやすくなる。そのため、(2) の手法よりも、モバイル機器が利用する Cloudlet サーバの変更が発生しにくくなる。

Cloudlet サーバの選択方法で定義された記号の定義を表 5.3 にまとめる。

表 5.3 Cloudlet サーバの選択方法で定義された記号の定義

記号	説明
l_c	継続割当て距離
t_f	先読み時間
$A(m_i, t)$	時刻 t にモバイル機器 m_i へ割当てられた Cloudlet サーバの座標
$P(x, y, t)$	時刻 t における座標 (x, y) にある Cloudlet サーバの割当て優先度

第 6 章

評価

ここまで説明したアルゴリズムの性能を調べるため、シミュレーションにより評価を行った。評価は、モバイル機器の割当て順序と Cloudlet サーバの選択手法の 2 つを評価した。

6.1 モバイル機器の割当て順序

6.1.1 評価方法

まず、混雑度順による割当て順序が単純な到着順と要求リソース順よりも結果が良くなることを確認する。そのため、図 6.1 に示すパターン 1 の移動経路を持つモバイル機器を入力データとしてシミュレーション評価を行う。*で示されているエリアは、モバイル機器が移

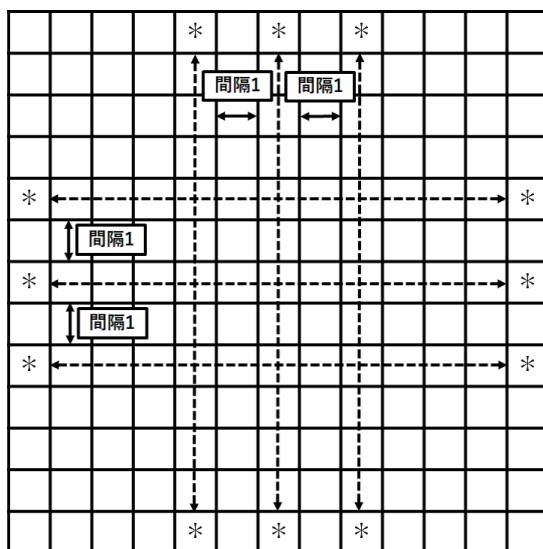


図 6.1 入力するモバイル機器の移動経路 (パターン 1)

6.1 モバイル機器の割当て順序

動を開始するエリアである。また、モバイル機器は、矢印線に従って移動し、*で示されているエリアに到達すると移動を終了しモバイル機器をシャットダウンする。そのため、矢印線の通っているエリアは多くのモバイル機器が通過する場所となる。中心部は多くのモバイル機器が交差するように移動するため、モバイル機器が通過するエリアでなくても、周辺へのリソース割当てにより負荷が発生する。そのため、モバイル機器が通過しないエリアであっても、リソース不足が発生する可能性がある。

また、図 6.1 よりもやや離れた移動経路を持つ、図 6.2 のパターン 2 と図 6.3 のパターン 3 の移動経路でもシミュレーション評価を行う。混雑度順は、加算距離というパラメータによって、近くにあるモバイル機器の要求リソース量も考慮しているが、この加算距離に入らなかったモバイル機器が、割当てに与える影響をこの異なるパターンを使って調べる。

また、やや離れた移動経路だけではなく、移動経路同士が密着した図 6.4 のパターン 4 を用いたシミュレーション評価も行う。

この評価では、モバイル機器の割当て順序による割当て距離の削減効果を見るため、割当て Cloudlet サーバの選択方法は、すべて最近傍順で行う。評価方法は、割当て処理を行って得た割当て計画から、各手法における最大割当て距離と平均割当て距離を比較する。

入力する Cloudlet サーバのデータは、 30×30 の Cloudlet サーバ集合とする。また、そ

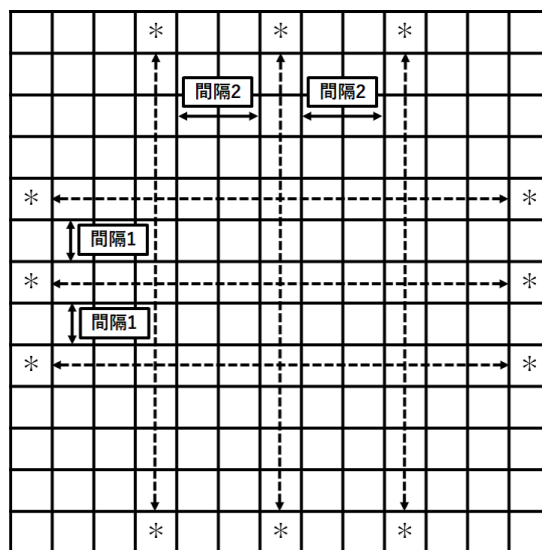


図 6.2 入力するモバイル機器の移動経路 (パターン 2)

6.1 モバイル機器の割当て順序

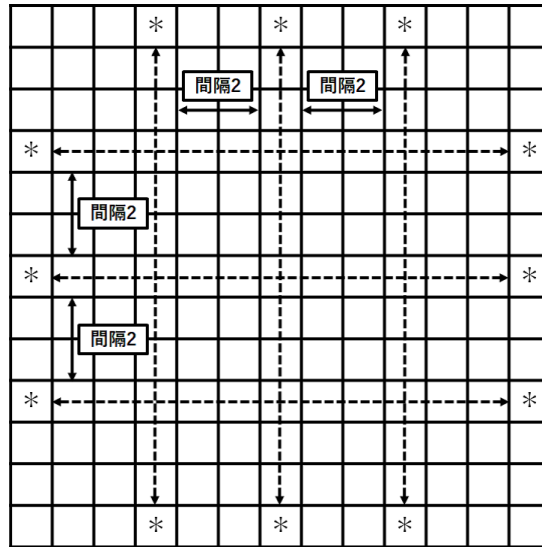


図 6.3 入力するモバイル機器の移動経路 (パターン 3)

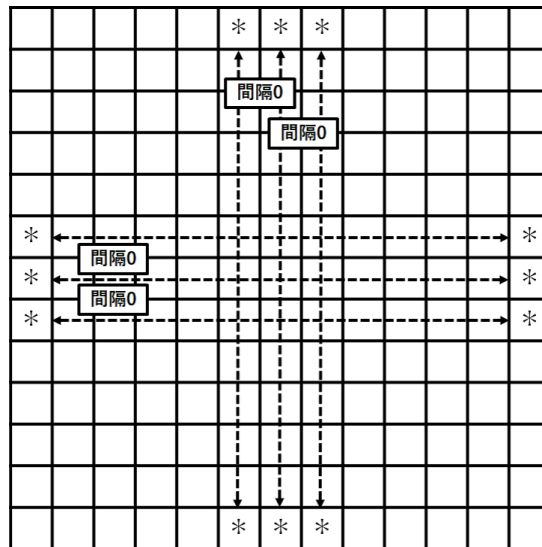


図 6.4 入力するモバイル機器の移動経路 (パターン 4)

れぞれが持つリソースは 200 と設定している。モバイル機器は要求リソース量は 1 から 5 のランダムで設定された値を持ち、各移動開始エリアで起動するモバイル機器数を変えた 6 つのデータを用意した。実際の環境で想定する 1 エリアの大きさは、携帯電話などの基地局のようなものを考え、数百 m から数 km の範囲を想定した。そのため、モバイル機器が多く通過するエリアでは、数十から数百台のモバイル機器が通過するように、モバイル機器数を設定した。これらのデータに対し、30 単位時間でのシミュレーションを実行する。

6.1 モバイル機器の割当て順序

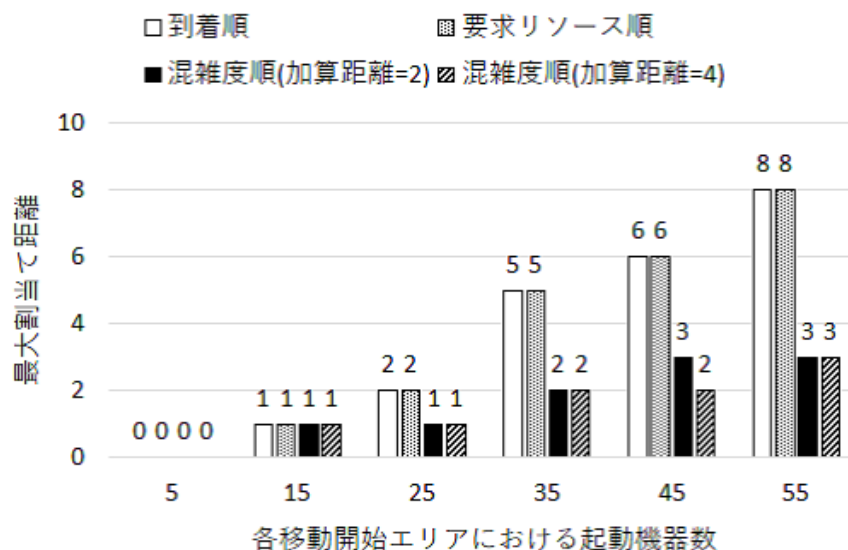


図 6.5 パターン 1 での最大割当て距離

評価指標には、全ての時刻での全てのモバイル機器での割当て距離のうち、最も大きなものを示す最大割当て距離と、全ての時刻での全てのモバイル機器の割当て距離の総和を、割当て成立回数で割った平均割当て距離を用いる。

6.1.2 評価

(1) パターン 1 の結果

移動経路パターン 1 を用いたシミュレーションによって得られた最大割当て距離を図 6.5 に示す。このグラフの縦軸は、各手法でのすべての割当て結果における最も大きかった割当て距離を示している。横軸は、各移動開始エリアでモバイル機器が起動する数である。各移動開始エリアで起動するモバイル機器数が 25 まででは大きな差は出ないが、35 以上になると、混雑度順による最大割当て距離が、到着順や要求リソース順と比べ、3 分の 1 近く小さい値となっている。これは、移動を開始するモバイル機器数が 25 を超えると、到着順や要求リソース順では移動経路が交差する中心部付近で、Cloudlet サーバのリソースが不足し、交差部分の外側にある Cloudlet サーバに対してモバイル機器を割り当てようとするため

6.1 モバイル機器の割当て順序

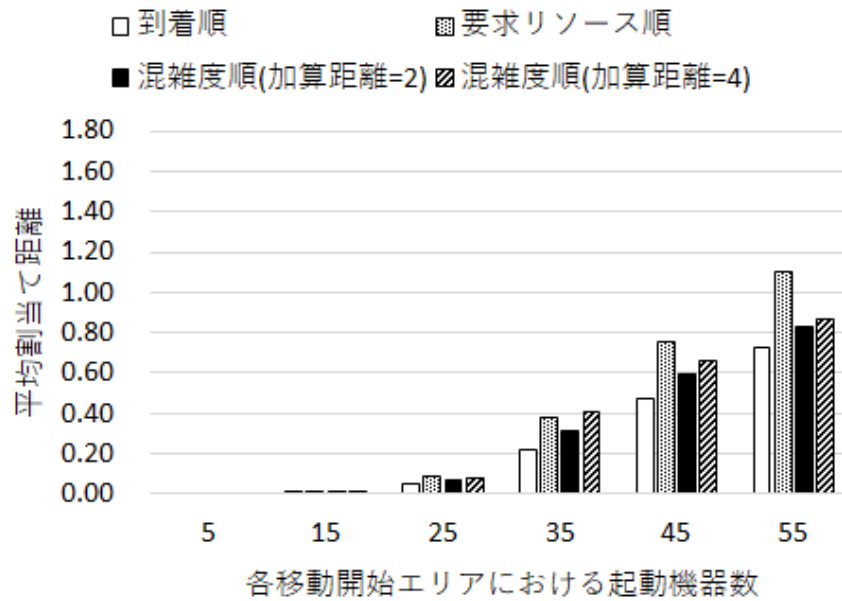


図 6.6 パターン 1 での平均割当て距離

あると考えられる。混雑度順では中心部から割り当てを行うことで、最大割当て距離を大きく抑制できていると考えられる。

混雑度順の中でも、加算距離が 2 の場合と 4 の場合を比較すると起動機器数が 45 の時に 2 の場合の方が、最大割当て距離が大きくなっていた。最大割当て距離が大きくなったモバイル機器を調べると、中心周辺のエリアにあるモバイル機器に対して割り当てを行う際に最大割当て距離が大きくなっていた。これは、加算距離が適切に設定されていない場合、近傍 Cloudlet サーバのリソースが加算距離外のモバイル機器に割当てられてしまい、遠くの Cloudlet サーバに割り当てる必要が生じたと考えられる。

また、モバイル機器の要求リソース順に割り当て処理を行った場合は、到着順と同じ結果になった。これについては、実験で使用したデータでは、モバイル機器の要求リソースが、Cloudlet サーバが持つリソースに対して非常に小さく、また、各モバイル機器の要求リソースに大きな差がなかったため、要求リソース順に並び替えることによる効果が出なかったと考えられる。

次に平均割当て距離の評価について述べる。平均割当て距離を図 6.6 に示す。このグラフ

6.1 モバイル機器の割当て順序

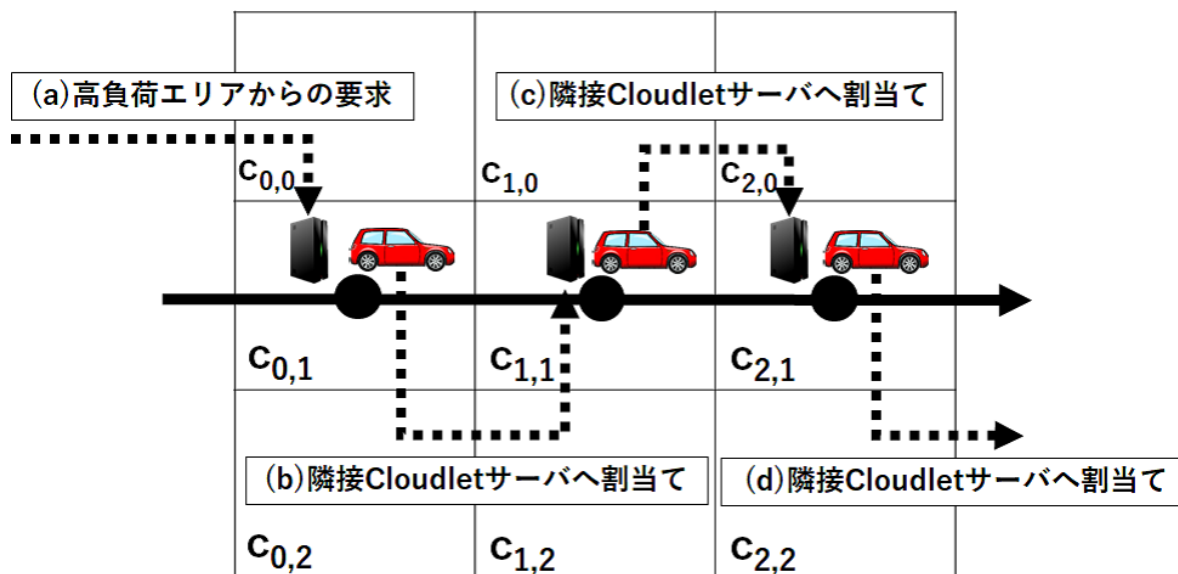


図 6.7 平均割当て距離が大きくなる例

の縦軸は、各手法での割当て距離の平均値を示している。低負荷状態では、3つの手法での大きな差は見られないが、負荷が増加するにつれ、要求リソース順と混雑度順では到着順と比較してやや大きな平均割当て距離となった。しかしながら、混雑度順では、起動機器数が35の場合、要求リソース順よりも大きくなっているものの、多くの場合では要求リソース順よりも良い結果を得られている。

混雑度順での平均割当て距離が到着順より大きい理由を考察する。混雑度順では、多くのモバイル機器が集まっているエリアから割当てが行われる。そのエリアにある Cloudlet サーバの所有リソース量を超過するリソース要求があった場合、近傍 Cloudlet サーバを利用することになる。その際、その近傍 Cloudlet サーバにリソース要求があるかどうかは考慮しないため、負荷が集中するエリアの近傍にあるモバイル機器は最寄 Cloudlet サーバに割当てられる可能性が低くなる。そのような割当ての例を図 6.7 に示す。図中の実線の矢印は、モバイル機器の移動経路を示し、モバイル機器が通過するエリアである。点線の矢印はそのエリアにあるモバイル機器に割当てられる Cloudlet サーバの場所を示している。まず (a) に示すように、Cloudlet サーバ $c_{1,0}$ のエリアに対し高負荷エリアから割当て要求を受けてリソースがすべて予約済みになったとする。そのため、Cloudlet サーバ $c_{0,1}$ を最寄

6.1 モバイル機器の割当て順序

Cloudlet サーバとするモバイル機器は (b) に示すように、隣接している Cloudlet サーバ $c_{1,1}$ に対し割当てが行われたとする。この時、Cloudlet サーバ $c_{1,1}$ を最寄 Cloudlet サーバとするモバイル機器は、Cloudlet サーバ $c_{1,1}$ に割り当てることができず、(c) に示すようにさらに隣の Cloudlet サーバである Cloudlet サーバ $c_{2,1}$ に対し割当てを行う。これによって、Cloudlet サーバ $c_{2,1}$ を最寄 Cloudlet サーバとするモバイル機器もまた最寄 Cloudlet サーバに割り当てることができないため、(d) に示すように他の Cloudlet サーバに対してリソース割当てを求める。このように、近傍 Cloudlet サーバを割り当てる際、他のモバイル機器がその Cloudlet サーバに対してリソース要求を行うことを考慮しないため、連鎖的に隣の Cloudlet サーバのリソースを割り当てることになる場合がある。これによって混雑度順では、平均割当て距離が大きくなると考えられる。また、高負荷エリアとその加算距離以内の低負荷エリアでは、高負荷エリアの影響を受け、そこから近い順番に割当てられる可能性があるため、このような現象が起きやすいと考えられる。加算距離が大きい場合、このような現象が広い範囲で起こるため、平均割当て距離が増加したと考えられる。加算距離が小さい場合は、狭い範囲でしか影響を与えないため、図 6.7 に示した割当て例になりにくいと考えられる。

到着順は、偏りの少ないランダムな到着順序であれば、広い範囲で到着順によって均等に割当てが行われる。そのため、各 Cloudlet サーバの所有リソースが超過する段階では、周囲のリソース要求のある Cloudlet サーバも同様にリソースがすでに割当てられている状態になっている可能性が高い。そのため、Cloudlet サーバの所有リソース超過後は、リソース要求の少ない Cloudlet サーバに割当てられることになると考えられる。そのため平均割当て距離が小さくなっていると考えられる。

(2) パターン 2 の結果

移動経路パターン 2 を用いたシミュレーションによって得られた最大割当て距離を図 6.8 に示す。

6.1 モバイル機器の割当て順序

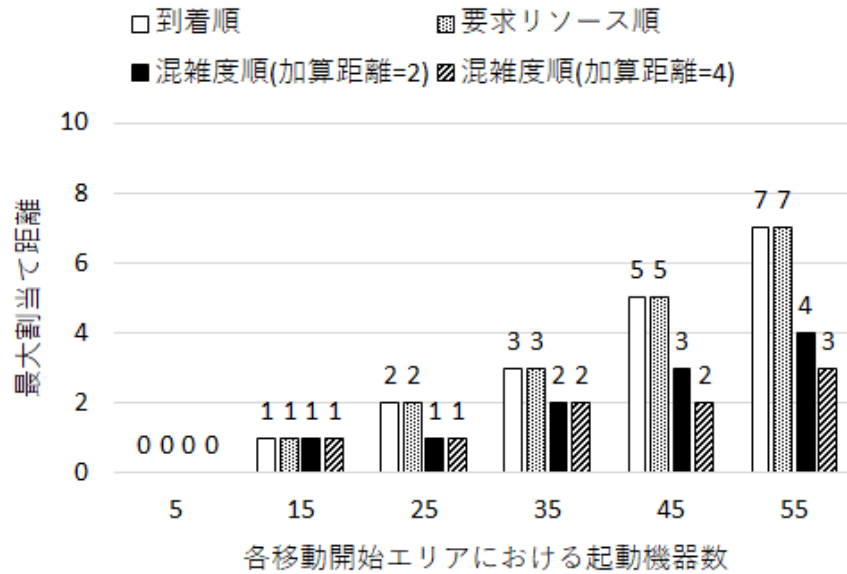


図 6.8 パターン 2 での最大割当て距離

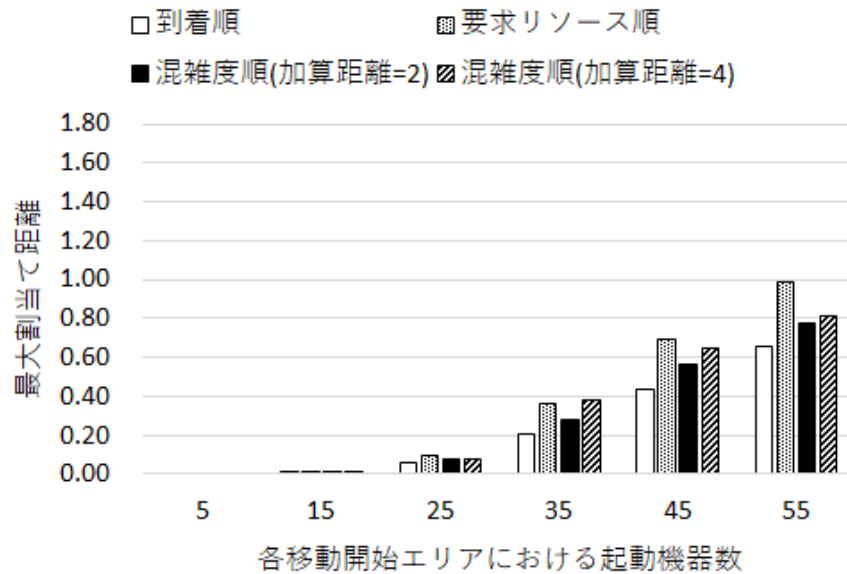


図 6.9 パターン 2 での平均割当て距離

このパターンでは、縦方向の移動経路間の間隔がパターン 1 より大きいため、負荷の集中がやや和らいでいる。そのため、到着順と要求リソース順では、パターン 1 では 8 まで増加していた最大割当て距離がパターン 2 では 7 に留まっている。しかし、加算距離が 2 の混雑

6.1 モバイル機器の割当て順序

度順では、起動機器数が55の場合、最大割当て距離が4となっており、パターン1よりも増加している。このように、混雑度順は適切な加算距離をとることができなかった場合、最大割当て距離がやや増加する。

次に平均割当て距離の評価について述べる。平均割当て距離を図6.9に示す。平均割当て距離については、パターン1とくらべて全体的に下がった以外には、とくに変化は見られなかった。そのため、移動経路間の距離を多少程度変えたとしても、性能あまり影響をあたえることはないと考えられる。

(3) パターン3の結果

パターン3を用いた場合の最大割当て距離を図6.10に示す。このパターンでは、縦と横の移動経路がそれぞれ2ずつ離れており、パターン1やパターン2よりもモバイル機器のさらに密度が低い。そのため、到着順と要求リソース順では各移動開始エリアにおける起動機器数が35のときの最大割当て距離がパターン2よりもさらに1下がっている。また各移動開始エリアにおける起動機器数が45の時では、混雑度順(加算距離=2)の場合、パターン2

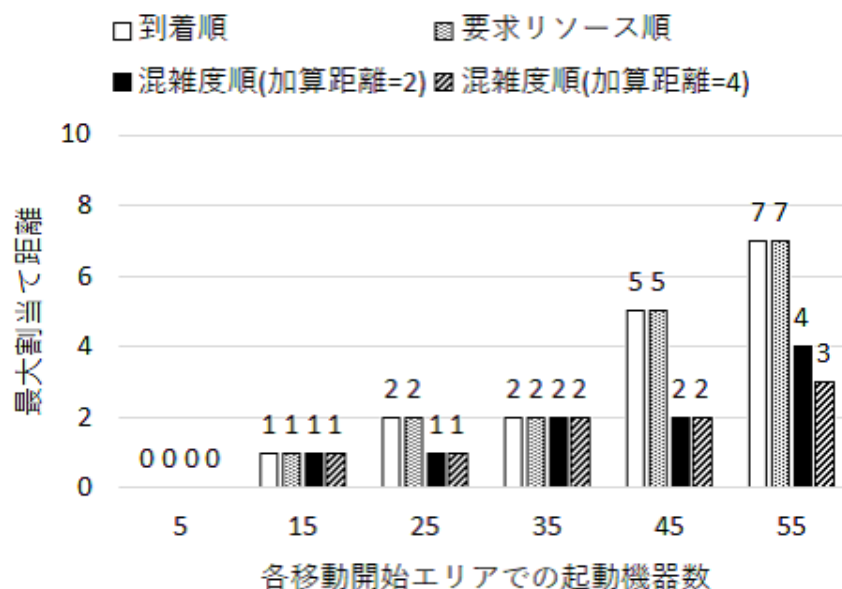


図 6.10 パターン3での最大割当て距離

6.1 モバイル機器の割当て順序

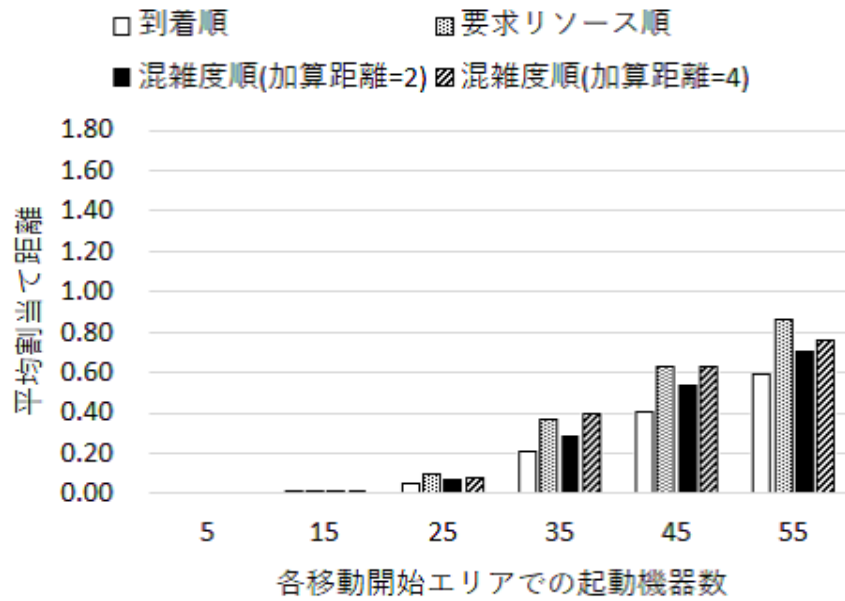


図 6.11 パターン 3 での平均割当て距離

では最大割当て距離が 3 であったのに対し、パターン 3 では 2 に下がっている。パターン 2 とパターン 3 のようにモバイル機器の密度を下げても、混雑度順 (加算距離=4) では割当て距離に変化は見られなかった。しかし一方で混雑度順 (加算距離=2) では密度を下げることによって結果が悪化した場合があった。加算距離よりも少しだけ大きな距離に負荷が非常に大きなエリアがある場合、それによって周囲の Cloudlet のリソースが不足し、加算距離以上に離れたエリアにも影響を与える可能性がある。そのような少し遠くの距離にあるモバイル機器の要求リソース量を考慮できないため、最大割当て距離がやや伸びるケースがあったと考えられる。

また、平均割当て距離を図 6.11 に示す。平均割当て距離については、パターン 3 ではパターン 2 との差は大きな差は見られなかった。また、4 つの手法の間での特徴に変化が見られなかったため、移動経路間の距離を多少変更したとしても大きく性能に影響を与えないと考えられる。

6.1 モバイル機器の割当て順序

(4) パターン 4 の結果

移動経路パターン 4 の場合の最大割当て距離を図 6.12 に示す。到着順と要求リソース順では、高負荷状態において、最大割当て距離が 9 という他のどのパターンよりも大きい結果となった。また、混雑度順では、加算距離 2 も加算距離 4 も同じ結果となった。加算距離 4 の場合では、高負荷状態での最大割当て距離が 4 という結果になり、他のどのパターンよりも大きい結果となった。この移動経路パターンでは、モバイル機器がより密集して移動を行うため、負荷が集中し、最大割当て距離が大きくなったと考えられる。

平均割当て距離を図 6.13 に示す。平均割当て距離は、高負荷状態で、他のパターンと比較して割当て距離が 0.5 前後ほど大きいという結果になった。

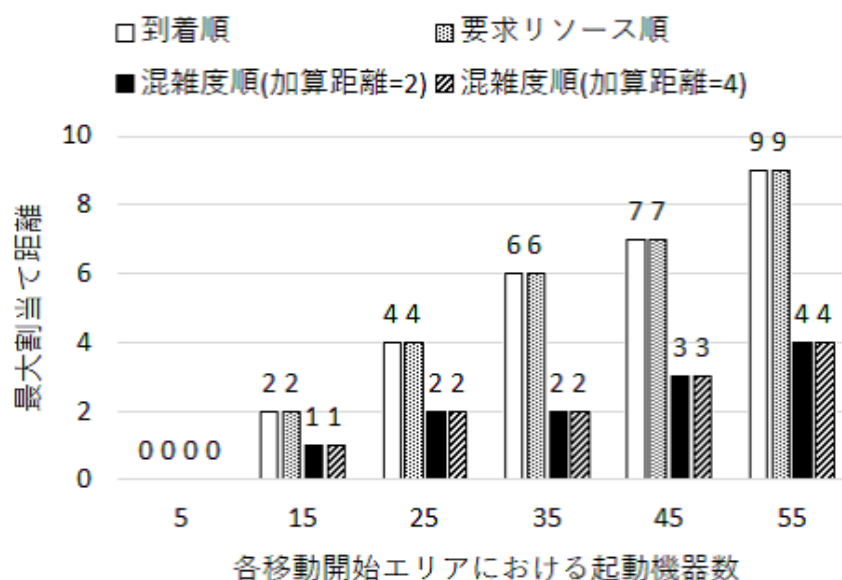


図 6.12 パターン 4 での最大割当て距離

6.2 Cloudlet サーバの選択方法

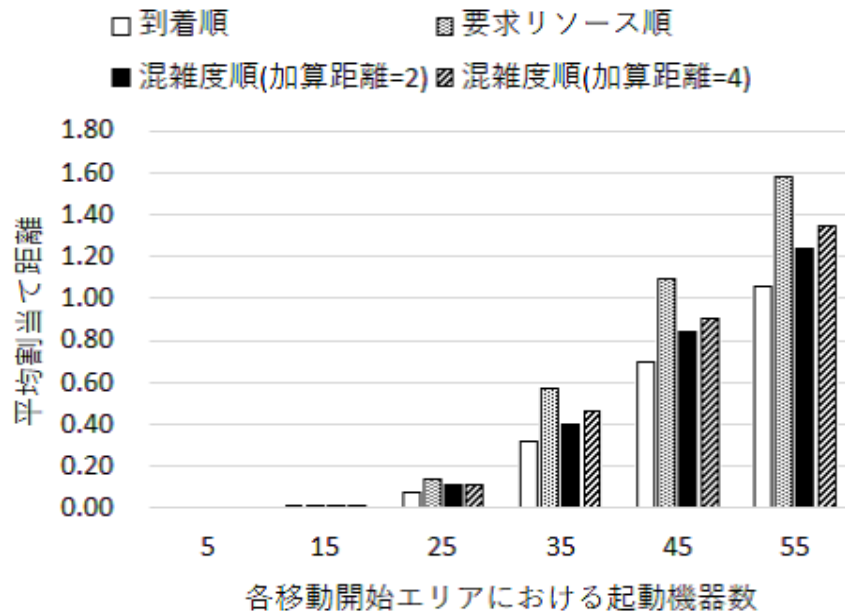


図 6.13 パターン 4 での平均割当て距離

6.2 Cloudlet サーバの選択方法

6.2.1 評価方法

ここでは、最近傍選択のみと、継続割り当て・最近傍選択、継続割り当て・移動経路優先選択の 3 つを比較する。これによって、継続割り当てによるアプリケーション起動処理の発生を抑制できることを確認し、移動経路優先選択によってその効果が高められることを確認する。また、継続利用や、移動経路優先選択は、モバイル機器から少し離れた位置にある Cloudlet サーバを優先して割当てるため、最近傍順と比較して最大割当て距離が増加する恐れがある。そのため、実際に最大割当て距離の増加が発生するかの確認も行う。入力されるモバイル機器集合と Cloudlet サーバ集合は 6.1.1 で用いたパターン 1 と同じものを利用する。継続割り当て距離は 2 に設定する。モバイル機器の割当て順序は、混雑度順を利用し加算距離は 4 に設定する。

評価対象は、アプリケーション起動発生率と、最大割当て距離である。アプリケーション起動発生率は、すべての Cloudlet サーバで発生したアプリケーション起動発生回数を、す

6.2 Cloudlet サーバの選択方法

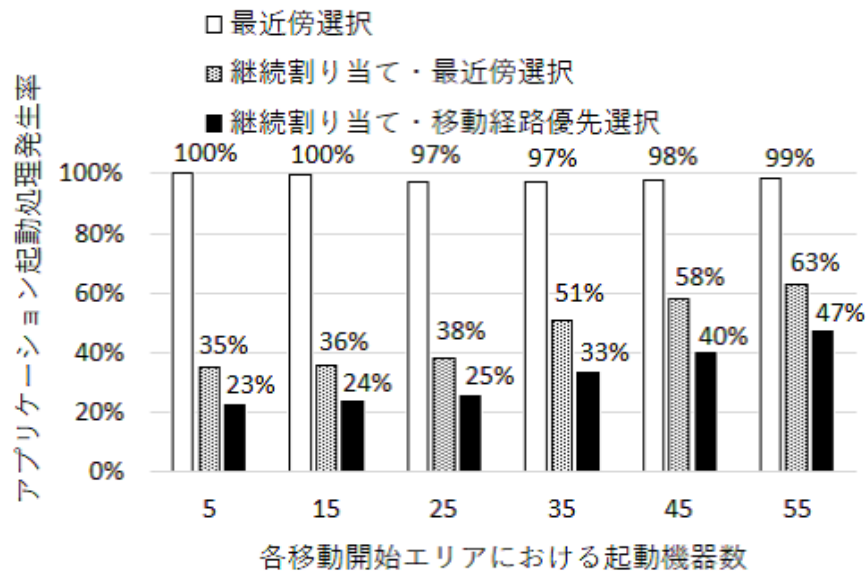


図 6.14 Cloudlet 選択方法の評価でのアプリケーション起動発生率

すべての割当て処理成立回数で割ったものとなる。割当て処理成立回数とは、モバイル機器に対して Cloudlet サーバの割当てが成立した回数である。また、参考として平均割当て距離も比較する。

6.2.2 評価

まず、アプリケーション起動処理発生率の評価を述べる。アプリケーション起動処理発生率を図 6.14 に示す。低負荷状態では、最近傍選択と比較して 70%から 80%ほど、アプリケーション起動発生率を削減できている。負荷が上がるにつれ、継続割り当て・最近傍選択で 63%、継続割り当て・移動経路優先選択で 47%まで増加するが、大きくアプリケーション起動発生率を削減できることがわかった。負荷が増加するとアプリケーション起動発生率が増加するのは、継続して割り当てたい Cloudlet のリソースがすでに他のモバイル機器に利用されているため割り当てることができない場合が増えるからだと考えられる。また、継続割り当てを用いている 2 つの手法を比較すると、移動経路優先選択を用いることにより、12%から 18%ほどアプリケーション起動発生率を削減できている。このため、継続割り当てを行うこと

6.2 Cloudlet サーバの選択方法

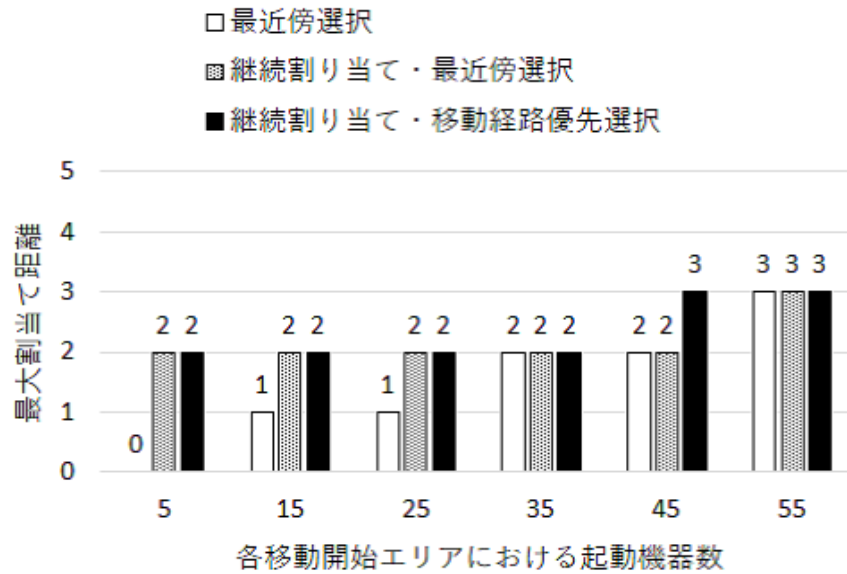


図 6.15 Cloudlet 選択方法の評価での最大割当て距離

によって、アプリケーション起動処理の発生を抑制し、Cloudlet サーバにかかる負荷を和らげることができると考えられる。また、継続割当てと移動経路優先選択を用いることにより、その効果をさらに増大させることができると考えられる。

次に最大割当て距離の評価を述べる。図 6.15 に最大割当て距離を示す。起動機器数 5 から 35 までの間、継続割当てを用いたものでは、最大割当て距離はすべて 2 となっている。しかし、継続割当てを用いていないものでは、0 から 2 までの間となっている。これより、低負荷状態では、継続割当てを行うことで、継続割当て距離として与えた 2 の分だけ最大割当て距離が増加していることがわかる。しかし、負荷が大きい状況では、3つの手法であまり大きな差は出ることにはなかった。このため、高負荷状態では、継続割当てによって特別大きな割当て距離が発生することはないと考えられる。起動機器数 45 を見ると、移動経路優先選択を用いた手法だけ最大割当て距離が 1 大きくなっている。これより、移動経路優先選択を用いることで、やや割当て距離が増加しやすいと考えられる。

平均割当て距離に着目すると、起動機器数 25 までの低負荷状態では、最近傍選択のみの場合、平均割当て距離は 0.1 以下であるのに対し、継続割当てを用いた場合、平均割当てよりが 0.9 以上になっている。負荷が増加するにつれ、最近傍選択のみの場合も、平均割当て

6.2 Cloudlet サーバの選択方法

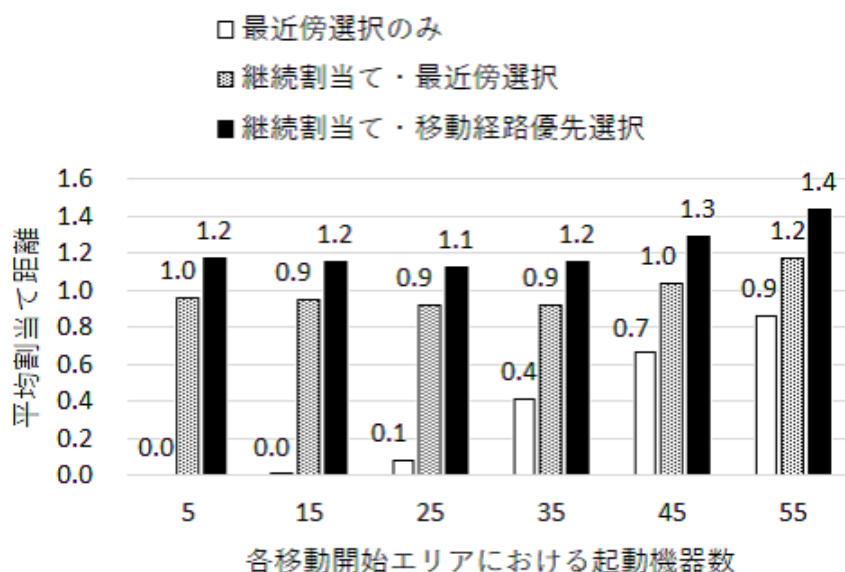


図 6.16 Cloudlet 選択方法の評価での平均割当て距離

距離が 0.9 まで増加するが、継続割当てを行う手法では、最近傍選択のみとくらべ、低負荷状態からあまり平均割当て距離が増加していない。また、起動機器数が 15 から 45 の場合、起動機器数が 5 のときと比べ平均割当て距離が少しだけ低下している。これは、継続割当て・最近傍選択では、継続割当てに失敗した場合、再度最近傍選択によって、近いものから割当てを行うためだと考えられる。また、継続割当て・移動経路優先選択では、継続割当てに失敗しても、移動経路から少し離れたモバイル機器の周囲の Cloudlet サーバを選択する場合があります、これによってモバイル機器により近い Cloudlet サーバが選ばれ、割当て距離が小さくなっていると考えられる。

第 7 章

おわりに

7.1 まとめ

近年、増加しているモバイル機器や、IoT 機器に対応するため、従来の一極集中型の処理ではなく、Cloudlet やエッジコンピューティングを用いた構成が研究されていることを述べ、また、それらには計算機資源の制約が問題となることを述べた。

その問題の解決方法として、ヒューリスティックなリソース割当てアルゴリズムで Cloudlet 環境でのリソース割当てを実現する手法を提示し、シミュレーション実験を行った。今回のシミュレーション実験により、モバイル機器の処理順序という点については、混雑度順にソートすることで、到着順や、要求リソース順と比較して大きく最大割当て距離を抑制することができることができた。割当てる Cloudlet サーバの選択という点については、継続割当てを行うことにより、モバイル機器に割当てられる Cloudlet サーバの変更を抑制し、アプリケーション起動処理を減らすことに成功した。また、移動経路優先割当てを行うことによりその効果を高めることに成功した。最大割当て距離を見ると、低負荷状態で継続割当て距離に指定した距離の分、最大割当て距離が増加するが、高負荷状態では、大きく変化することはなかった。

7.2 今後の課題

実際の環境では、Cloudlet サーバの所有リソース量は場所によって異なっていることが考えられる。しかし、現在のリソース割当てアルゴリズムはこのような Cloudlet サーバの

7.2 今後の課題

リソース量の偏りを考慮していないため、この点については改良する必要がある。

また、本稿では、各モバイル機器が利用するアプリケーションはすべての Cloudlet サーバ上でリソースが不足していなければ自由に起動することができた。しかし、実際の環境では、多種類のアプリケーションが動作し、Cloudlet サーバによって起動できるアプリケーションが限定される可能性が考えられる。そのようなモデルの拡張とリソース割当てアルゴリズムを改良を行う必要がある。

本稿では、混雑度順によるモバイル機器の割当て順序に使われる加算距離は事前指定により決定されていた。しかし実際の適切な加算距離は、モバイルモバイル機器の移動経路の偏りなどによって変化すると考えられる。そのため、加算距離を、負荷やモバイル機器の移動経路によって自動決定することができるような仕組みが必要である。

謝辞

本研究を行うにあたり，4年近くのだんご指導，ご鞭撻を頂きました指導教員であり主査の横山和俊教授に心より感謝致します。また，様々なご指摘を頂きました副査の高田喜朗准教授，松崎公紀准教授に心より感謝致します。共同で研究を行っていた福永昂輝君を始めとする研究室の皆様には，シミュレータの開発やリソース割当手法について大変参考になる意見を頂き心より感謝致します。

参考文献

- [1] 塩田純, 滝澤充, 田中裕之, 高橋紀之, 小林英嗣. 画像処理におけるエッジコンピューティングを用いた垂直分散処理方法の検討. 情報処理学会研究報告, Vol. 2016-DPS-167, No. 2, pp. 1–7, 2016.
- [2] 加藤由佳, 田中麻美子. 簡易ロボット遠隔ナビゲーションにおけるネットワーク遅延の影響評価. 情報処理学会研究報告, Vol. 2016-DPS-167, No. 19, pp. 1–7, 2016.
- [3] 田添聡士, 福永アレックス. 仮想マシン配置問題に対する厳密アルゴリズム. 情報処理学会研究報告, Vol. 2009-MPS-76, No. 11, pp. 1–7, 2009.
- [4] 網代育大. 仮想マシンのためのパッキングアルゴリズム. 研究報告システム評価 (EVA), Vol. 2010-EVA-33, No. 4, pp. 1–7, 2010.
- [5] 高橋里司, 竹房あつ子, 繁野麻衣子, 中田秀基, 工藤知宏, 吉瀬章子. 省電力化のためのマッチングに基づく仮想計算機パッキングアルゴリズム. 情報処理学会論文誌コンピューティングシステム, Vol. 5, No. 5, pp. 33–42, 2012.
- [6] 藪崎仁史, 中越洋, 村山耕一, 加藤崇利. クラウドにおける自律的な応答性向上に向けた広域スケーリング. 情報処理学会論文誌, Vol. 57, No. 2, pp. 562–572, 2016.
- [7] 鈴木俊裕, 張勇兵, 計宇生. 移動端末からクラウドサービスを利用するためのプロキシサーバ構築に関する研究. 情報処理学会研究報告, Vol. 2013-MBL-69, No. 15, 2013.
- [8] Rong Yu, Yan Zhang, Stein Gjessing, Wenlong Xia, and Kun Yang. Toward cloud-based vehicular networks with efficient resource management. *IEEE Network*, Vol. 27, pp. 48–55, 2013.