5L-08

# フォグ環境における過剰な割当てを抑制したリソース割当て手法

湯浅主基† 高橋竜一‡ 深澤良彰†

早稲田大学 基幹理工学研究科 情報理工・情報通信専攻†
茨城大学 理工学研究科 情報工学専攻‡

## 1. はじめに

フォグ環境では限定されたフォグ層のリソースを効率的に利用するアプローチの一つとして,到着したタスクをクラウド層とフォグ層へユーザの得られる総満足度が最大となるように割当てを行なっている.

しかしながら,現状の満足度モデルはユーザの設定した応答時間の上限のみを考慮したもの[1]が多く,ユーザの満足度をうまく反映できておらず,過剰なリソースの割当てにつながる恐れがある.また,ユーザの総満足度を最大化するアルゴリズムでは,フォグ層で実行しなくても,高い満足度を得られるユーザにフォグ層のリソースを割り当ててしまう無駄が発生する場合がある.特にリソースひっ迫時に,限られたリソースをこのようなユーザが占有することで満足度のばらつきが発生し,フォグ層のリソースの利用に公平性の観点で弱点がある.

そこで,本研究ではリソースの過剰な割当てを抑制するためにユーザの要求をより細かに反映した満足度モデルと,フォグ層のリソースひっ迫時にリソース利用の公平性を考慮した,ユーザ間の満足度の分散を抑えるリソース割当てアルゴリズムを提案する.

## 2. ユーザへの過剰な割当てを抑制する満足度モデル

現状の満足度モデルはユーザの設定した応答時間の期限に対して,期限内でより早く応答が返ることで高い満足度が得られるとしている.タスク$i$の応答時間の期限$deadline_i$に対して,応答時間$delay_i$で得られる満足度$P_i$は式(1)のように表される[1].

$$P_i = \begin{cases} \dfrac{delay_i}{deadline_i}, & if\ delay_i < deadline_i \\ 0, & otherwise \end{cases} \quad (1)$$

しかし,応答が早く返ることでより高い満足度が得られる遅延に敏感なタスクと,ある時間より早く応答が返っても満足度の変わらないタスクが存在する.たとえば,異常検知などのタスクはより早く応答が返ることで,より高い満足度が得られる.一方で,ドローンの操作などの早く応答が返っても期限内であれば,人間の感知上の問題などから満足度が変わらないタスクも存在する.

このように応答時間の上限のみを考慮した従来の満足度モデルは,クラウド層より早い応答を返しても満足度が変わらないユーザをフォグ層に割り当てることで,限定されたリソースを過剰に割当ててしまう恐れがある.

そこで,ユーザの求める満足度をより詳細に反映するために従来の応答時間の上限に加えて,遅延に敏感なタスクは最大満足度係数$maximum$,満足度の変わらないタスクは従来の上限より遅い満足度が 0 となる最大許容応答時間$tolerance$を考慮した満足度モデルを提案する.提案するモデルから算出される遅延に敏感なタスク$i$が得られる満足度$P_i$を式(2), 期限に敏感なタスク$j$が得られる満足度$P_j$を式(3)で表す.

$$P_i = \begin{cases} maximum * \dfrac{delay_i}{deadline_i}, & if\ delay_i < deadline_i \\ 0, & otherwise \end{cases} \quad (2)$$

$$P_j = \begin{cases} 1, & if\ delay_j < deadline_j \\ \dfrac{(delay_j - deadline_j)}{tolerance_j - deadline_j} & if\ \ delay_j < tolerance_j \\ 0, & otherwise \end{cases} \quad (3)$$

## 3. 過剰な割当てを抑制するリソース割当て手法

ユーザの満足度の総和を最大化する従来のアルゴリズムでは,フォグ層で実行しなくても,高い満足度を得るユーザにフォグ層のリソースを割り当ててしまう無駄が発生する場合がある.

例えば,フォグ層のリソースひっ迫時に,クラウド層の満足度に対するフォグ層の満足度の増加量が同等であるが,クラウド層の満足度に差のあるタスク A とタスク B が到着したとする.このとき,クラウド層の満足度が高いタスク A フォグ層に割り当ててしまうと,満足度の総和は最大化される一方で,タスク間の満足度のばらつきは大きくなってしまう.

このようにクラウド層で他ユーザより高い満足度を得られるユーザをフォグ層に割り当てることは,満足度のばらつきが大きくなり,リソース利用の公平性の観点で弱点となる.限定されたフォグリソースがひっ迫している場合は,一部のユーザが高い満足度を得るのではなく,多くのユーザが同程度の満足度を得ることがリソース利用の公平性の観点から望ましい.

そこで,リソースひっ迫時にリソース利用の公

Resource allocation method to control over-allocation in fog environment

Kazuki Yuasa† Ryuichi Takahashi‡ Yoshiaki Fukazawa†

Graduate School of Fundamental Science and Engineering, Waseda University†
Graduate School of Science and Engineering, Ibaraki University ‡

平性を考慮した，タスク間の満足度の分散を抑えるリソース割当てアルゴリズムを提案する．図 1 に提案手法の疑似コードを添付する．提案手法では遺伝的アルゴリズムを用いて，満足度の分散が小さいフォグサーバを利用するタスクの集合を探索する．まず，染色体として表現するフォグサーバを利用できるタスクをフォグ層に最大数割当てを行い，次にフォグサーバ内で満足度の総和を最大化するように割当てを行う．最後にフォグサーバ内で満足度の総和を最大化するように割当てを行った場合に得られたユーザ間の満足度の分散の逆数を適応度として，最も高い適応度を得る組合せを遺伝的アルゴリズムで探索する．

**Algorithm 1** Proposal Algorithm

```
N = Number of users
M = Number of servers {0: CloudServer, 1 ≤: FogServer}
for 1, N do
    Chromosome[i] = random(0, 1)
end for
repeat
    for 1, N do
        assignedServer[i] = 0
        if Chromosome[i] = 1 and Chromosome[i] can be assigned to any FogServer then
            assignedServer[i] = 1
        end if
    end for
    priorityUser← IF(assignedServer[i]=1)
Require:    assignedServer[i] ≥ 1    IF(i in priorityUser)
    assignedServer[i] = Server number when total user profit is maximized
    var = variance of profit among users
    fitness = 1/var
until Reach the generation specified by the Genetic Algorithm
Select the assignedServer of the best fitness Chromosome
```

図 1. 提案アルゴリズムの疑似コード

## 4. 評価

提案手法を表 1 のパラメータ [2] に従い，20 タイムスロットでシミュレーションを行い，満足度の総和最大化を行う手法と，満足度の低い順からフォグ層へ最大数割り当てる手法に対して，比較を行った． シミュレーション結果として，フォグ層のひっ迫状況を示す各タイムスロットにおけるクラウドに割り当てられたタスク数を図 2 に，割当てタスクの満足度の平均と分散を示したグラフを図 3, 4 として添付する．

表 1. シミュレーションで利用したパラメータ

| | |
|---|---|
| タイムスロットあたりのタスク到着数 | [2, 30] |
| タスクのメモリ要求量 | [50, 200] |
| タスクの要求実行量 | [100, 500] |
| タスクの要求応用時間 | [100, 500] |
| 遅延敏感ユーザの最大満足度 | 2.0 |
| 期限敏感ユーザの最大許容応答時間 | [101, 500] |
| フォグ層に存在するサーバ数 | 5 |
| フォグサーバのメモリ量 | [150, 250] |
| フォグサーバの単位当たり実行量 | [500, 2000] |
| クラウドサーバの単位当たり実行量 | 10000 |
| フォグ層への推定伝送遅延 | 5 |
| クラウド層への推定伝送遅延 | 250 |

結果として，提案手法はフォグ層がひっ迫しているタイムスロットにおいて満足度の分散を抑えながら高い満足度平均を得ることができた．特にタイムスロット 15 では提案手法の満足度の平均値が 1.19 であるのに対して総和最大化は 1.35 だが，分散の値は提案手法が 0.21 に対して，総和最大化が 0.52 となっており，11.9%の平均値の減少で大幅に分散を抑えることができている．また，タイムスロット 8 において，提案手法はクラウド層へ割り当てるタスクが増加した一方で，分散を抑えることができていることから，クラウド層で高い満足度を得るユーザによるフォグ層の利用を抑えていることをわかる．

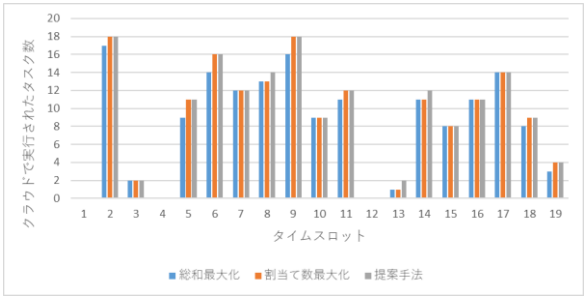


図 2. 各タイムスロットにおけるクラウド層で実行されたタスク数

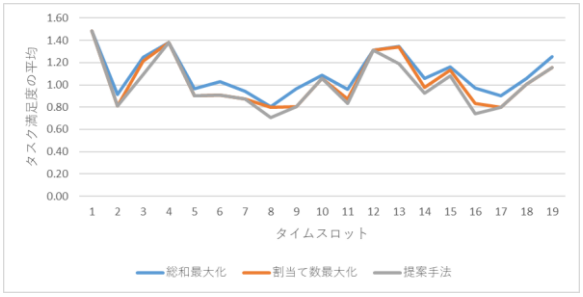


図 3. 各タイムスロットにおけるタスクの満足度の平均

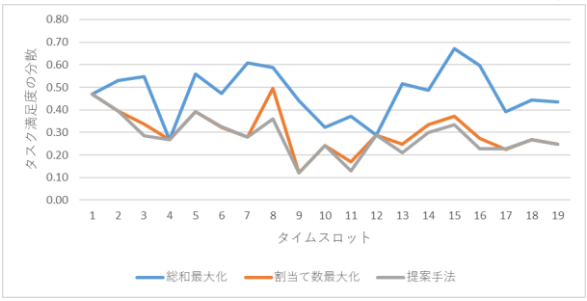


図 4. 各タイムスロットにおけるタスクの満足度の分散

## 5. おわりに

本研究では過剰なリソースの割当てを抑制するためにユーザの要求をより細かに反映する満足度モデルと，リソース割当て手法を提案した．結果として提案手法は満足度平均の減少に対して大きく分散を抑えることができた．

## 参考文献

[1] J. Li, W. Liang, W. Xu, Z. Xu, J. Zhao, Maximizing the Quality of User Experience of Using Services in Edge Computing for Delay-Sensitive IoT Applications, MSWiM '20:

[2] F. Hoseiny, S. Azizi, M. Shojafar, R. Tafazolli, Joint QoS-aware and Costefficient Task Scheduling for Fog-cloud Resources in a Volunteer Computing System. ACM Trans. Internet Technol. 21, 4, Article, June 2021