一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
NS2021-141(2022-03)

エッジネットワーク内の 通信・計算資源の効率・公平利用を両立する資源割当手法

秋吉　翔太† 　妙中　雄三†† 　塚本　和也†

† 九州工業大学大学 　〒820-8502 福岡県飯塚市川津 680-4
†† 奈良先端科学技術大学院大学 　〒630-0192 奈良県生駒市高山町 8916-5
E-mail: †akiyoshi.shota190@kyutech.ac.jp, tsukamoto@csn.kyutech.ac.jp, ††yuzo@is.naist.jp

あらまし　IoT 時代では，地域で得られる多様なデータを地域で活用する異分野データ連携が注目されている．我々はこれまでに，物理位置に基づいて構成したエッジネットワークで異分野データ連携を可能にする新しいデータ連携基盤として，Geo-Centric Information Platform（GCIP）を提案した．GCIP では，各地理範囲 (メッシュ) 内で IoT データを収集し，様々な事業者が各地理範囲内に設置したサーバを活用して，IoT データを組み合わせて時空間コンテンツ（STC）を動的に生成する．しかし 1 つの STC 作成に大量かつ多様なの IoT データとそのデータ処理が必要な場合，エッジネットワーク内特定の区間のネットワークと特定のサーバが過負荷となる一方で，その他のネットワーク・サーバは資源が余る状態となる．そこで本研究では，転送経路上で複数のサーバで分散してデータ前処理を行うことで全サーバ資源を効果的に活用する方策を提案する．しかし生成する STC に応じて必要なデータ転送量（通信資源）と計算量（計算資源）が異なるため，複数の STC を同時に生成する場合には，データ転送する経路と処理サーバ次第で (1) エッジネットワークで処理する合計スループットの低下（効率性の問題），(2) 資源配分の不公平 (公平性の問題) が生じる．そこで本研究では，通信・計算資源を合わせた合算資源に基づいた柔軟な資源割当を提案し，効率性と公平性の両立を実現する．数値検証の結果，提案手法は効率性・公平性において，それぞれ最大 26%,38% 改善できることを明らかにした．

キーワード　資源配分，IoT，異分野データ連携，地理指向

A resource allocation method of network and computational resources in edge networks

Shota AKIYOSHI†, Yuzo TAENAKA††, and Kazuya TSUKAMOTO†

† Kyusyu Institute of Technology
†† Nara Institute of Science and Technology
E-mail: †akiyoshi.shota190@kyutech.ac.jp, tsukamoto@csn.kyutech.ac.jp, ††yuzo@is.naist.jp

Abstract　Cross-domain data fusion is becoming a key driver to growth of the numerous and diverse applications in IoT era. We have proposed a concept of new information platform, Geo-Centric information platform (GCIP), that enables IoT data fusion based on geolocation. GCIP produces new and dynamic contents by combining cross-domain data in each geographic area and provides them to users. However, when a large amount of IoT data is required for STC creation, it places a heavy load on the network and computation resources in the mesh. In order to reduce this load, we propose to pre-process the data in advance on the transfer path. However, when flows with different loads on network and computation resources flow into a mesh with multiple paths, depending on the allocation of network and computation resources, (1) throughput degradation (efficiency issues) and (2) inequity in resource allocation (fairness issues) may occur. In this study, we propose a flexible resource allocation method that efficiently and fairly utilizes both communication and computation resources in preprocessing to achieve both efficiency and fairness. As a result of numerical verification, we found that the proposed method can improve efficiency and fairness by up to 26% and 38%, respectively.

Key words　Resource allocation, IoT, Cross-Domain Data Fusion, Geolocation

1. 研究背景，課題

近年，人々の生活圏内のヒト／モノ等から得られる多種多様な異分野データ（IoT データ）を連携し [1]，データが生成された地域で処理・活用するデータの地産地消の実現が期待されている．我々は，これらを実現する基盤のひとつとして，地理空間を意識して構成したエッジネットワークにおいて，データの収集，処理，配信を地域内で行う地理指向情報プラットフォーム (Geo-Centric Information Platform: GCIP) [2] を提案している．

GCIP では，地理空間を階層構造のメッシュに分割し，一意のコードを割り当てる (図 1)．このコードを IP アドレスに含めて経路制御することで，メッシュ内で発生した全てのデータを，メッシュ内に設置したデータストア (DS) サーバに収集する [3]．各メッシュ内に事業者が設置した異分野連携 (DF) サーバが，メッシュ単位で収集された IoT データに対し，様々な前処理 [4] とデータ解析を行い時空間コンテンツ (STC) を生成し，地域内のユーザへ提供する．

このエッジネットワークでは，ネットワーク（通信）資源と計算資源が限られるため，それらの資源を効率的に活用しながら STC を生成する必要がある．STC 生成に必要な前処理や解析，コンテンツ生成処理を 1 台の DF サーバで実行すると，エッジネットワーク内の一部の DF サーバのみ高負荷となり，全ての計算資源の有効利用とならない．そこで先行研究 [5] では，計算資源の効率利用のために，DF サーバが DS サーバから IoT データを取り寄せるデータ転送途中に，メッシュ内の計算資源に余裕のある他の DF サーバを前処理 (DP) サーバとし，前処理に活用する．つまり，計算資源に余裕のある 1 つ以上の DF サーバにおいて STC 生成に必要となる (複数の) 前処理を IoT データの転送途中に実施することで効率化を図っている．

しかし，複数の STC を並行して生成する場合には，生成する STC に応じて前処理に必要な通信資源と計算資源の組み合わせが様々となるため，通信・計算資源両方の効率利用が困難となる．例えば，必要なデータ量が大きいものの，前処理負荷は少ない場合や，必要なデータ量が少ないものの，前処理負荷が大きい場合がある．このような通信負荷と計算負荷が異なる複数の前処理を転送途中に実施すると，競合によって通信資源と計算資源のどちらかがボトルネックとなり，資源の最大限の活用が困難となる．その上，データ転送経路と資源配分次第で (1) エッジネットワークで処理する合計スループットの低下 (効率性の低下)，(2) 資源配分の不公平が生じる事になる．

そこで本研究では，通信・計算資源を効率かつ公平に利用する柔軟な資源割当手法を新たに提案する．STC 毎に必要な通信資源と計算資源の量が異なることを利用して，通信・計算資源を合わせた合算資源に基づいて資源割当を行うことで効率性と公平性の両立を実現する．それによって，(1) メッシュ内全体での転送スループットの改善 (効率性) かつ (2) 要求計算資源に依らない公平な資源割当 (公平性) を実現する．

本論文の構成は以下の通りである．2 節では資源割当手法に関する関連研究を紹介し，3 節では提案手法を述べる．4 節では検証内容と検証結果，考察について説明し，5 節でまとめと今後の課題について述べる．

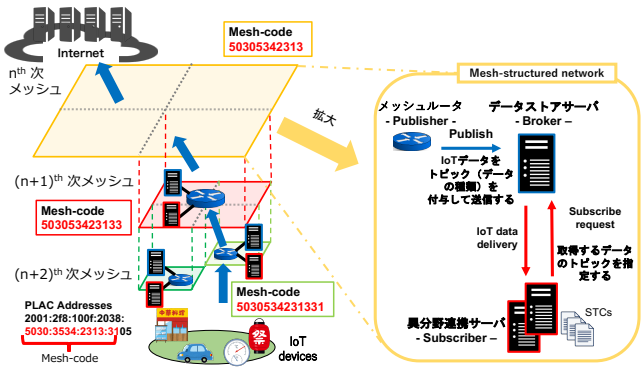

図 1　GCIP のデータ流通のイメージ

2. 関 連 研 究

ユーザが所持するモバイル機器の計算資源の制限を克服するために，計算処理をエッジネットワークにオフロードする研究が多くなされている．しかしエッジネットワーク上の計算資源は，クラウドネットワーク上の計算資源よりも小さいため，限られた計算資源を効率的に利用するための動的な資源割当が必須となる．この資源割当を実現するための基本的な考え方は，通信資源と計算資源のトレードオフの調整といえる．

文献 [6] [9] では，コントローラーがエッジノードの利用可能な計算資源をデータ転送前に事前に把握し，各サービス要求に対して最短の処理遅延を実現する計算資源を持つサーバにおいて処理を実施する資源割当手法が提案された．文献 [7] では，通信・計算遅延及び解析精度の多目的最適化のための資源管理を提案している．文献 [8] [10] では，複数のエッジネットワークを対象に，タスクの分散処理のためのスケジューリングアルゴリズムを提案している．しかしこれらの手法では，「通信資源は公平に共有され，計算資源は First-Come-First-Serve(バッチ処理，シングルタスク処理) による共有」を前提としている．その結果，新規フローがエッジネットワークに流入する場合，当該フローに必要な計算処理が小さくても，エッジネットワーク内で処理中の計算負荷の大きなフローの処理完了待ち時間の増大によって，当該フローの処理完了までの時間が増大する performance anomaly 問題が生じる．

それに対し，本研究では DS サーバと DF サーバ間で前処理を実施し，DP サーバがマルチタスク処理でフロー単位で計算資源を配分する上，ネットワーク上での競合を考慮した通信資源配分を合わせて柔軟な資源管理を行う．この資源管理によって，競合フローの種類によらず，効率的な通信・計算資源管理が可能となる．

3. 提 案 手 法

3.1 エッジネットワーク内での前処理の手順

本節では，データ転送経路中での前処理手法の手順を説明する．本提案では，DS サーバ，DF サーバ，また計算負荷の低い DF サーバの中から選出された 1 台の DP サーバが前処理を 1

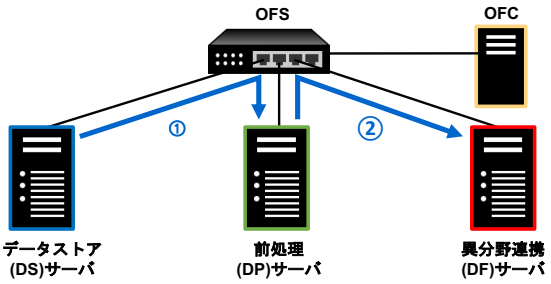

図 2　提案手法の概念図 (物理トポロジ)

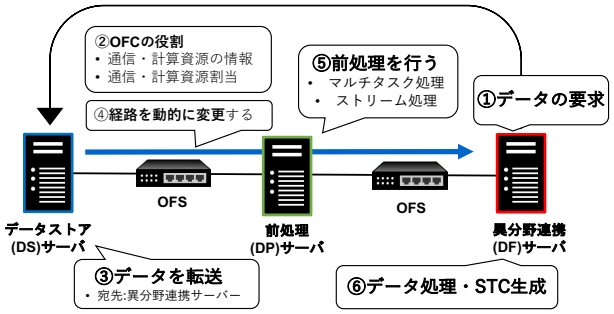

図 3　提案手法の概念図 (論理トポロジ)

回実行することを前提とする (図 2)．また，DP サーバは前処理以外に計算負荷がかかっていないものとする．なお本稿では，複数の DP サーバや複数回の前処理は対象外とする．図 2 の物理トポロジを，論理トポロジにしたものを図 3 に示す．本研究では，これらのサーバは同一のメッシュ (サブネット) 内に存在し，次の手順でデータの転送と処理を行う．

（1） DF サーバが DS サーバに対し，STC 生成に必要なデータをまとめて要求する．同時に必要な前処理も通知する．

（2） SDN コントローラーが通信・計算資源の情報を取得し，資源割当を決定/配分する．

（3） DS サーバは要求されたデータを 1 つの通信フローでまとめて DF サーバに送信する．

（4） スイッチが (2) で決定した通信資源・経路を経由して DP サーバへ通信フローを送信する．

（5） DP サーバが配分した計算資源を用いてデータの前処理を行う．

（6） DF サーバが受信データを処理し，STC を生成する．

上記の過程で，エッジネットワークの通信・計算資源の効率・公平利用を行う．以降の節では，（2）の資源割当について説明する．

3.2　前処理 (DP) サーバの前提条件

DP サーバは，簡単化のため CPU はシングルコア・マルチスレッドなし，計算資源は前処理以外に利用されないことを前提とする．また，複数の前処理を実行する場合には，マルチタスク実行して並列処理する．この時，CPU スケジューリングは一般的な Linux で用いられる CFS[11] に従う．CFS では，CPU のパワーを実行可能な全プロセスに公平に分配して，複数のプロセスを同時に実行する．また DP サーバではストリーム処理を行い，データ受信と並行して処理を実施する．

3.3　コントローラ型の資源管理

本節では，動的に柔軟な資源割当に必要な要件を整理し，想定する利用技術について説明する．まず，本研究で提案する資源割当は，ネットワークの物理区間毎の情報と DP サーバごとの情報を合わせて考慮するため，「ネットワークの物理区間毎の物理帯域幅と利用帯域幅の情報を収集できる」，「DP サーバ単位の CPU 計算資源量と利用計算資源量を収集できる」，「2 つの情報を同期して，まとめて利用できる」の 3 つの要件を満たす必要がある．

本研究では，以上の条件を満たす技術として，コントローラ型で集約してネットワークの経路制御を柔軟に行える Software Defined Network(SDN) を用いる．具体的には，SDN 実装の 1 つである OpenFlow1.3 [12] を用い，ネットワークの情報を 1 台のコントローラ (OFC) で一括して取得し，さらに OFC にサーバの情報収集・資源制御を行う拡張機能の追加を想定する．また 1 台の OFC へ情報・制御集約することで，通信・計算資源を統合した資源配分の決定と，経路（通信資源）制御・計算資源制御を同時に実行可能となる．OFC が決定した経路をスイッチ (OFS) に通知することで，DF サーバ宛のパケットを DP サーバを経由して転送可能となる．

3.4　提案：通信・計算資源の動的割当手法

本節では，通信・計算資源配分量を決定するアルゴリズムについて説明する．本研究では，通信・計算資源の両方を合算して 1 つの資源として扱い，効率・公平利用を両立した資源割当手法を実現する．

3.4.1　効率・公平な資源割当の定義

効率・公平な資源割当をそれぞれ次のように定義する．効率的な資源割当とは，あるフロー j が DF サーバへ到達するスループット ($Sum\text{-}Th_j$) が最大となる時の資源割当と定義する．公平な資源割当とは，通信資源 (100%) と計算資源 (100%) を合算した時の，各フローが獲得した資源割合が同じであることと定義する．

3.4.2　効率・公平のパレート最適点

3.4.1 節で説明した公平性と効率性はトレードオフの関係にあるため，一方の効用 (満足) を犠牲にしなければ，もう一方の効用を高めることができない状態である (パレート最適) 点が複数存在する場合がある．例えば，2 本のフローの要求計算資源が DP サーバの CPU 資源より (1) 極めて小さい/(2) 極めて大きい組み合わせを考える．この時，効率 (エッジネットワークで前処理・出力する合計スループット) を最大にする場合，(1) のフローに全ての通信・計算資源割当を行うと良いことは明らかである．しかし，(1) のフローが資源を独占しているため公平性は損なわれる．一方で公平性を最大化 (合算資源の均等配分) する場合，(2) のフローにも計算資源を配分することになるため，エッジネットワークが処理する合計スループット（効率性）は低下してしまう．以上のように，効率・公平のパレート最適となる組合せが複数存在するため，本研究では，複数のパレート最適点の中から選択する 1 点に向けて最適化する手法を提案する．なお，最適点の選択はエッジネットワークの運用ポリシから決定されるものであり，本研究ではあらかじめ与えられるも

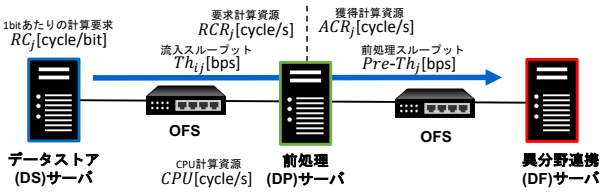

図 4　各数式の関係性を示す概念図

のとする．

3.4.3　資源利用量の算出方法

本節では DP サーバに複数フローが流入する際の，通信・計算資源利用量の算出方法について説明する．まず事前に，OFC は前処理サーバの計算機資源の性能である CPU 周波数 [cycles/s] や CPU コア数，GPU の有無などの情報を取得する．さらに各前処理サーバで実行可能な前処理の種類やトポロジの構成についても取得する．また資源割当のために，OFC は通信資源の情報として，物理帯域幅 ($P\text{-}BW$)[bps]，利用帯域幅 ($A\text{-}BW$)[bps] を，計算資源の情報として，CPU 計算資源 (CPU 周波数 : CPU)[cycles/s]，利用計算資源量 [cycles/s] を取得する．

n 本のフローに対する通信資源割当の組合せ数が i[組] 存在する時，0〜n 本目までのそれぞれのフロー j における 1bit あたりの計算要求を RC_j[cycles/bit]，通信資源割当を i の組合わせで行った時の，フロー j の DP サーバに流入するスループットを Th_{ij}[bps] とする．このとき，DP サーバで要求する計算資源 (要求計算資源 : RCR_j)[cycles/s]，DP サーバで実際に獲得する計算資源 (獲得計算資源 : ACR_j)[cycles/s] は式 1，式 2 となる．フローごとの要求計算資源に対して，DP サーバの残余計算資源が不足している場合は，実際の獲得計算資源は減少する．この時，DP サーバから流出するスループット (前処理スループット : $Pre\text{-}Th_j$) は次のような式で計算できる．また図 4 に，各数式の関係性を示す．

$$RCR_j[cycles/s] = Th_{ij}[cycles/bit] \times RC_j[bit/s] \quad (1)$$

$$ACR_j[cycles/s] = CFS \text{ アルゴリズムに従う} \quad (2)$$

$$Pre\text{-}Th_j[bps] = T_{ij}[bps] \times \frac{ACR_j[cycles/s]}{RCR_j[cycles/s]} \quad (3)$$

ここで CFS による CPU 計算資源の共有について説明する (式 2)．CPU[cycles/s](=[Hz]) の CPU 計算資源を持つ 1 台の前処理サーバに到着する n[本] のフローの要求計算資源の合計値 ($SRCR[cycles/bit]$) は $\sum_{j=1}^{n} RCR_j$ となる．この時，次のように資源が配分される．

- $SRCR \leq CPU$ の場合

$$ACR_j = RCR_j \quad (4)$$

- $SRCR > CPU$ の場合

$$ACR_j = CPU[cycles/s] \div n[\text{ 本 }] \quad (5)$$

  – ただし，$ACR_j > RCR_j$ の場合，残余計算資源 ($ACR_j - RCR_j$) は他フロー $(n-1)$[本] が公平に獲得する．

次に具体例を用いて説明する．100[Mcycles/s](=[MHz]) の CPU 計算資源を持つ 1 台の前処理サーバにおいて，それぞれ (1)8[Mcycles/Mbit] と (2)16[Mcycles/Mbit] の計算要求を持つ 2 本のフローが $Th_{i1}, Th_{i2} = 5[Mbps]$ で流入する場合を考える．それぞれの要求計算資源は，式 1 よりそれぞれ (1)40[Mcycles/s],(2)80[Mcycles/s] となる．この場合，$SRCR > CPU$ となるため，式 5 に従い，ACR=100[Mcycle/s]/2=50[Mcycles/s] を獲得するが，本来は 50% (50Mcycles) ずつ共有することになるが，(1) は $RCR = 40[Mcycle/s]$ のため，残余計算資源 (50-40Mcycls/s) は (2) が使う．したがって，獲得計算資源は (1)40[Mcycle/s], (2)60[Mcycle/s] となる．この時,(2) の計算要求に対して不足する 20[Mcycles/s] の処理はバッファに蓄積されていくことになる．この場合，式 3 より，前処理スループット ($Pre\text{-}Th_1$,$Pre\text{-}Th_2$) は，それぞれ次の通りとなる．

$$Pre\text{-}Th_1 = 5[Mbps] \times \frac{40[Mcycles]}{40[Mcycles]} = 5[Mbps]$$

$$Pre\text{-}Th_2 = 5[Mbps] \times \frac{60[Mcycles]}{80[Mcycles]} = 3.75[Mbps]$$

ここでは (2) のフローは，$RCR > ACR$ となるため，$Th_2 > Pre\text{-}Th_2$ となり，前処理スループットが低下している．

3.4.4　通信・計算資源割当アルゴリズム

以下に資源割当アルゴリズムを示す．

STEP1　OFC が通信・計算資源の情報を取得する

STEP2　フローごとの通信資源の割当組合せの算出範囲（探索する i の範囲）の決定

- 局所探索 (初期解：要求計算資源の逆比) を実施

STEP3　STEP2 で決定した範囲において，以下の値を算出

- 式 6 に従い，前処理スループットの合計値 ($Sum\text{-}Th_j$)
- 通信・計算資源利用率 ($Co\text{-}RU_j$，$Cu\text{-}RU_j$) をそれぞれ合計した値である統合資源利用率 ($INT\text{-}RU_j$：式 9) の標準偏差 ($SINT\text{-}SRU_j$：式 11)

STEP4　算出した 2 値間のパレート最適点から 1 点を選択

式 10 に統合資源利用率の平均 ($AINT\text{-}SRU_j$) を示す．

$$Sum\text{-}Th_j[bps] = \sum_{j=1}^{n} Pre\text{-}Th_j \quad (6)$$

$$Co\text{-}RU_j[\%] = \frac{A\text{-}BW_j[bps]}{P\text{-}BW[bps]} \times 100 \quad (7)$$

$$Cu\text{-}RU_j[\%] = \frac{ACR_j[cycles/s]}{CPU[cycles/s]} \times 100 \quad (8)$$

$$INT\text{-}RU_j[\%] = Co\text{-}RU_j + Cu\text{-}RU_j \quad (9)$$

$$AINT\text{-}RU_j[\%] = \frac{1}{n} \sum_{j=1}^{n} INT\text{-}RU_j \quad (10)$$

$$SINT\text{-}RU_j[\%] = \sqrt{\frac{1}{n} \sum_{j=1}^{n} (INT\text{-}RU_j - AINT\text{-}RU_j)^2} \quad (11)$$

4.　性 能 評 価

4.1　実 験 環 境

本検証では，図 5 のトポロジを使用する．検証における前提を表 1 に示す．この環境において，計算要求を静的に 1~30[Mcycles/s] の範囲で与えた 3~5[本] のフローを流入させるシナリオを 100 パターン作成し，効率性と公平性の両立の可能性に関して数値検証を行う．

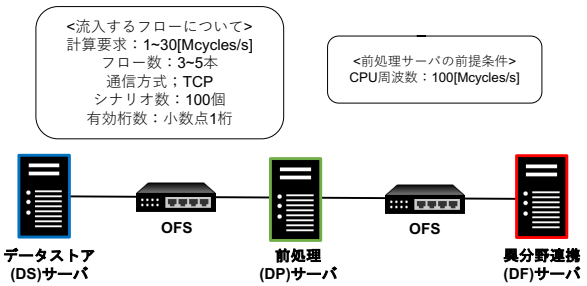

図 5　検証トポロジ

表 1　検証における前提

情報収集に用いる技術	SDN(OpenFlow)
要求計算資源	1~30[Mcycles/Mbit](静的に仮定)
フロー数	3~5[本]
実験回数	100
前処理サーバの CPU 周波数	100[Mcycles/s(=MHz)]
前処理回数	1 回
帯域幅	10[Mbps](Full-duplex)
通信の種類	TCP(1 フレーム:1480byte)
データサイズについて	前処理前後で変化しない
前処理サーバでの処理方式	ストリーム処理

4.2　評 価 指 標

本研究では 2 つの評価指標を用いる．1 つ目はシステムの効率性に関する指標として，前処理スループットの合計値 ($Sum\text{-}Th_j$)[bps](式 6) を用いる．2 つ目はフローごとの公平性に関する指標として，資源利用率の標準偏差 ($SINT\text{-}SRU_j$)[%](式 11) を用いる．

4.3　比 較 手 法

手法 1：通信資源のみを公平分割する手法

本手法は，データ転送に TCP を用いる事を想定しており，通信資源を公平分割する．この時，DP サーバにおける RCR[cycles/s] の比は，RC[cycles/bit] の比と同じ値になる．

4.4　結果・考察

図 6，図 8 に，システム内に流入するフロー数が 3 本の場合の前処理スループットの累積分布図と箱ひげ図を示す．図において，提案手法におけるパレート最適点のうち，公平性最大となる点を Maximum Fairness，効率性最大となる点を Maximum Efficiency と示す．またこの 2 点間に存在するパレート最適点において，フローごとの資源利用率の標準偏差 (公平性) が x% まで異なることを許容する提案手法を，Proposed(x%) と示す．

図 6 より，提案手法は，パレート最適点におけるどの点を選択しても，比較手法より約 2%～10% ほどスループットを改善できることがわかる．また図 8 より，提案手法は，比較手法より中央値で，約 7%～24% 前処理スループットの値が高いこともわかる．また 1 つのシナリオにおける，提案手法と比較手法の前処理スループットの差は，瞬間的に最大約 26%(公平性最大時)～45%(効率性最大時) 生じることがわかった．また，常にトラヒックが流通する場合は時間経過に応じて前処理スループットの差分が累積し大きくなる．本結果のように，1[Mb/s] の差分がある場合，3600[s] で 3600[Mbit](=450Mbyte) の転送量に差が生まれる．これは 1[Gb/s] の帯域幅に換算すると，360[Gbit](=45Gbyte) の差となり，瞬間的な差が累積することで，非常に大きな差が生じる．

この結果の要因は，提案手法の通信・計算資源の両方を考慮

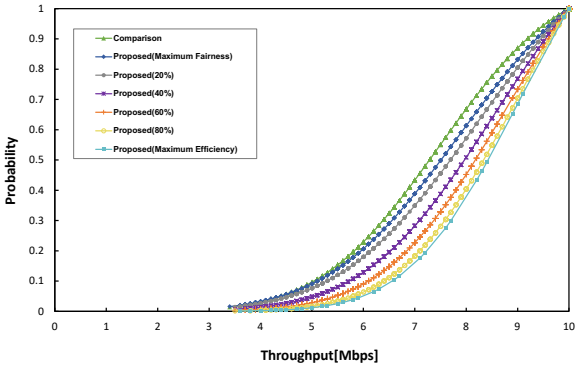

図 6　スループットの累積分布図 (フロー数:3)

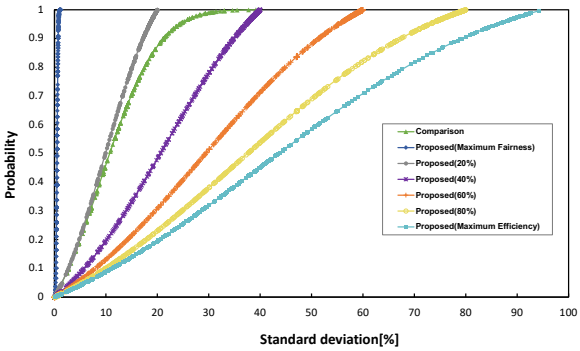

図 7　資源利用率の標準偏差の累積分布図 (フロー数:3)

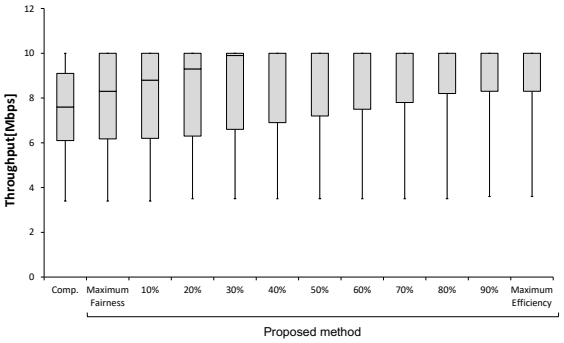

図 8　スループットの箱ひげ図 (フロー数:3)

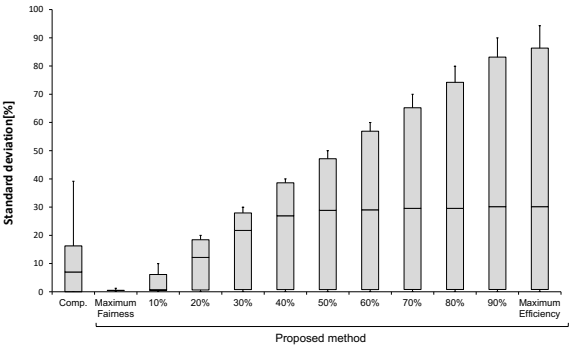

図 9　資源利用率の標準偏差の箱ひげ図 (フロー数:3)

する資源割当によって，要求計算資源に応じた通信資源を割り当てる事が可能となり，前処理が効率化できるためである．一方で比較手法は，要求計算資源の大小にかかわらず，通信資源を均等分割するため，計算資源が逼迫し，前処理スループットが大きく減少している．以上の結果から，提案手法は比較手法よりも，効率性の高い資源割当が可能であると言える．

図 7 より，提案手法の公平性最大点と資源利用率の標準偏差が 20% まで異なることを許容する Proposed(20%) は，比較手法よりも資源利用率の標準偏差の差分が約 18%〜39% 小さいことがわかる．また図 9 より，最大で約 38% ほど提案手法の方が資源利用率の標準偏差の差分が小さいこともわかる．効率性を高めることで，比較手法よりも資源利用率の標準偏差の差分が大きい場合もある．以上の結果より，提案手法は，比較手法よりも効率・公平利用を両方する資源割当が行えていることから，有用性があると言える．

また提案手法は，図 8，図 9 より，効率性と公平性の間でトレードオフの関係にあることがわかる．公平性最大となる点の前処理スループットと資源利用率の差の平均値は，それぞれ 7.9[Mbps]，0.5[%] である．一方で効率性最大となる点の前処理スループットと資源利用率の差の平均値は，それぞれ 9.1[Mbps],40.3[%] である．このように提案手法では，パレート最適点から利用者の好みに応じて最適な点を選択することができることから，柔軟性の高い資源割当が可能であると言える．

5. ま と め

地産地消の異分野データ連携基盤においては，地域の限られた計算資源を用いて，絶え間なく発生する IoT データを効率的に処理する必要がある．しかし通信・計算資源の需要が異なるフローがメッシュ内に流入する場合，資源割当次第で効率性・公平性の低下が生じる．そこで，本研究では通信・計算資源効率・公平利用を両立する計算資源割当手法を提案した．数値検証の結果，提案手法は効率性・公平性において，それぞれ最大 26%,38% 改善できることを明らかにした．また，パレート最適点から利用者の好みに応じて最適な点を選択することができる柔軟性の高い資源割当が可能であることも示した．今後はフロー数の増加に伴う効率的な計算資源割当の限界を考慮し，前処理サーバの台数を増加させた場合の資源割当と複数の前処理サーバに対する経路選択を行うアルゴリズムの提案を行い，有効性の評価をする．

謝　　辞

本研究成果は，JSPS 科研費 JP21H03430 の委託研究により得られたものです．ここに記して謝意を表す．

文　　献

[1] A. Al-Fuqaha, et al. “Internet of Things: A Survey on Enabling Technologies, Protocols and Applications,” IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347-2376, June 2015.

[2] K. Tsukamoto, et al. “Geolocation-centric Information Platform for Resilient Spatio-temporal Content Management,” IEICE Trans. Commun. , Online ISSN 1745-1345, Print ISSN 0916-8516, Sep. 2020.

[3] K. Nagashima, Y. Taenaka, A. Nagata, K. Nakamura, H. Tamura, and K.Tsukamoto,“Experimental Evaluation of Publish/Subscribe-based Spatio-Temporal Contents Management on Geo-Centric Information Platform,” Advances in Networked-based Information Systems, Advances in Intelligent Systems and Computing, vol 1036. Springer, Cham, 2020, pp 396-405.

[4] Salvador Garcia, Sergio Ramirez-Gallego,“Big data preprocessing: methods and prospects."Big Data Analytics, December 2016

[5] 長尾 健太郎他，“多種多様な時空間コンテンツ生成のための異分野データ収集処理手法の提案，”信学技報, NS2018-209,pp. 93-98, 2019 年 3 月

[6] M. Amadeo, et al.“SDN-managed Provisioning of Named Computing Services in Edge Infrastructures，” IEEE Trans, Aug. 2019.

[7] Q. Liu, et al.“An Edge Network Orchestrator for Mobile Augmented Reality，” IEEE INFOCOM, April. 2018

[8] H. Tan, Z. Han, X.-Y. Li, and F. C. Lau, “Online Job Dispatching and Scheduling in Edge-Clouds,”in IEEE INFOCOM 2017, Atlanta, GA, USA, May 2017, pp. 1–9.

[9] L. Tong, Y. Li, and W. Gao,“A Hierarchical Edge Cloud Architecture for Mobile Computing,” in IEEE INFOCOM 2016, San Francisco, CA, USA, April 2016, pp. 1–9.

[10] M. Jia, W. Liang, Z. Xu, and M. Huang,“Cloudlet Load Balancing in Wireless Metropolitan Area Networks,” in IEEE INFOCOM 2016, SanFrancisco, CA, USA, April 2016, pp. 1–9.

[11] “Completely Fair Scheduler”，https:www.belbel.or.jp/opensuse-manuals-ja/cha-tuning-taskscheduler.html

[12] OpenFlow Switch Specification Version1.0.0 “https://www.opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.0.0.pdf”

[13] 谷口 功 (2014),“SDN/OpenFlow のしくみ”，ソシム