ICN を用いた In-Network Computing のタスクスケジューリング

Task Scheduling for ICN-based Autonomous In-Network Computing

新部 裕樹[1]　　上山 憲昭[2]
Yuki Niibe　　Noriaki Kamiyama
立命館大学大学院 情報理工学研究科[1]
Graduate School of Information Science and Engineering, Ritsumeikan University
立命館大学 情報理工学部[2]
College of Information Science and Engineering, Ritsumeikan University

1. まえがき

昨今では，5G のような高速大容量ネットワークの実現により，IoT デバイスが急速に普及してきている．そのような中で，センサーや各種ハードウェアを搭載した IoT デバイスからセンシングされる膨大なデータを，連携させながら処理することで利用したいといった需要が高まっている．しかし，このような需要を遠隔地にあるクラウドで実現することは効率性に問題がある．そのため，これらデータに対する処理を転送と同時に行う，ネットワーク内処理という手法が検討されている．

このようなネットワーク内処理において，多種多様な連携処理に対する需要を満たすためには，効率よく柔軟に処理を連携し実行できる，汎用的な手法が求められる．そのため，情報指向ネットワーク (ICN) を組み合わせた，効率的かつ柔軟なサービスファンクションチェイニング (SFC) が考えられている．その中でも，有向非巡回グラフ (DAG) 構造を想定した ICN での SFC が最も汎用的な処理フローを実現でき，アプリケーション構造の複雑化にも対応できる [1].

しかしながら，このような自律分散型の処理環境で DAG 構造のアプリケーションを実行することは，同一タスクが複数のノードで重複して割り当てられる可能性があり，計算資源利用の観点から問題がある．DAG 構造の処理フローを ICN で実現する場合，一般に Exit タスクから Start タスクに向かって処理結果への要求を送信し，各タスクの実行ノードを確定させていく必要がある．このとき，各タスクは複数の先行タスクを依存関係として持つ場合がある．タスクを実行するためには全ての先行タスクから処理結果を受け取る必要があるため，タスクがノードに割当られた後，更にその先行タスクへ処理結果の要求を行う．ここで，処理結果の要求先は各ノードの FIB(経路表) から選択される．そのため，例え同じ先行タスクに対する処理結果の要求だとしても，要求元のノードによっては異なる場所へ要求が送信される可能性がある．自律分散環境において，どの要求がどこへ送信されているのかについて同期を取ることは困難である．その結果，同一のタスクが複数ノードに重複して割り当てられ，冗長なタスク実行となる．

そこで本研究では，タスクの重複割当を回避するため，DAG 上のタスクを一筆書きのように要求しノードへ割り当てるタスクスケジューリング手法を提案する．本手法のために，ReadyList を用いた段階的なタスクの順序付けと，未解決の要求を束ねて転送する方法を導入する．

2. 提案手法

既存手法では，各タスクが先行タスクに対して一斉に要求送信することでタスクが重複割当されていた．そこで本研究では，既存手法 (以下 multicast 的手法) とは異なり，必ず 1 つずつ先行タスクへ要求を送信することを考える．具体的には，アプリケーションの DAG 構造上のタスクを一筆書きのように順序付け，1 つ 1 つ要求を送信することでタスクを割り当てる手法を提案する．この重複割当を回避したタスクスケジューリングを実現するために ReadyList を導入し，段階的なタスクの順序付けと一筆書きでの要求送信パスの形成を行う．また，全ての要求が到達することを維持するため，未解決の Interest を一筆書きの要求経路に束ねる．

2.1 ReadyList

ReadyList は，全ての後続タスクが割当済みとなっている未割当のタスクの集合である．初めに，アプリケーションの DAG 構造を確認して ReadyList を作成する．そしてその中からタスクを 1 つ取り出し，割り当てるために要求を送信する．タスクが割り当てられた後は，再度 ReadyList から取り出し要求送信を行う．ここで，ReadyList が空になった場合には再度 DAG を確認して再作成する．この操作を繰り返しながら必ず 1 つずつ順番に割当を行うことで，DAG 構造上に一筆書きのようにタスク割当のパスを形成することができる．

2.2 Interest の集約

アプリケーションを実行可能な状態でスケジューリングするためには，全てのタスクがそれぞれの先行タスクから処理結果を受け取れるようにする必要がある．そのために，Interest に未解決の要求を束ねながら Interest を転送する．

3. 性能評価

本提案手法を評価するため，ICN を用いたワークフロー型のシミュレータである icn-sfcsim[2] で実験を行う．ここでは，ノードが 500 台ある環境に対してタスク数 30 のランダムなアプリケーションを投入する．そして，既存手法である multicast 的手法と提案手法による一筆書きでは，アプリケーションの応答時間にどのような変化があるかを検証する．ここで，アプリケーションの特性は CCR (Communication to Computation Ratio) によって特徴付けられ，その特徴によって応答時間も変化すると考えられる．CCR は，アプリケーションの通信と処理の比率を示すものである．そのため，評価においては投入するアプリケーションの CCR を 0.1 10.0 まで変化させながら応答時間を比較する．図 1 に，実験結果のグラフを示す．縦軸がアプリケーションの要求からデータの受取までの時間を示す TurnaroundTime となっており，横軸は CCR である．この結果から，提案手法では重複割当を防いだことにより，ネットワークの負荷が低減し TurnaroundTime も減少していることがわかる．一方で，既存手法と比較して実行にかかる時間が増加傾向にあるとわかる．これは，一筆書きでタスクを割当てたことによりタスク間の経路長が伸び，処理結果の受け渡しにかかる時間が増加したためと考えられる．そのため，提案手法は計算資源の有効活用という観点ではよいが，処理結果のやり取りについて改善の余地があると分かった．

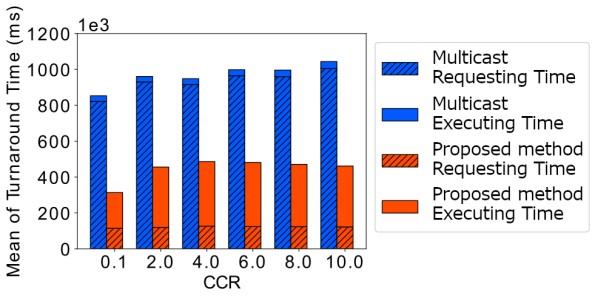

図 1: 各手法の TurnaroundTime とその内訳の比較

6. まとめ

本研究では，ICN を用いた SFC において DAG アプリケーションを実行した際に生じる，タスクの重複割当を解決する手法を提案した．今後はより詳細な性能評価を行い，本提案手法が具体的にどのような状況において有効であるかという点について明らかにする．

謝辞　本研究成果は JSPS 科研費 23K11086，23K21664，23K28078 の助成を受けたものである．ここに記して謝意を表す．

参考文献

[1] 金光永煥, 花田真樹, 金井謙治, 中里秀則, ワークフローにおける ICN を用いたファンクションチェイニング, 信学技報, vol.119, no. 461, IN2019-120, pp. 249-254, 2020 年 3 月.
[2] icn-sfcsim github repository, https://github.com/ncl-teu/ncl icn-sfcsim