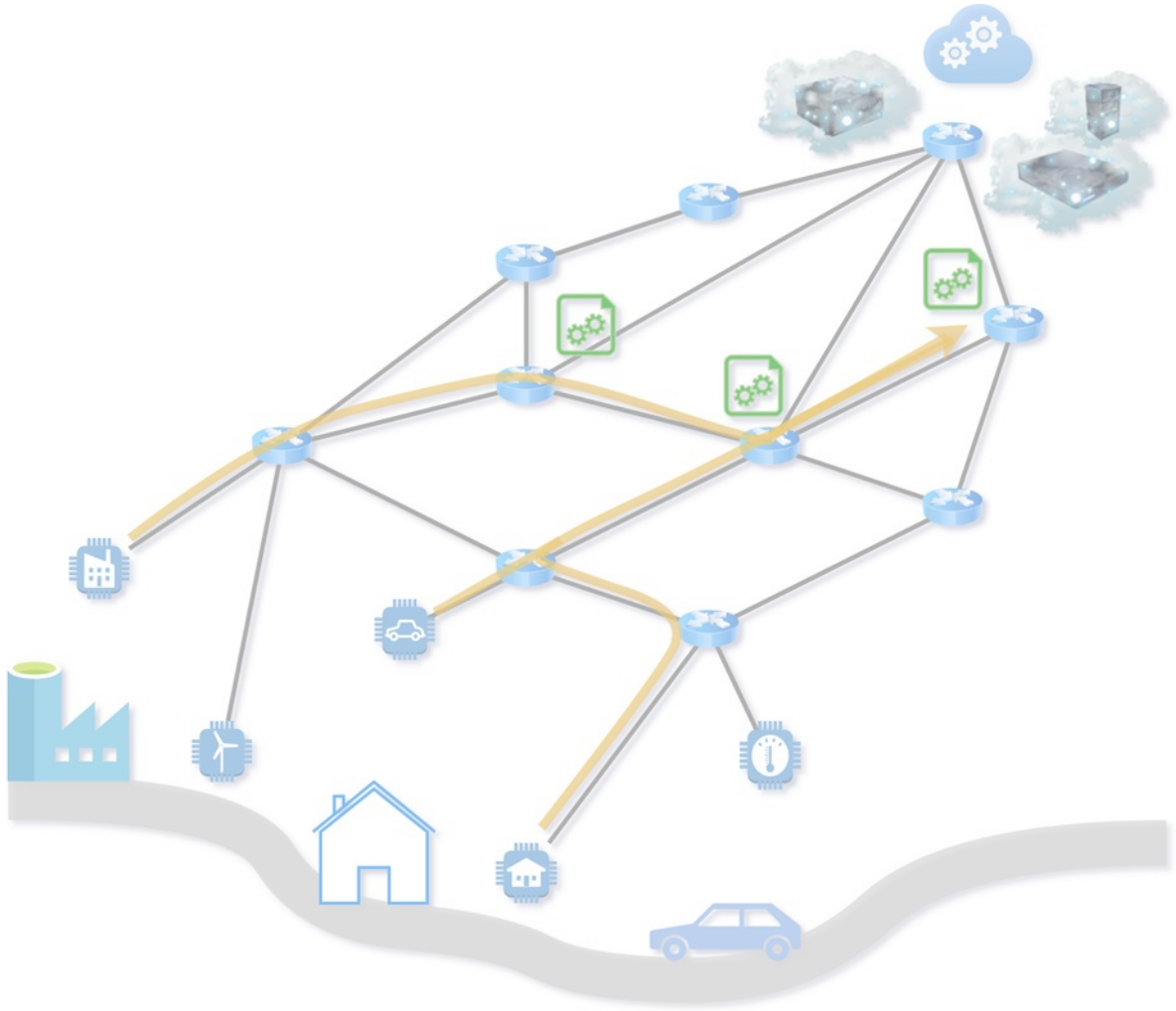
自律的ネットワーク内処理における タスク粒度を考慮した重複スケジューリング

新部 裕樹†，上山 憲昭††
† 立命館大学大学院 情報理工学研究科
†† 立命館大学 情報理工学部

目次

- 研究背景
- 既存研究
- 課題
- 研究目的
- 提案
- 評価
- まとめ



研究背景

- IoT デバイスの普及とビッグデータの台頭
  - IoT BigData (IoTBD)
  - 生データをそのまま利用することはできない
  - 「スマートデータ」への変換が必要
    ⇢ 情報の “集約・抽出・融合”等の処理

- IoTBD を利用するアプリケーションの需要増
  - スマートシティ，デジタルツイン / CPS ， IoV (Internet of Vehicles)

これらの需要に答えるために
⇢ 多種多様な連携処理を汎用的に実行する基盤が求められる

研究背景

- 多種多様な処理の実行
  - IoTBD (IoT Big Data)
    - これらのデータは遠隔地のクラウド上で処理される

IoT デバイス：ネットワークのエッジ

データの収集

⇩

クラウド：ネットワークのコア

データの処理

- 効率性に問題
- ネットワーク
  トラヒックの圧迫

⇢ **In-Network Computing の適用**

ネットワークのエッジからコアに向かってデータを転送させていく中で，処理も同時に行う

研究背景

- In-Network Computing
  - これまでの **In-Network Computing**： 中央集権型
    - マスターノードが一元的に管理
      - 各ノードの場所，タスクの配備状況の管理
      - タスク間の通信経路・タスクの実行先の決定
    - IoT デバイスは今後も増加傾向 [1]
      - 2030 年には 2020 年比 10 倍の市場規模に成長する可能性

    ⇢マスターノードが全てを管理・決定する方法では，スケールしない

  - これからの **In-Network Computing**： 自律型のものが必要
    - ネットワーク内デバイスが自律的にタスクを管理・実行する

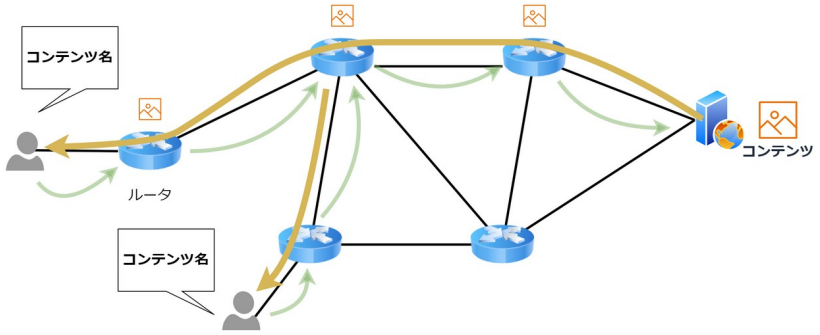
1 S. Al-Sarawi, M. Anbar, R. Abdullah and A. B. Al Hawari, "Internet of Things Market Analysis Forecasts, 2020 − 2030," 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), London, UK, 2020, pp. 449-453.

既存研究

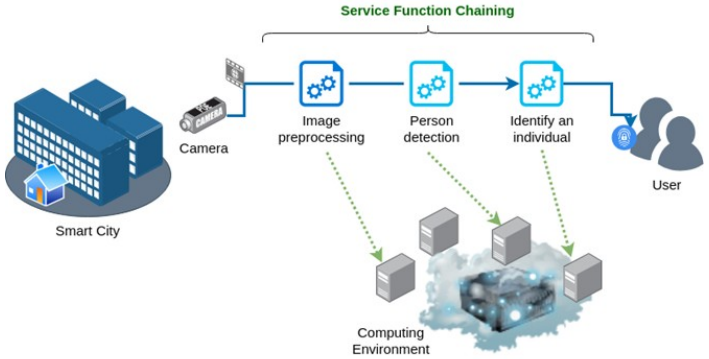
In-Network Computing の実現のために

- **情報指向ネットワーク (ICN)** + **Service Function Chaining (SFC)**

コンテンツ / ファンクションの位置に
とらわれない自律的なデータのやり取り



ネットワーク上のファンクションを
連携させ処理を実行する仕組み



= ICN-SFC [2]

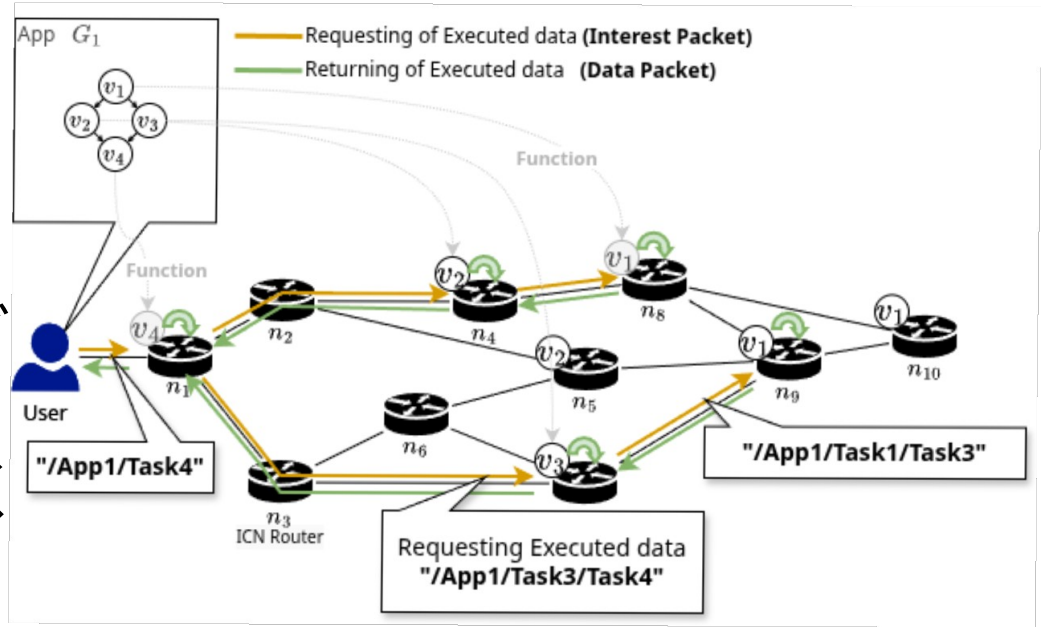
➡ **自律的な In-network computing**

- 多種多様な連携処理への要求に対して，効率よく柔軟に答えることができる
- データやデバイスの増加に対して，スケールすることで対処可能

2 金光永煥 , 花田真樹 , ”ICN ワークフローにおける自律的なタスク割当てアルゴリズムの性能解析” , 信学技報 , vol. 125, no. 13, CS2025-3, pp. 13-18, 2025 年 4 月 .

既存研究： ICN-SFC[2]

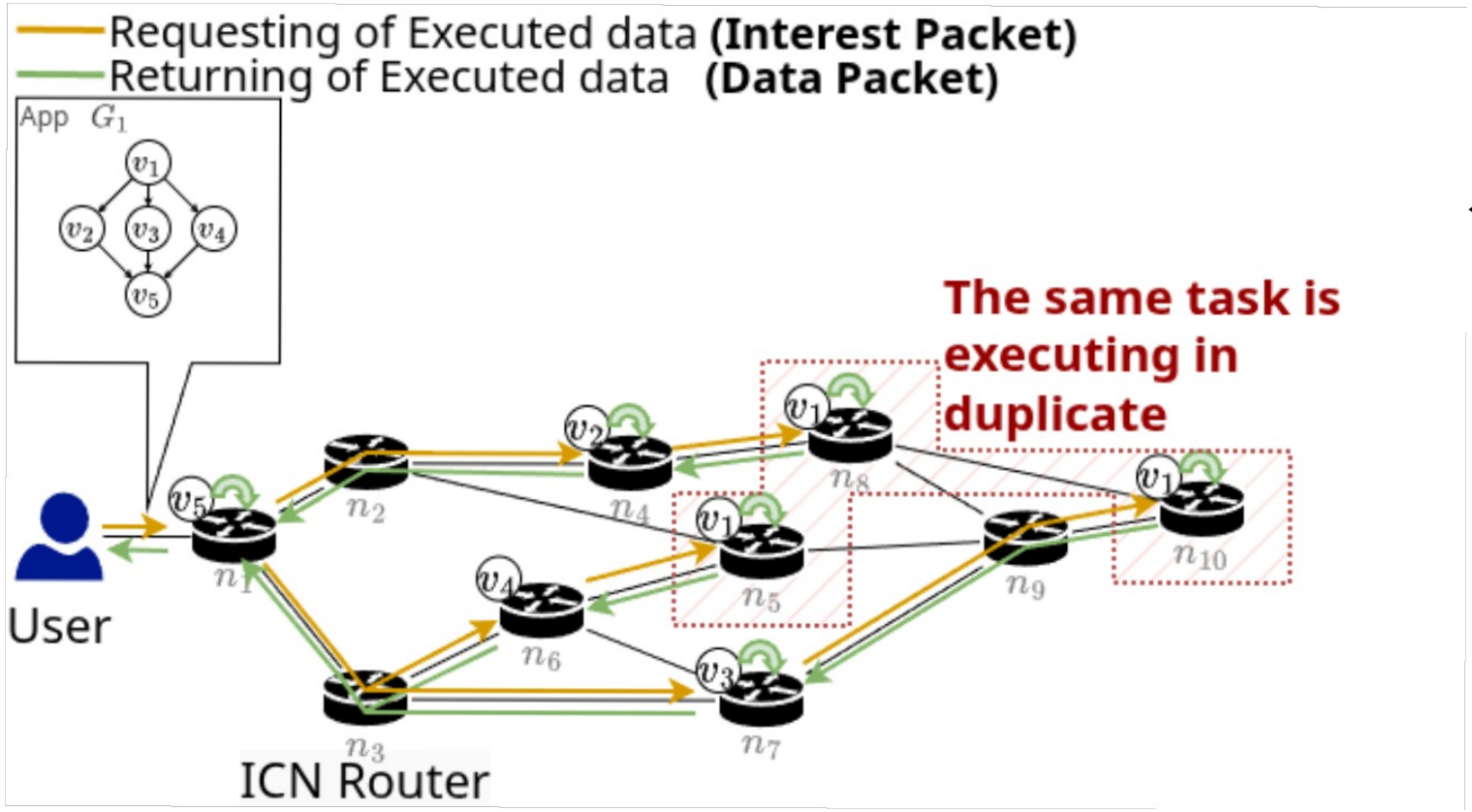
- 後続から先行へ順に処理要求 → 先頭から順に実行
  - 要求されたパスを逆順に辿って処理結果を届ける
  - 名前で処理結果を要求
    “/AppID/ 要求先タスク / 要求元タスク”
  - Longest Match ルーティング
    - 要求はタスクを保持する
      ノードへ向かって飛んでいく
    - 実行時間 (Makespan) が短縮できるノードが見つかれば
      タスクを保持していなくても実行場所とする (= 新規割当)



2 金光永煥 , 花田真樹 , ”ICN ワークフローにおける自律的なタスク割当てアルゴリズムの性能解析” , 信学技報 , vol. 125, no. 13, CS2025-3, pp. 13-18, 2025 年 4 月 .

課題

- 本研究：自律的な In-Network Computing（ICN-SFC)
  - DAG (有向非循環グラフ) 構造のアプリケーションを実行すると
    タスクの重複配備•重複実行が発生してしまう



⇩

タスクの冗長実行により計算資源が浪費

課題

- 本研究：自律的な In-Network Computing (ICN-SFC)
  - 単純にタスクの重複を解消すればよいわけではない
    - トレードオフの関係

      計算資源の消費量　↔　アプリケーションの実行性能

      - 利用可能な計算資源が限られる場合，計算資源利用効率を最大化したい
      - 追加の計算資源が利用可能な場合，限定的に重複実行し性能を向上させたい
  - 真の課題：トレードオフの関係に対してバランスを取れるか

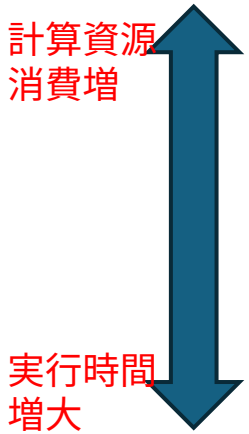
研究目的

- 目的：
  - 自律的な In-network computing において，
    DAG アプリケーション実行時の トレードオフの関係に対する
    有効な割当手法を提案する
    ⇛ 計算資源の消費量を抑えつつ，性能を向上させる

- 意義：
  - 計算資源を有効に活用し，省エネに貢献
  - 不必要な計算資源の消費を抑えつつ，性能向上し
    今後も増大する需要に答えることができる

提案

- DAG アプリケーション実行時の 計算資源の浪費を削減
  ＋応答時間の改善



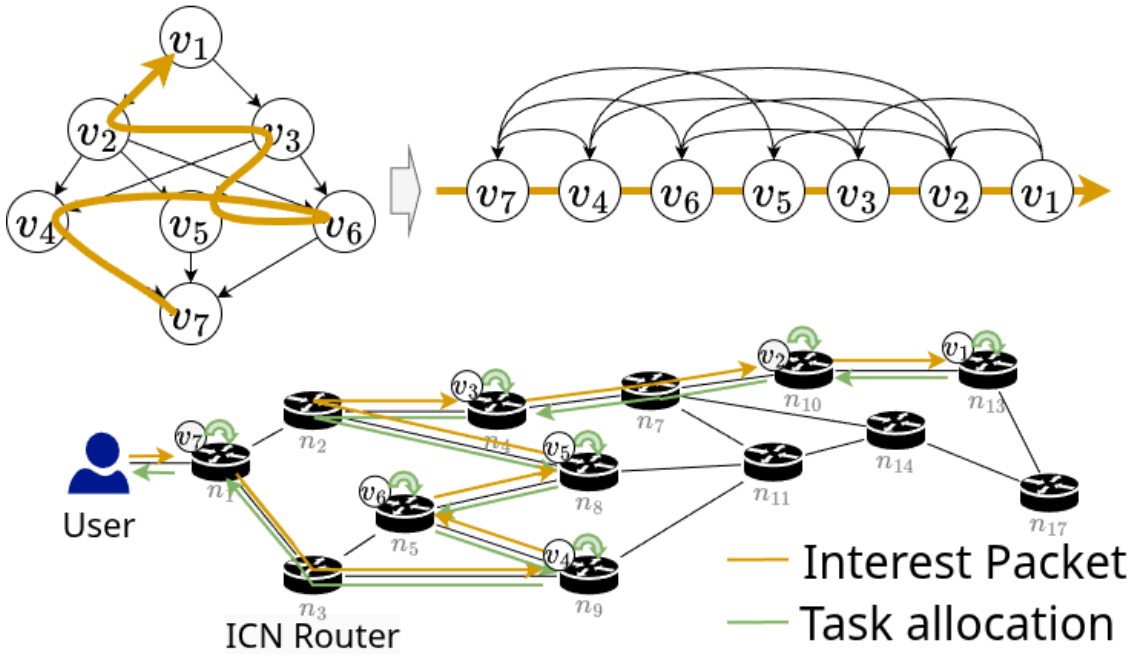
- **[従来手法]** DAG の形状に沿った割当 (重複が発生)
- **[提案手法]** 実行時間短縮のため 必要なタスクのみ重複割当を行う
- **[以前の提案]** DAG の重複しない割当を行う [3, 4]

3 新部 裕樹 , 上山 憲昭 : ICN を用いたネットワーク内処理における重複割当を回避したタスクスケジューリング , 信学技報 , vol. 124, no. 251, CCS2024-49, pp. 35-40, 2024 年 11 月 .
4 Y. Niibe and N. Kamiyama, “Task Scheduling with Duplication Avoidance for ICN-based Autonomous In-Network Computing,” 2025 IEEE 31st International Symposium on Local and Metropolitan Area Networks (LANMAN), Liile, France, 2025, pp. 1-2.

以前の提案：DAG の重複しない割当

- 計算資源利用効率向上のため重複割当を排除する制御
  - DAG 内のタスクをトポロジカルソートの操作によって並び替え，必ず１つずつ割り当てる
    - 既存手法：DAG のタスク割当を並列に行うことで重複が発生

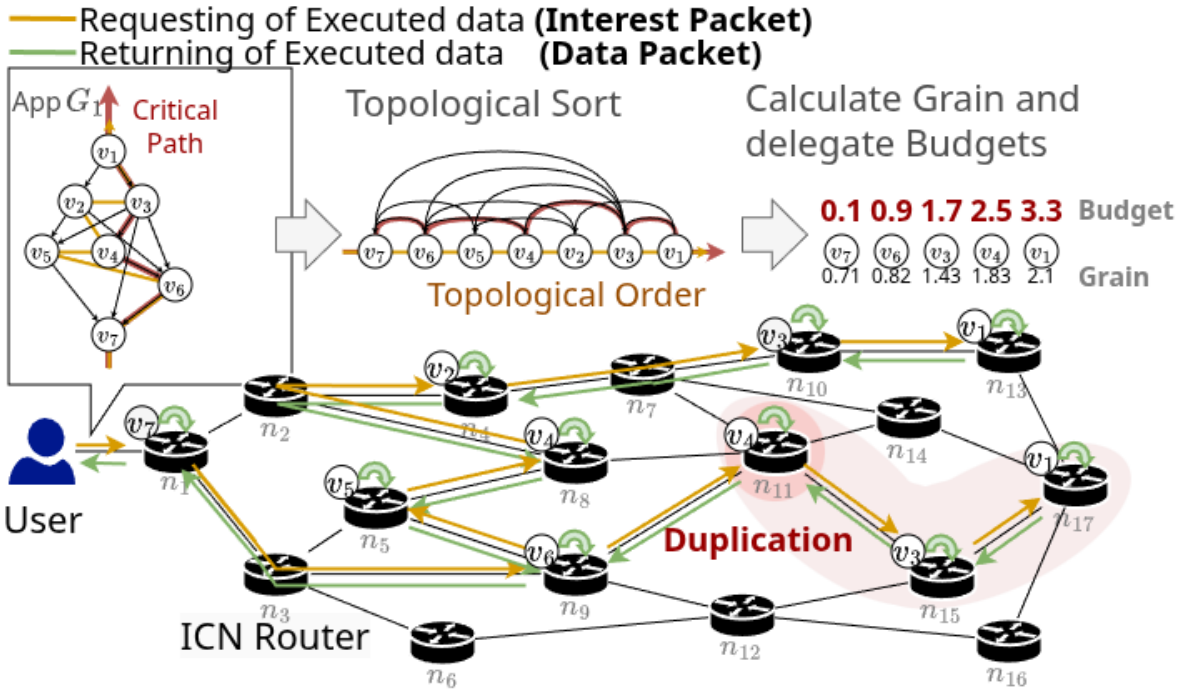
      ⇢タスクの割当操作に並列部がなくなることで，
      必ず重複せずに割当可能



提案手法：必要なタスクのみ重複割当

- 実行時間削減の観点から，必要なタスクを重複割当する

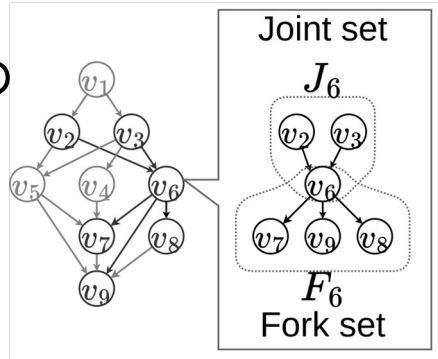
- 全体の流れ
  - **静的解析フェーズ**
    - ✓重複候補選出
    - ✓全体の最大重複数を決定
  - **動的判断フェーズ**
    - ✓重複候補を動的経路情報で重付け，再度並び替える
    - ✓実行時間削減の効果が高いものから重複割当



実行時間削減に必要なタスクのみを，重複数の爆発を抑えつつ重複

提案手法：必要なタスクのみ重複割当

- 実行時間削減の観点から，必要なタスクを重複割当する
  - **静的解析フェーズ** ＠一番初めに処理要求するノード（ユーザ）
    - DAG 上の Critical Path (CP) を特定 → 重複対象候補
      - CP の実行時間がアプリケーション全体の実行時間を決定するため
    - CP の各タスクに対し grain $g(v_x)$ を計算
      - 時間削減のしやすさで順序付けるため



$$g(J_x) = \frac{\min\limits_{k=1:m} D_p(v_x, \alpha_{END})}{\max\limits_{k=1:m} D_c(d_{k,x}, L_{END})}$$

$$g(F_x) = \frac{\min\limits_{k=1:m} D_p(v_x, \alpha_{END})}{\max\limits_{k=1:m} D_c(d_{x,k}, L_{END})}$$

$$g_{static}(v_x) = \min g(F_x), g(J_x)$$

    - DAG の理想の並列度 W / L を計算 → 全体の最大重複数（重複予算）を決定
      - Brent の定理の適用 [5,6]
    - 順序付けられた CP に対し，重複予算を線形的に割り振る

$$B_{total} = \frac{W}{L} \cdot |CP|$$

5 BRENT, R. P. 1974. The parallel evaluation of general arithmetic expressions. J. ACM 21, 2 (Apr.), 201-206.
6 Y. Niibe and N. Kamiyama, “Task Scheduling with Duplication Avoidance for ICN-based Autonomous In-Network Computing,” 2025 IEEE 31st International Symposium on Local and Metropolitan Area Networks (LANMAN), Liile, France, 2025, pp. 1-2.

提案手法：必要なタスクのみ重複割当

- 実行時間削減の観点から，必要なタスクを重複割当する

  - **動的判断フェーズ** ＠ 処理要求を受信したノード
    - 経路統計情報を用いて， CP 上の各タスクの $\mathrm{grain}_{g(v_x)}$ を重み付ける
      - 現時点での時間削減のしやすさで再度順序付けるため (予算の再分配)

      - 経路統計情報 ：In-band Network Telemetry のように，
        処理要求のパケットへ付与した
        ネットワークの統計情報

$$L_{avg} = \frac{\sum_{i=0}^{|P(p_i)|} L_{i,i+1}}{|P(p_i)| - 1}$$

$$g(J_x) = \frac{\min\limits_{k=1:m} D_p(v_x, \alpha_{avg})}{\max\limits_{k=1:m} D_c(d_{k,x}, L_{avg})}$$

$$g(F_x) = \frac{\min\limits_{k=1:m} D_p(v_x, \alpha_{avg})}{\max\limits_{k=1:m} D_c(d_{x,k}, L_{avg})}$$

$$g_{dynamic}(v_x) = \min g(F_x), g(J_x)$$

    - 重複予算が残っていれば，そのタスクを重複して要求する → 重複割当
      - 予算が残っていない → 重複することができない
      - 予算が残っている　→ タスクの重複要求を実行し，予算を 1 減らす

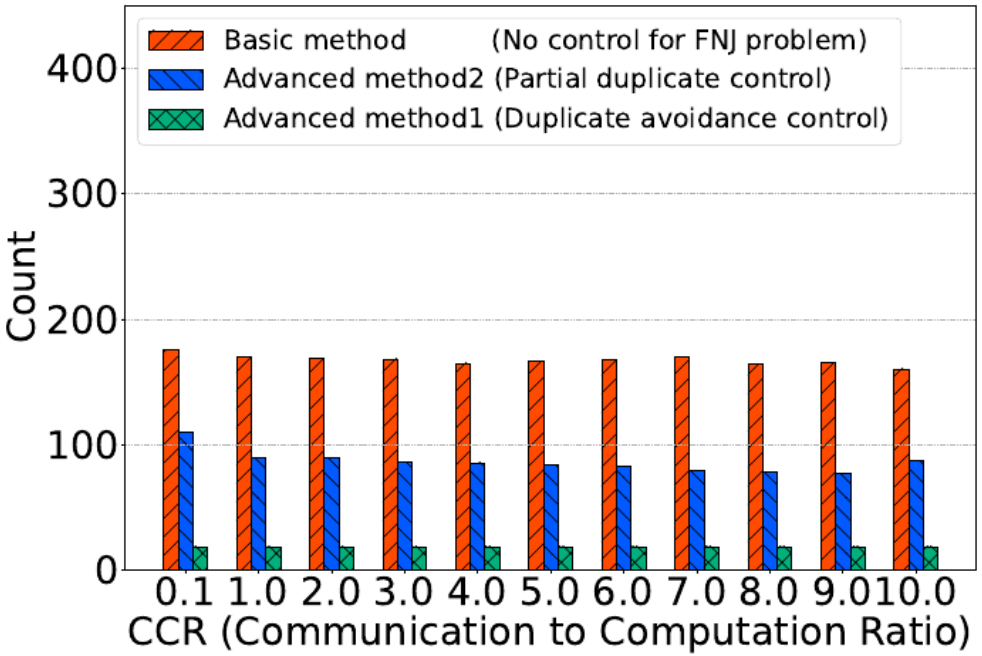
評価

- 評価シミュレーションを実施
  - o シミュレーション環境
    - ランダムネットワーク：
      - 構成　　：ユーザ 100, ネットワークノード 500
      - 処理能力： 2000 ～ 4000MIPS, 帯域幅： 100 ～ 1000Mbps
  - o 投入アプリケーション
    - ランダム DAG ：
      - タスク数： 20
      - 様々なアプリケーションを想定するため， CCR を変動させる

        CCR ： Communication to Computation Ratio
        アプリケーション全体で見て通信と処理のどちらに重きがあるか
        = アプリケーションを特徴付ける指標

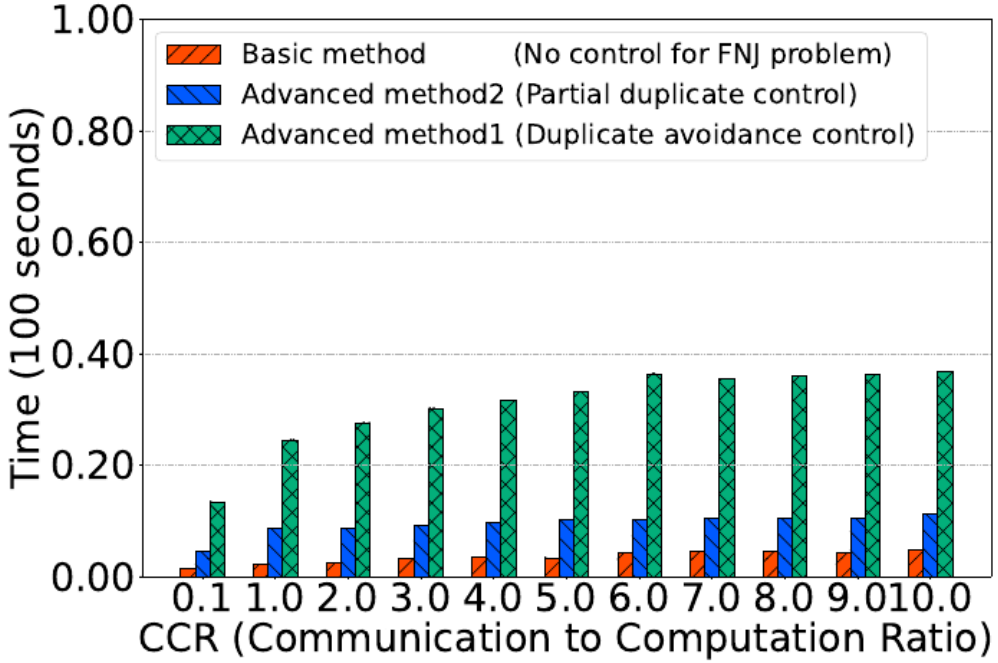
評価

- タスクの割当回数
  - o タスク数 (=20) 以上の割当分が重複に該当
    - 従来手法は最大で 180 回の重複割当
    - 以前の手法は重複割当なし (重複回避)
    - 提案手法は従来手法と以前の手法の中間程度の割当回数となっている



評価

- 実行時間
  - o 重複回数に応じて実行時間が増減
    - 重複回数大→純粋に実行時間が減少
      - 今回は比較的大規模なネットワークで実験
      - 計算資源が余っているため，重複実行すればするだけ得という状況に
    - 提案手法は以前の手法（重複回避）よりも実行時間が短い



＝必要最小限の重複実行で，アプリケーション全体の高速化に成功

まとめ

- 自律的な **In-Network Computing** : ICN-SFC + DAG アプリ
- ICN-SFC ではタスクの重複割当•実行が発生
- 計算資源の利用効率を改善するアルゴリズムを提案
  - [従来手法] DAG に沿って重複割当が生じるアルゴリズム
  - [以前の手法] 重複割当を防ぐアルゴリズム
  - [提案手法] 実行時間削減の観点から，必要な分だけ重複割当するアルゴリズム

- 今後： 提案手法の詳細な評価
  - アプリケーションの複雑性を変化させての性能評価
  - 実世界のアプリケーションへの性能評価
  - ネットワークの規模を変化させての性能評価

付録

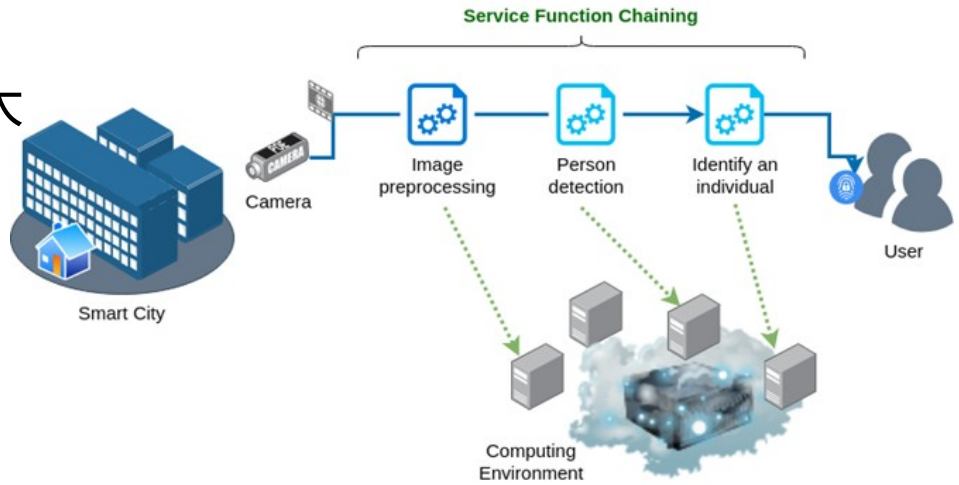
既存研究： Service Function Chaining

自律的な In-Network Computing の実現のために

ネットワーク仮想化における，機能の連携に用いられる技術

- 経由するべき処理 / 機能を，連携させ順番に経由させる
  データを転送させていく過程で
  加工•処理できる

- ネットワーク機能に限らず，
  様々なアプリケーションに適用範囲を拡大
  ✔️汎用的に様々なタスク間の連携を実現



既存研究：情報指向ネットワーク

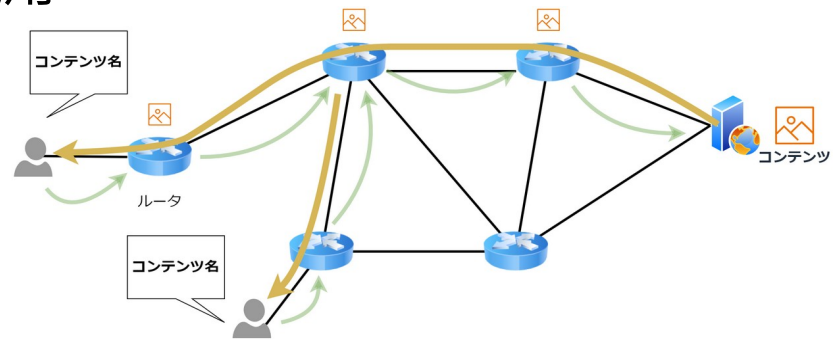
自律的な In-Network Computing の実現のために

効率的なコンテンツ配信のための手法

- コンテンツの場所 (IP アドレス等) ではなく，“コンテンツの名前”を指定して通信
  実際にコンテンツがどこにあるか知る必要がない (要求後に発見される)
- ネットワーク内キャッシュの利用
  ユーザの近い場所からコンテンツを取得•配信可能

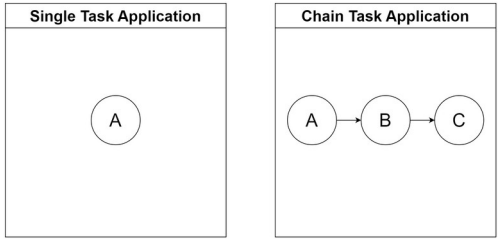
⇢ In-network computing に適用

- 名前を指定して取得するパラダイムを処理に適用
  ✔️ 中央サーバの必要なしにファンクションを名前で発見し，処理結果を取得
- 処理結果をキャッシュとして再利用
  ✔️ 同一の要求に対して計算資源を節約



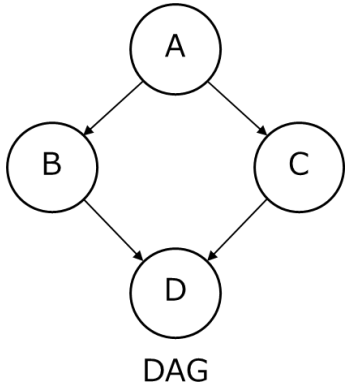
既存研究

- 自律的な In-Network Computing : **ICN-SFC**
  - いくつかのアーキテクチャが提案
    - 単一タスク構造のアプリケーション [7]
    - チェイン構造のアプリケーション [8]



  - 多種多様な処理を実行するためには，様々なアプリケーション構造に対応しなければならない
    - 昨今のアプリケーション構造の複雑化

⇢ **DAG (有向非巡回グラフ) を想定** [2]
アプリケーション構造は一般化すると DAG に



2 金光永煥 , 花田真樹 , ”ICN ワークフローにおける自律的なタスク割当てアルゴリズムの性能解析” , 信学技報 , vol. 125, no. 13, CS2025-3, pp. 13-18, 2025 年 4 月 .
7 C. Tschudin and M. Sifalakis, "Named functions and cached computations," 2014 IEEE 11th Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, USA, 2014, pp. 851-857.
8 L. Liu et al., "ICN-FC: An Information-Centric Networking based framework for efficient functional chaining," 2017 IEEE International Conference on Communications (ICC), Paris, France, 2017, pp. 1-7.

従来手法：タスク割当（重複制御なし）