Optical Micro Disaggregated Data Centers Capable of Executing Many Tasks Simultaneously

大阪大学 大学院情報科学研究科 村田研究室
博士後期課程 3 年
生駒 昭繁

2025/2/10 1

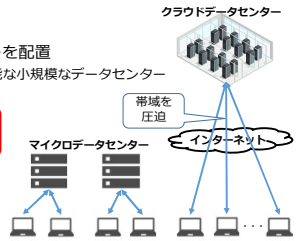
1

マイクロデータセンターによるエッジサービスの提供

- 近年、クラウドコンピューティングを通して多くのサービスが提供される
  - データセンターとの通信集中によって帯域が圧迫
  - 負荷の削減が求められる
- ユーザの近くにマイクロデータセンターを配置
  - マイクロデータセンター：エッジに配置可能な小規模なデータセンター
  - 通信負荷の分散が可能

ただし、小規模であるため、タスクの実行に利用できる資源 (CPU,GPU等) が少ない

マイクロデータセンター



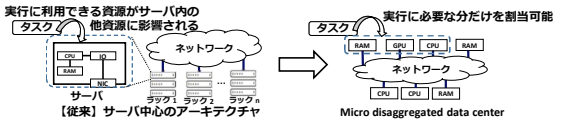
マイクロデータセンターとクラウドデータセンター

出典：https://www.zelladc.com/about-us/, last accessed on 2024-11-24. 2

2

Resource disaggregation

- サーバ内に集約されていた資源を分離させ、ネットワークによって接続して構成するアーキテクチャ
  - 各タスクの実行に必要な分だけを割当可能
  - ➢ 資源の利用効率を向上させることが可能 [1]
- Resource disaggregation を適用したマイクロデータセンターの構成によって、多くのタスクを同時実行できるエッジ基盤を構築
  - 以降、Micro disaggregated data center (μDDC) と呼ぶ



サーバ中心のデータセンターとμDDCの比較 [1]

[1] R. Lin, Y. Cheng, M. D. Andrade, L. Wosinska and J. Chen, "Disaggregated Data Centers: Challenges and Trade-offs," in IEEE Communications Magazine, vol. 58, no. 2, pp. 20-26, February 2020. 3

3

Beyond 5G超大容量無線通信を支える次世代エッジクラウドコンピューティング基盤の研究開発における μDDC

- 光通信技術をコアにしたエッジクラウドコンピューティング基盤の開発 [2]
  - マイクロデータセンターによってエッジコンピューティング基盤を構成

開発項目
1. 高速大容量データ転送を実現する革新的ハードウェア技術の研究開発
2. 多種多様なサービスに対応可能な高機能エッジクラウド情報処理基盤の研究開発
   1. マルチコアファイバを用いた極低遅延スイッチング技術の研究開発
   2. マルチコアファイバで連結したリソース分離型コンピューティング技術の研究開発
   3. マルチコアファイバを用いたネットワークスライシング技術の研究開発
   4. エッジクラウドコンピューティングを活用した実証実験の実施

本プロジェクトの一部として
光インターコネクトを用いた μDDC の構成を検討

[2] NICT 革新的情報通信技術研究開発委託研究
Beyond 5G 超大容量無線通信を支える次世代エッジクラウドコンピューティング基盤の研究開発 4

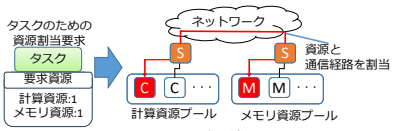
4

本論文で扱う μDDC のモデル

- 計算資源・メモリ資源が光ネットワークによって接続された μDDC を想定
  - 同種の複数資源は一定数ごとに 1つのスイッチに接続され、資源プールを構成
- 光回線交換スイッチ (OCS) とパケットスイッチ (PS) によりネットワークを構成
  - OCS: ポート間を直接接続することで光信号のまま通信できる経路 (光パス) を確立
  - PS: パケットの情報に基づいてデータを適切なポートにルーティングすることで通信
- 随時タスク資源割当要求が到着し、対応する資源と経路が割り当てられる

資源種別	役割
計算資源 (CPU,GPU)	タスクの実行
メモリ資源 (RAM)	タスクに利用するデータの保存

各資源種別の役割



μDDC における資源割当

5

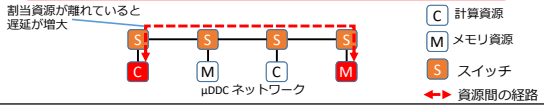
5

μDDC の課題と必要事項

- 資源間の通信遅延がサービスの性能に影響
  - ➢ 通信遅延が大きすぎる場合タスクの性能要件を満たすことができない
- 光インターコネクトによって資源間を接続した μDDC を構成
  - 高帯域・低遅延通信を安定して可能

ただし、遅延は経路設定にも依存

1. 低遅延な通信経路を確立できる資源割当
2. 性能要件を満たす資源割当が可能なネットワークトポロジ

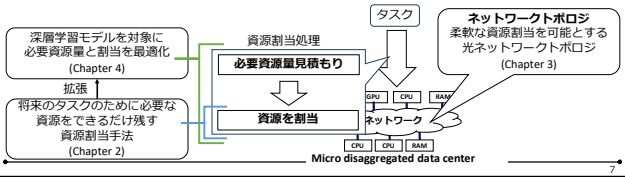


6

6

研究の目的とアプローチ

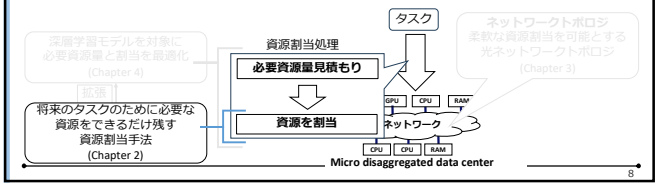
- 研究の目的
  - 多くのタスクを同時実行可能な光 μDDC の構成
- アプローチ
  - 各タスクが性能要件を満たすことができる資源割当手法を確立
  - その資源割当を実現できる光ネットワークトポロジ設計を確立



7

Chapter 2: Resource Allocation Considering Impact of Network on Performance in Micro Disaggregated Data Center

1. Akishige Ikoma, Yuichi Ohsita, Masayuki Murata, " Resource Allocation Considering Impact of Network on Performance in a Disaggregated Data Center, " IEEE Access, vol. 12, pp. 67600-67618, May 2024.
2. Akishige Ikoma, Yuichi Ohsita, Masayuki Murata, " Disaggregated Micro Data Center: Resource Allocation Considering Impact of Network on Performance, " in Proceedings of 2023 IEEE 20th Consumer Communications Networking Conference (CCNC), pp. 360-365, January 2023.
3. Akishige Ikoma and Yuichi Ohsita, Masayuki Murata, " Resource allocation method considering future resource requests in a disaggregated micro data center, " Technical Reports of IEICE (IN2021-36), vol. 121, no. 434, pp. 31-36, March 2022.



8

Chapter 2 の研究目的とアプローチ

- 研究の目的
  - μDDC において多くのタスクを性能要件を満たしながら実行するための資源割当手法の確立
- アプローチ

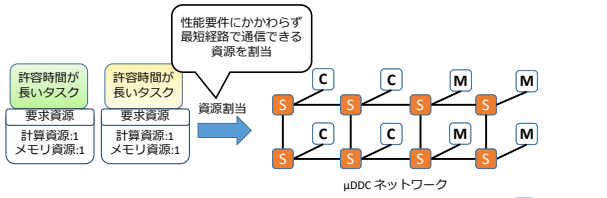
性能要件を満たしながら将来のタスクの実行のために必要な資源をできるだけ残す

- 資源割当に基づいた実行時間の見積もり
  - タスクの性能要件を満たす資源割当であるかどうかを検証可能に
- 性能要件を満たしながら将来のタスクに必要な資源の割当を最小化する最適化問題を定義
  - 将来のタスク実行に必要であるかという観点で資源やネットワークリンクにコストを付与

9

将来のタスクが必要とする資源をできるだけ残すために

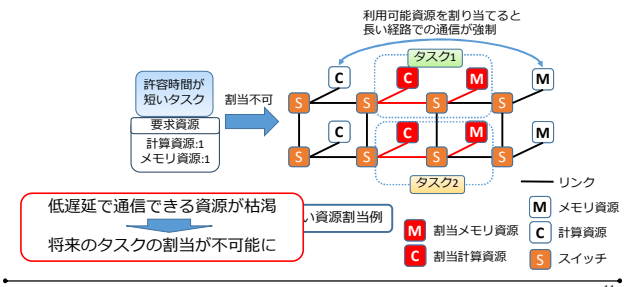
- タスクの性能要件を満たすかどうかを考慮した割当資源の選択が必要



10

将来のタスクが必要とする資源をできるだけ残すために

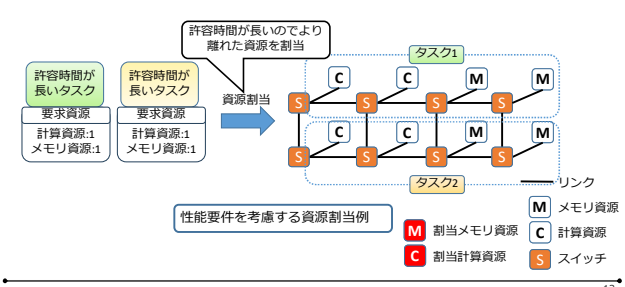
- タスクの性能要件を満たすかどうかを考慮した資源選択が必要



11

将来のタスクが必要とする資源をできるだけ残すために

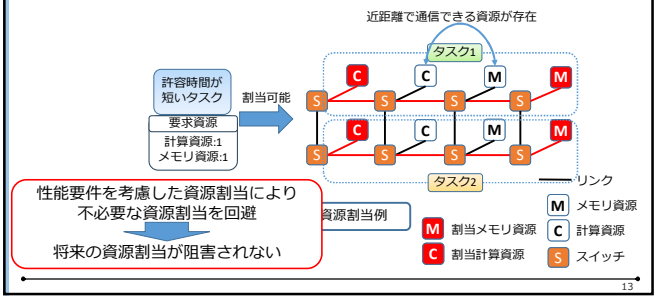
- タスクの性能要件を満たすかどうかを考慮した資源選択が必要



12

将来のタスクが必要とする資源をできるだけ残すために

- タスクの性能要件を満たすかどうかを考慮した資源選択が必要



13

13

資源割当手法 RA-CNP の提案

RA-CNP: 性能要件を満たしながら将来のタスクに必要な資源をできるだけ残すことを目的とする資源割当手法

提案内容

- 資源割当に基づくタスク実行時間の見積もり
- 将来のタスク割当に必要な資源であるかに基づいた資源割当コストの定義
- 性能要件を満たしながら資源割当コストを最小化する資源割当問題を定義し、最適な割当資源とその間の経路を決定

14

14

資源割当に基づくタスク実行時間の見積もり

- 計算資源内の処理時間とメモリからデータを読み出す遅延の和として見積もり

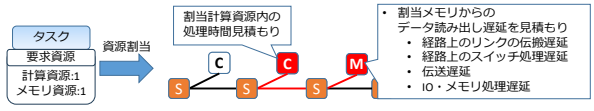
計算資源内処理時間 $\sum_{j\in N^s}\left(\delta^N_{c',j}\frac{\sigma^c_p}{F^c_j}\right)$ タスク実行に必要なクロック数を割当資源のクロック周波数で除算

データ読み込みにかかる通信遅延 $\left\{\left(\frac{\sum_{j\in N^s}\delta^N_{c',j}V_j}{B}\right)\sigma^n_p + T^l_{e^l_{c',m'},p}\right\} + \left[\sigma^I_p\right] + \left[\sum_{j\in N^s}(\delta^N_{c',j}\cdot T^I_j + \delta^N_{m',j}\cdot T^R_j)\right]$

通信発生回数　IO 処理＋メモリ資源内の処理遅延

伝送遅延　レイテンシ：(スイッチ処理遅延＋伝播遅延) × ホップ数

混雑時のデータ衝突回避のためのバッファリング遅延を考慮
- M/D/C 待ち行列モデルによって導出



15

15

将来のタスク割当に必要であるかに基づいた資源割当コスト

将来のタスク割当に必要となる可能性の高い資源やリンクの定義
- 高性能かつ多くの利用可能資源をもつプール内の資源
- 資源間の低遅延で通信できる経路となる可能性の高いリンク

これらに基づきコストを定義

計算資源コスト: $C^s_c \cdot K_c$　同じプール内の未割当資源数とFLOPS の積

メモリ資源コスト: M^s_m　同じプール内の未割当資源数

リンクコスト:
$$\begin{cases} \sum_{c\in N^c, m\in N^m} \frac{N^r_{c,m}(e)}{N^r_{c,m}} \frac{W^c_c \cdot W^m_m}{H_{c,m}} & e \notin E^{alc} \\ \epsilon & e \in E^{alc} \end{cases}$$

資源ペアの通信しやすさ（資源のコストの積をホップ数で除算）

全最短経路のうちそのリンクを通る経路の割合

割当済みのリンクのコストは極小値に

16

16

将来のタスクに必要な資源の割当を最小化する資源割当問題

- 性能要件を満たし、コストが最小となる資源割当を決定

資源割当問題の定義

制約条件:
$$\forall i \in N^v,\ \sum_{j\in N^s}\delta^N_{i,j} = 1$$
$$\forall x \in E^v, \forall k, s \in N^s,\ \sum_{y\in R_{k,s}}\delta^E_{x,y} = \delta^N_{n^{vs}_x,k}\cdot\delta^N_{n^{vd}_x,s}$$
$$\forall c \in N^c,\ C^s_c - \sum_{c'\in C^v}\delta^N_{c',c} \geq 0$$
$$\forall m \in N^m,\ M^s_m - \sum_{m'\in M^v}\delta^N_{m',m} \geq 0$$
$$\forall t \in S,\ T^e_t \leq T^a_t$$

割当に関する制約
- 1 要求資源に対して 1 資源が割り当てられる
- 通信する割当資源間に経路が割り当てられる
- 空き領域のある資源にのみ割り当てる

目的関数:
$$minimize\ \sum_{c\in N^c}\sum_{c'\in C^v}\delta^N_{c',c}(W^c_c) + \sum_{m\in N^m}\sum_{m'\in M^v}\delta^N_{m',m}(W^m_m) + \sum_{i,j\in N^s}\sum_{y\in R_{i,j}}\left[1_{\sum_{x\in E^v}\delta^E_{x,y}>0}\left(\sum_{e\in y}W^e_e\right)\right]$$

本研究では、Ant Colony Optimization (ACO) を利用して解を導出

17

17

将来のタスクに必要な資源の割当を最小化する資源割当問題

- 性能要件を満たし、コストが最小となる資源割当を決定

資源割当問題の定義

制約条件:
$$\forall i \in N^v,\ \sum_{j\in N^s}\delta^N_{i,j} = 1$$
$$\forall x \in E^v, \forall k, s \in N^s,\ \sum_{y\in R_{k,s}}\delta^E_{x,y} = \delta^N_{n^{vs}_x,k}\cdot\delta^N_{n^{vd}_x,s}$$
$$\forall c \in N^c,\ C^s_c - \sum_{c'\in C^v}\delta^N_{c',c} \geq 0$$
$$\forall m \in N^m,\ M^s_m - \sum_{m'\in M^v}\delta^N_{m',m} \geq 0$$
$$\forall t \in S,\ T^e_t \leq T^a_t$$

見積もったタスクの実行時間が許容時間以下

目的関数:
$$minimize\ \sum_{c\in N^c}\sum_{c'\in C^v}\delta^N_{c',c}(W^c_c) + \sum_{m\in N^m}\sum_{m'\in M^v}\delta^N_{m',m}(W^m_m) + \sum_{i,j\in N^s}\sum_{y\in R_{i,j}}\left[1_{\sum_{x\in E^v}\delta^E_{x,y}>0}\left(\sum_{e\in y}W^e_e\right)\right]$$

本研究では、Ant Colony Optimization (ACO) を利用して解を導出

18

18

将来のタスクに必要な資源の割当を最小化する資源割当問題

- 性能要件を満たし、コストが最小となる資源割当を決定

資源割当問題の定義

制約条件:

$$\forall i \in N^v,\ \sum_{j \in N^s} \delta^N_{i,j} = 1$$

$$\forall x \in E^v, \forall k, s \in N^s,\ \sum_{y \in R_{k,s}} \delta^E_{x,y} = \delta^N_{n^{vs}_x,k} \cdot \delta^N_{n^{vd}_z,s}$$

$$\forall c \in N^c,\ C^a_c - \sum_{c' \in C^v} \delta^N_{c',c} \geq 0$$

$$\forall m \in N^m,\ M^s_m - \sum_{m' \in M^v} \delta^N_{m',m} \geq 0$$

$$\forall t \in S,\ T^e_t \leq T^a_t$$

割当資源とリンクの総コストが最小となるように割当

目的関数:

$$minimize\ \sum_{c \in N^c} \sum_{c' \in C^v} \delta^N_{c',c}(W^c_c) + \sum_{m \in N^m} \sum_{m' \in M^v} \delta^N_{m',m}(W^m_m) + \sum_{i,j \in N^s} \sum_{y \in R_{i,j}} \left[1_{\sum_{x \in E^v} \delta^E_{x,y} > 0} \left(\sum_{e \in y} W^e_e\right)\right]$$

本研究では、Ant Colony Optimization (ACO) を利用して解を導出

19

19

評価内容

- 小規模環境における定義した資源割当問題の最適性の検証
  - 比較対象：最適解、ACO で導出した解、比較手法 (右表参照) の解
  - ➢最適解において割当要求数を最大化させることに成功
  - ➢ACO によって同様の数の割当が可能であることも確認

比較手法	リンク割当方針
NP (従来手法) (Network Performance)	資源間の通信遅延最小化
SP (Shortest Path)	資源間を最短経路で接続

本評価の比較手法 (リンクの割当方針が RA-CNP と異なる)

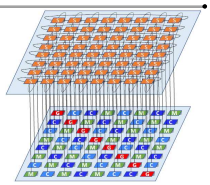
- より大きい環境における従来手法との比較評価
  - 性能要件を満たす割当に成功した数を比較
  - 比較対象：RA-CNP、比較手法 (右表参照)
  - ➢利用スイッチの異なる 2 ネットワークで有効性を確認
    - OCS によるネットワーク と PS によるネットワーク
  - 以降で PS によるネットワークにおける結果を説明
- RA-CNP の計算時間評価
  - RA-CNP が資源割当として現実的な時間内で処理可能であることを示すため
  - ➢10 秒以内の資源割当に成功し、実行前の資源割当として許容可能であることを確認

20

20

評価環境

- ネットワーク (2 種のネットワークで評価)
  - 6×6 2Dトーラス：計算資源数：294, メモリ資源数：288
  - 8×8 2Dトーラス：計算資源数：520, メモリ資源数：576
- 資源割当要求
  - 300 分間資源割当要求が分単位でランダムに生成
    - 各要求に対する割当資源は 90 分で解放
    - 性能要件の異なる 4 種の要求を生成
  - 4 ケースで評価
    - Base: 比較のための基準ケース
    - Many requested resources: 要求される資源数が多いケース
    - Short acceptable time: 要求の許容時間が短いケース
    - Many resource requests: 要求の生成数が多いケース
- 評価指標：棄却された資源割当要求数
  - 性能要件を満たすことができない場合、棄却



評価に利用するネットワーク (8×8 2Dトーラス)

M メモリ資源プール (24 資源)
C 低性能 CPU プール (16 資源)
C 高性能 CPU プール (16 資源)
G GPU　S スイッチ

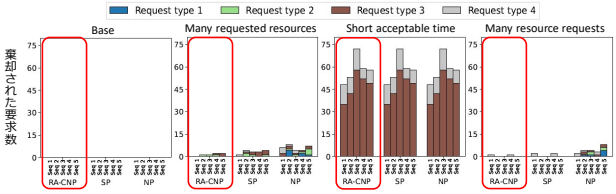
21

21

RA-CNP の評価

- 全環境で RA-CNP は比較手法と同等以下の棄却数を実現
  - 任意の環境において効率的な資源割当が可能

➢将来のタスクに必要な資源の割当をできるだけ避けることで多くのタスクを割当可能



各手法と棄却数の関係 (8 × 8 2Dトーラスネットワークの結果を抜粋)

22

22

Chapter 2 のまとめ

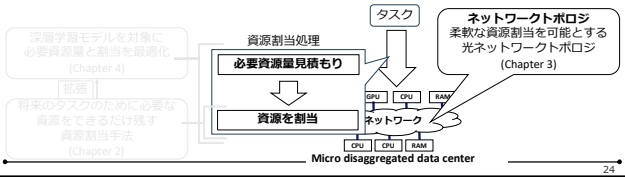
- μDDC において多くのタスクを性能要件を満たしながら実行するための資源割り当て手法 RA-CNP を提案
  - 性能要件を満たしながら将来のタスクの実行のために必要な資源をできるだけ残す
  - ネットワークがタスクの性能に与える影響をモデル化
  - 性能要件を満たしながら将来のタスクに必要な資源の割当を最小化する最適化問題を定義
- RA-CNPにより、多くのタスクを実行できるかを評価
  - 任意の環境で従来の手法と比較し同等以上のタスクの割当に成功
  - 性能要件を満たしながら多くのタスクを実行できる資源利用を実現

23

23

Chapter 3: Optical Network Topology Design to Execute Many Tasks Simultaneously in Micro Disaggregated Data Center

1. Akishige Ikoma, Yuichi Ohsita, and Masayuki Murata, “Optical network topology design to execute many tasks simultaneously in a disaggregated data center,” IEEE/OSA Journal of Optical Communications and Networking, Vol. 16, No.7, pp. 764–780, July 2024



24

24

Chapter 3 の研究目的とアプローチ

- 背景
  - 資源割当手法はネットワークトポロジに依存
    - 資源間の位置関係がトポロジ上で離れていれば、低遅延な通信経路の確立は不可能

  ➢各タスクに対して性能要件を満たす資源割当を可能とするネットワークトポロジが必要

- 研究の目的
  - 光回線スイッチ (OCS) とパケットスイッチ (PS) を併用した柔軟な資源割当が可能な物理的なネットワークトポロジの確立

- アプローチ
  - μDDC ネットワークトポロジにおける多数のタスクの同時実行能力を評価する指標を定義
    - 各資源に対する、性能要件を満たす通信経路を確立できる資源の数の割合
  - 評価指標を最適化するネットワークトポロジ設計問題を定義し、トポロジを生成

25

25

光回線スイッチとパケットスイッチによるネットワーク構成

- OCS と PS の併用により低遅延かつ柔軟な経路設定が可能なネットワークを構成

OCS
- ポート間の直接接続によって光パス を確立して通信
- ➢ 低遅延だが経路設定の柔軟性低下

PS
- パケット情報に基づいたポートへの柔軟なルーティングが可能
- ➢ OCS より遅延は大きいが、柔軟な経路設定が可能

PS 間に光パスを確立することで、1 つの光パスで複数資源間の通信を収容可能

26

26

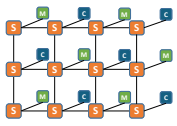
μDDC ネットワークトポロジ評価指標 CSTE

CSTE: 各資源に対する、性能要件を満たす通信経路を確立できる資源の数の割合

CSTE が高いほど

タスクの実行が可能な資源ペアが多いため、柔軟な資源割当が可能

CSTE の導出フロー



(1) 各資源が一定ホップ (HR) 内の資源と経路を確立
例：HR = 5 であれば、 5 つのメモリ資源のいずれかと通信

(2) 全資源の通信時にノード間を通過する光パスの数を見積もり
➢ ノード間のリンク数 ≧ 見積もった光パスの数であれば経路確立可能

(3) 経路確立できる最大 HR 内かつ、性能要件を満たすために許容できる数のスイッチを通過する経路で通信可能な資源数の割合を導出
例：最大HR = 5, 性能要件を満たすホップ数 = 3 である場合、
CSTE = 全資源数に対する、各資源から3ホップ内の資源数の割合

27

27

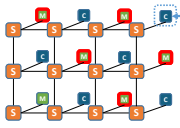
μDDC ネットワークトポロジ評価指標 CSTE

CSTE: 各資源に対する、性能要件を満たす通信経路を確立できる資源の数の割合

CSTE が高いほど

タスクの実行が可能な資源ペアが多いため、柔軟な資源割当が可能

CSTE の導出フロー



(1) 各資源が一定ホップ (HR) 内の資源と経路を確立
例：HR = 5 であれば、 5 つのメモリ資源のいずれかと通信

(2) 全資源の通信時にノード間を通過する光パスの数を見積もり
➢ ノード間のリンク数 ≧ 見積もった光パスの数であれば経路確立可能

(3) 経路確立できる最大 HR 内かつ、性能要件を満たすために許容できる数のスイッチを通過する経路で通信可能な資源数の割合を導出
例：最大HR = 5, 性能要件を満たすホップ数 = 3 である場合、
CSTE = 全資源数に対する、各資源から3ホップ内の資源数の割合

28

28

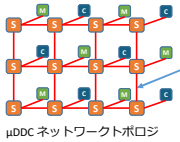
μDDC ネットワークトポロジ評価指標 CSTE

CSTE: 各資源に対する、性能要件を満たす通信経路を確立できる資源の数の割合

CSTE が高いほど

タスクの実行が可能な資源ペアが多いため、柔軟な資源割当が可能

CSTE の導出フロー



(1) 各資源が一定ホップ (HR) 内の資源と経路を確立
例：HR = 5 であれば、 5 つのメモリ資源のいずれかと通信

(2) 全資源の通信時にノード間を通過する光パスの数を見積もり
➢ **ノード間のリンク数 ≧ 見積もった光パスの数**であれば経路確立可能

(3) 経路確立できる最大 HR 内かつ、性能要件を満たすために許容できる数のスイッチを通過する経路で通信可能な資源数の割合を導出
例：最大HR = 5, 性能要件を満たすホップ数 = 3 である場合、
CSTE = 全資源数に対する、各資源から3ホップ内の資源数の割合

29

29

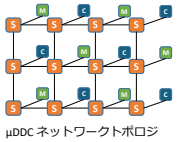
μDDC ネットワークトポロジ評価指標 CSTE

CSTE: 各資源に対する、性能要件を満たす通信経路を確立できる資源の数の割合

CSTE が高いほど

タスクの実行が可能な資源ペアが多いため、柔軟な資源割当が可能

CSTE の導出フロー



(1) 各資源が一定ホップ (HR) 内の資源と経路を確立
例：HR = 5 であれば、 5 つのメモリ資源のいずれかと通信

(2) 全資源の通信時にノード間を通過する光パスの数を見積もり
➢ ノード間のリンク数 ≧ 見積もった光パスの数であれば経路確立可能

(3) 経路確立できる最大 HR 内かつ、性能要件を満たすために許容できる数のスイッチを通過する経路で通信可能な資源数の割合を導出
例：最大HR = 5, 性能要件を満たすホップ数 = 3 である場合、
CSTE = 全資源数に対する、各資源から3ホップ内の資源数の割合

30

30

CSTE を最大化するネットワークトポロジ設計問題

- CSTE が最大となるような資源-スイッチ間、スイッチ間の接続と
  それらを接続する光ファイバの数を決定

ネットワークトポロジ設計問題の定義

制約条件:

$\forall s \in S^c \quad \sum_{n \in N} x_{s,n} \leq \eta_s^c$ — OCS のポート数を超える光ファイバケーブルを接続できない

$\forall s \in S^p \quad \sum_{n \in N} x_{s,n} \leq \eta_s^p$ — PS のポート数を超える光ファイバケーブルを接続できない

目的関数:

$maximize \quad CSTE(x,T)$ — CSTE が最大となるネットワークトポロジを出力

本章では遺伝的アルゴリズムを用いて上記問題の解を導出

31

31

評価内容

- CSTE と同時実行可能タスク数の関係評価 (RA-CNP を利用)
  - CSTE に基づいてネットワークトポロジを生成し割当に成功したタスク数を計測
  - ➢以降で結果を説明
- サーバ中心のマイクロデータセンターとの比較評価
  - μDDC と resource disaggregation を行わないデータセンターを比較
  - ➢適切な性能のネットワークを構築することで多くのタスクを実行できることを確認
- OCS と PSの併用の効果に関する評価
  - OCS と PS の最適な併用方法について CSTE をもとに議論
  - ➢OCS をベースとして、PS を必要箇所に接続したネットワークが望ましい
- 利用光ファイバケーブル数の評価
  - トポロジ構築コストの観点から利用光ファイバ数を削減する方法を議論
  - ➢マルチコアファイバによる同一ノード間の光ファイバ集約によって対処可能
    - マルチコアファイバ：ケーブル内に別データを送信可能な複数のコアを搭載した光ファイバ

32

32

評価環境

- 生成トポロジ
  - 資源プール数：16
  - 利用 OCS と PS
    - 資源プール構成のための PS
    - プール間接続に利用する OCS (16 or 24 個)
    - プール間接続に利用する PS (必要時に 1 つ追加)
  - 資源プール内の資源数を 2 通りに変更
- 比較トポロジ
  - OCS 16 個利用：2D トーラス、3Dトーラス
  - OCS 24 個利用：FatTree
- 資源割当要求
  - 2 ケースにおいて割当可能な限界まで要求を生成
    - 全要求の割当成功時、計算資源利用率が 100%
    - Case1: 許容時間の長いタスクの要求が頻発
    - Case2: 許容時間の短いタスクの要求が頻発
- 評価指標：
  - 生成された全要求数に対する棄却数の割合

接続資源数のパラメータ

接続資源パターン	1	2
プールあたりの計算資源数	576	480
計算資源数の合計	8064	6720
プールあたりのメモリ資源数	4608	3840
メモリ資源数の合計	9216	7680

生成する資源割当要求

資源割当要求種別	1	2	3
許容時間	400 ms	300 ms	200 ms
要求割合 (Case 1/2)	0.6/0.2	0.2/0.2	0.2/0.6
必要計算資源数	5 または 7 (ランダム)		
必要メモリ資源数	5 または 7 (ランダム)		

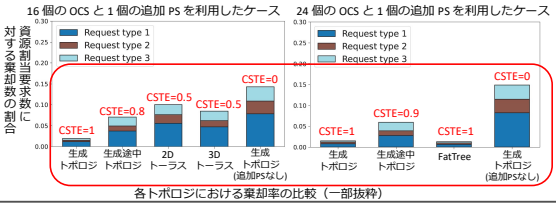
33

33

評価結果

- CSTE が高いほど多くの要求を割り当てることができることを確認
  - 最大で50 %以上タスクの棄却率を削減

➢性能要件を満たす通信が可能な資源ペアが多いトポロジ設計により
多くのタスクを実行可能



各トポロジにおける棄却率の比較（一部抜粋）

34

34

Chapter 3 のまとめ

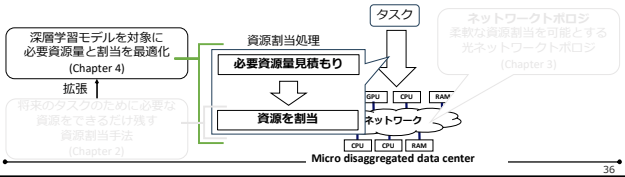
- OCS と PS の併用による柔軟な資源割当が可能な μDDC ネットワークを構成
  - 性能要件を満たす通信が可能な資源ペア数に基づいて構成
  - ➢柔軟な資源割当が可能
- 性能要件を満たす通信が可能な資源ペアを多く持つ
  トポロジを生成し、その設計の有効性を評価
  - ➢性能要件を満たす通信が可能な資源ペアが多いほど、多くのタスクを同時実行できる

35

35

Chapter 4: Resource Aware Deep Learning Model Partitioning and Allocation to Execute Many Deep Learning Tasks

- Akishige Ikoma, Yuichi Ohsita and Masayuki Murata, “Resource aware deep learning model partitioning and allocation for inference task in clusters with heterogeneous graphics processing units,” submitted for publication, October. 2024.



36

36

Chapter 4 の研究目的とアプローチ

- 背景
  - Chapter 2, 3 章において、タスク実行に必要な資源はあらかじめ固定
    - ➢ 不必要に多くの資源を割り当ててしまう可能性
  - 将来のタスクに必要な資源を残すためには割当資源数の考慮も必要
- 研究の目的
  - 割当資源数の最適化によって多くのタスクを同時実行できる資源割当を実現
- アプローチ
  - パイプライン並列化による深層学習（DL）モデルの実行を対象に割当資源数を最適化
    - DL モデルの分割の考慮により割当資源数を最適化
  - モデル分割と資源割当がタスクの実行性能に与える影響の定式化
    - RA-CNP で定義したネットワークが性能に及ぼす影響に対して拡張
  - DL モデルの分割と資源割当を包括的に考慮した最適化問題を定義
    - RA-CNP で定義した資源割当問題を拡張

37

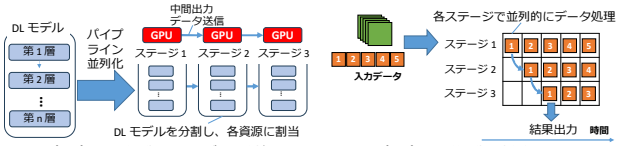
37

パイプライン並列化による DL タスクの実行

- DL モデルを複数のステージに分割し、各ステージを順番に処理
  - ステージ：パイプライン並列化における DL モデルの処理単位
  - 複数ステージの並列実行によりスループット向上
  - モデルの分割により 1 資源の持つメモリでは賄えない大きさのモデルも実行可能

ステージの数と割当資源の数は対応

モデルの分割数を最適化することにより割当資源数を制御



パイプライン並列化によるモデルの分割　パイプライン並列化の実行例

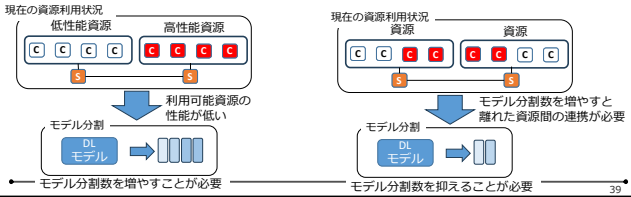
38

38

モデル分割の最適化における考慮事項

現在の資源利用状況によって、
性能要件を満たすことのできるモデル分割は異なる

モデル分割や割当資源の組み合わせにおける
タスクの性能の見積もりが必要



モデル分割数を増やすことが必要　モデル分割数を抑えることが必要

39

39

モデル分割と資源割当を最適化する手法 RAMPA の提案

RAMPA: DL モデルの分割と資源割当の双方を最適化することで
将来のタスクが必要とする資源の割当を最小化する手法

提案内容

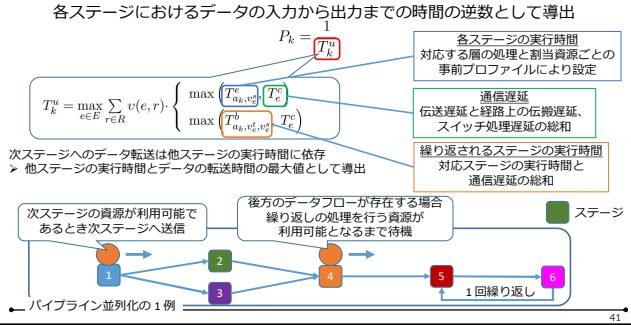
- モデル分割と割当資源の組み合わせに基づく
  スループット・実行時間の見積もり
- 将来のタスクに対する重要度に基づいたコストを定義
  - RA-CNP と同様に、高性能な資源とその間の経路のコストを高く設定
- 性能要件を満たしながら割当資源のコストを最小化できる
  モデル分割と資源割当の組み合わせを決定する最適化問題を定義

40

40

スループットの見積もり

各ステージにおけるデータの入力から出力までの時間の逆数として導出



パイプライン並列化の 1 例

41

41

実行時間の見積もり

各データフローにおいて通過するステージのデータの入力から出力までの時間
と通信遅延の総和の最大値として導出

$$T_k^r = \max_{v \in P_{v_k^t, v_k^s}} \left\{ T_k^u + \sum_{e \in y} \sum_{r \in R} v(e,r) \left(T_k^u + T^c_{a_k,e} \right) \right\}$$

各フローの最大値　各ステージにおける
データの入力から出力までの時間　ステージ間の
通信遅延

パイプライン並列化の 1 例

ステージ

42

42

モデル分割・資源割当問題の定義

- 性能要件を満たしながら将来のタスクに必要な資源の利用を最小化するモデル分割・資源割当の組み合わせを決定

モデル分割・資源割当問題の定義

制約条件:

$$\forall v \in V_{a_k},\ \sum_{g \in G} \chi(v,g) = 1$$

$$\forall e \in E_{a_k},\ \sum_{g^1,g^2 \in G} \sum_{r \in R_{g^1,g^2}} v(e,r) = \chi(v_e^s, g^1) \cdot \chi(v_e^t, g^2)$$

$$\forall k \in K,\ \gamma_k \le P_k$$

$$\forall k \in K,\ \delta_k \ge T_k^e$$

$$\forall k \in K, \forall g \in G,\ \sum_{v \in V_{a_k}} \chi(v,g) w_v + \sum_{e \in E_{a_k}} \overline{\chi}(v_e^s, g)\chi(v_e^t, g) d_e \le M_g$$

割当に関する制約
・DL モデルの層 1 つに対して 1 つの資源を割当
・データの転送が行われる実行資源間に経路を割当

目的関数:

$$minimize \quad \sum_{g \in G} 1_{\sum_{v \in V_{a_k}} \chi(v,g) > 0} C_g^g + \sum_{r \in R} \sum_{e \in E_{a_k}} v(e,r) \sum_{l \in r} C_l^l$$

RA-CNP と同様にAnt Colony Optimization によって解を導出

43

モデル分割・資源割当問題の定義

- 性能要件を満たしながら将来のタスクに必要な資源の利用を最小化するモデル分割・資源割当の組み合わせを決定

制約条件:

$$\forall v \in V_{a_k},\ \sum_{g \in G} \chi(v,g) = 1$$

$$\forall e \in E_{a_k},\ \sum_{g^1,g^2 \in G} \sum_{r \in R_{g^1,g^2}} v(e,r) = \chi(v_e^s, g^1) \cdot \chi(v_e^t, g^2)$$

$$\forall k \in K,\ \gamma_k \le P_k$$

$$\forall k \in K,\ \delta_k \ge T_k^e$$

$$\forall k \in K, \forall g \in G,\ \sum_{v \in V_{a_k}} \chi(v,g) w_v + \sum_{e \in E_{a_k}} \overline{\chi}(v_e^s, g)\chi(v_e^t, g) d_e \le M_g$$

性能要件
- スループット要件
  - 見積もられたスループットが要件以上
- 実行時間要件
  - 見積もられた実行時間が許容時間以下
- メモリ要件
  - モデルの分割のデータサイズが割当資源のメモリに収まる

目的関数:

$$minimize \quad \sum_{g \in G} 1_{\sum_{v \in V_{a_k}} \chi(v,g) > 0} C_g^g + \sum_{r \in R} \sum_{e \in E_{a_k}} v(e,r) \sum_{l \in r} C_l^l$$

RA-CNP と同様にAnt Colony Optimization によって解を導出

44

モデル分割・資源割当問題の定義

- 性能要件を満たしながら将来のタスクに必要な資源の利用を最小化するモデル分割・資源割当の組み合わせを決定

モデル分割・資源割当問題の定義

制約条件:

$$\forall v \in V_{a_k},\ \sum_{g \in G} \chi(v,g) = 1$$

$$\forall e \in E_{a_k},\ \sum_{g^1,g^2 \in G} \sum_{r \in R_{g^1,g^2}} v(e,r) = \chi(v_e^s, g^1) \cdot \chi(v_e^t, g^2)$$

$$\forall k \in K,\ \gamma_k \le P_k$$

$$\forall k \in K,\ \delta_k \ge T_k^e$$

$$\forall k \in K, \forall g \in G,\ \sum_{v \in V_{a_k}} \chi(v,g) w_v + \sum_{e \in E_{a_k}} \overline{\chi}(v_e^s, g)\chi(v_e^t, g) d_e \le M_g$$

割当資源とリンクの総コストが最小となるようなモデル分割と割当を決定

目的関数:

$$minimize \quad \sum_{g \in G} 1_{\sum_{v \in V_{a_k}} \chi(v,g) > 0} C_g^g + \sum_{r \in R} \sum_{e \in E_{a_k}} v(e,r) \sum_{l \in r} C_l^l$$

RA-CNP と同様にAnt Colony Optimization によって解を導出

45

評価内容

- 資源割当に成功したタスク数の評価
  - 割当資源とその数も含めた最適化によって より多くのタスクを実行できることを示すため
  - 比較対象：RAMPA、 比較手法 (右表参照)
    - 比較手法はモデル分割数について 20 通り計測

➢以降で結果を説明

	目的	分割数最適化
RAMPA	将来のタスクに必要な資源の割当を最小化	○
NCAR		×
PE (従来手法)	スループット最大化	×

本評価の比較手法

- 割当タスクのスループットと実行時間における RAMPA と従来手法の比較
  - RAMPA と従来手法におけるタスクの性能の違いを考察するため
  - 比較対象：RAMPA、 PE

➢従来手法よりも性能が低くなるものの、性能要件を満たすことができることを確認

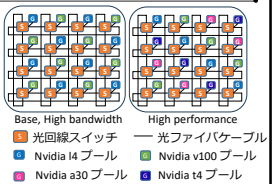
- RAMPA の計算時間の評価
  - RAMPA が現実的な時間内で処理可能であることを示すため

➢10 秒以内で資源割当処理を完了でき、実行前の資源割当として許容可能であることを確認

46

評価環境

- ネットワーク
  - 複数種の GPU で構成された GPU クラスタを構成
    - Base: 比較のための基準
    - High performance: 高性能 GPU のみで構成
    - High bandwidth: 高帯域幅のネットワークで構成
- 資源割当要求
  - 3 種のサービスタスクに対する資源割当要求を生成
    - High throughput：スループット要件が高い
    - Low delay：許容時間が短い
    - Huge model：利用モデルサイズが大きい
  - 各種タスクが高頻度で要求される 3 環境と全タスクの要求頻度が同じ環境で評価
- 評価指標：性能要件を満たす割当に成功した資源割当要求数



性能要件	High throughput	Low delay	Huge model
許容時間 (s)	0.1 s	0.05 s	10 s
スループット (tps)	60	30	0.1
利用モデル	yolos-base	vit-huge	gemma-2-27b

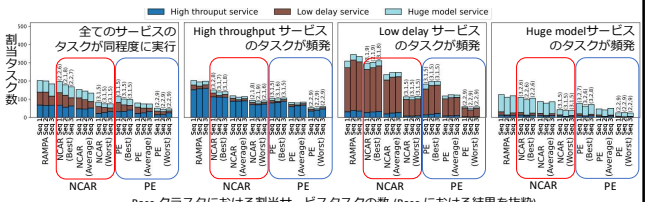
各サービスタスクの許容時間とスループット要件

47

RAMPA の目的設定の有効性評価

- 目的設定においてのみ異なる NCAR と PE を比較
  - NCAR は RAMPA と同様の目的設定

➢NCAR は PE よりも多くのタスクの割当に成功し、RAMPA の目的設定の有効性を確認

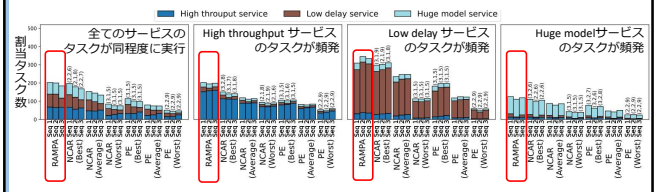


Base クラスタにおける割当サービスタスクの数 (Base における結果を抜粋)

48

モデル分割数も含めた資源割当の最適化の効果

- RAMPA が任意の環境において他手法より多くのタスクの割当に成功
  - ➢割当資源数も含めた資源割当の最適化により実行可能タスク数を向上可能



Base クラスタにおける割当サービスタスクの数 (Base における結果を抜粋)

49

Chapter 4 のまとめ

- DL モデルの分割を最適化することで将来のタスクのために必要な資源の割当を最小化する手法 RAMPA を提案
  - DL モデルの分割と割当資源の組み合わせに基づくスループット・実行時間の見積もり
  - 性能要件を満たしながら将来のタスクに必要な資源の割当を最小化するモデル分割と資源割当の組み合わせを決定
- RAMPA により、多くのタスクを同時実行できることを確認
  - タスクの性能要件を満たしながら、従来手法よりも多くのタスクの割当に成功
  - ➢割当資源数も含めた資源割当の最適化によってより多くのタスクを実行可能

50

博士論文のまとめ

- 資源割当や光ネットワークトポロジの設計により、多くのタスクを同時実行可能な μDDC を構成可能
  - 将来のタスクの実行のために必要な資源の割当をできるだけ避ける
  - ➢タスクの要求時に必要な資源が利用可能となる状態を確保
  - 性能要件を満たすのに十分低遅延な通信が可能な資源ペアの数を最大化させるようにネットワークトポロジを構成
  - ➢多くのタスクの実行下でも資源割当の柔軟性を保ち、より多くのタスクの実行を実現
- 今後の課題
  - タスクの実行時に資源間で発生するトラヒック等の実行情報の見積もり手法の検討
    - タスクの実行に必要なクロック数やデータ量、通信されるデータ量は事前のプロファイルで把握可能という前提を置いていた
      - ➢ オフローディングのような事前にどのような処理を行うか把握できない処理への対応が困難
    - 資源割当が性能に及ぼす影響を拡張することで対応

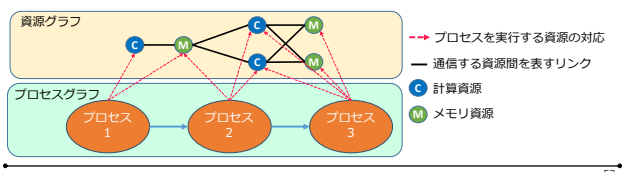
51

付録

52

Chapter2: RA-CNP における資源割当要求

- 資源割当要求を資源グラフとプロセスグラフによってモデル化
  - 資源グラフ：通信する割当資源間の関係を表すグラフ
  - プロセスグラフ：タスクの実行に必要な処理の実行順を表すグラフ
- 資源グラフの各ノードにμDDC 資源をマッピング
- 資源グラフの核リンクにμDDC 内の経路 (リンクの集合) をマッピング



53

Chapter2: ACO に基づく資源割当

- 複数のエージェントが、資源割当コストとフェロモンに基づき確率的に解を探索
  - フェロモン：各資源とリンクに付与される値
  - 低いコストかつ、高い値のフェロモンを持つ資源やリンクが選択される確率が高い

資源割当のステップ

1. 資源探索フェーズ
   各資源のコストとフェロモンに基づき、割当資源を確率的に選択
2. ネットワークリンク探索フェーズ
   1. 資源探索フェーズで選択した各資源の組み合わせに対し、複数のエージェントを生成
   2. 各エージェントが資源間の経路となるリンクを1ホップずつ確率的に選択
3. 実行時間計算フェーズ
   提案する実行時間の見積もりをもとに、選択した資源と経路の組み合わせで性能良い鵜権を満たすことができるかを検証
4. フェロモン更新フェーズ
   最もコストの低かった組合わせの資源やリンクのフェロモンを高く設定
   それ以外の資源やリンクのフェロモンを低く設定

本研究では上記の処理を 20 回繰り返す

54

Chapter2: CSTE の定義

CSTE $CSTE(x,T) = \sum_{t \in T} A_t \cdot \left(\prod_{b,c \in R_t} \frac{V_{b,c}(t,\theta(x,T))}{|S_c|} \right)$

最大 HR $\theta(x,T) = \max\left(k \in \mathbb{N} \mid \forall i,j \in E,\ D_{i,j}(x,k,T) \le x_{i,j}\right)$

隣接ノード間を通過する光パスの数の期待値 $D_{i,j}(x,k,T) = \sum_{g \in V} \frac{\sum_{t \in T} \sum_{b,c \in R_t} Q_{i,j,g}(x,k,t,b,c)}{\kappa_g}$

その隣接ノードを通過する光パスで通信する資源ペア数の期待値 $Q_{i,j,g}(x,k,t,b,c) = \sum_{s \in J_b} \sum_{p \in C_{s,c}(t,k)} P^c_{i,j,g}(x,k,t,s,p,c)$

55

55

Chapter2: CSTE の定義

最大 HR 内かつ性能要件を満たすために許容できる数のスイッチを通過する経路で通信できる資源数

実行タスクごとに導出しタスクの到着割合で加重平均をとる

μDDC 内の資源の総数

そのタスクの実行時通信する資源ペアの種別

CSTE $CSTE(x,T) = \sum_{t \in T} A_t \cdot \left(\prod_{b,c \in R_t} \frac{V_{b,c}(t,\theta(x,T))}{|S_c|} \right)$

最大 HR $\theta(x,T) = \max\left(k \in \mathbb{N} \mid \forall i,j \in E,\ D_{i,j}(x,k,T) \le x_{i,j}\right)$

隣接ノード間を通過する光パスの数の期待値 $D_{i,j}(x,k,T) = \sum_{g \in V} \frac{\sum_{t \in T} \sum_{b,c \in R_t} Q_{i,j,g}(x,k,t,b,c)}{\kappa_g}$

その隣接ノードを通過する光パスで通信する資源ペア数の期待値 $Q_{i,j,g}(x,k,t,b,c) = \sum_{s \in J_b} \sum_{p \in C_{s,c}(t,k)} P^c_{i,j,g}(x,k,t,s,p,c)$

56

56

Chapter2: CSTE の定義

各隣接ノードを通過する光パスの数が
そのノード間の光ファイバ数以下であるかときの HR の最大値を導出

CSTE $CSTE(x,T) = \sum_{t \in T} A_t \cdot \left(\prod_{b,c \in R_t} \frac{V_{b,c}(t,\theta(x,T))}{|S_c|} \right)$

最大 HR $\theta(x,T) = \max\left(k \in \mathbb{N} \mid \forall i,j \in E,\ D_{i,j}(x,k,T) \le x_{i,j}\right)$

隣接ノード間を通過する光パスの数の期待値 $D_{i,j}(x,k,T) = \sum_{g \in V} \frac{\sum_{t \in T} \sum_{b,c \in R_t} Q_{i,j,g}(x,k,t,b,c)}{\kappa_g}$

その隣接ノードを通過する光パスで通信する資源ペア数の期待値 $Q_{i,j,g}(x,k,t,b,c) = \sum_{s \in J_b} \sum_{p \in C_{s,c}(t,k)} P^c_{i,j,g}(x,k,t,s,p,c)$

57

57

Chapter2: CSTE の定義

その隣接ノードを通過する光パスで通信する資源ペア数の期待値

全 PS ペアについて
そのノード間を通過する光パスの数を導出

PS の 1 ポートで収容できる資源ペアの通信数

CSTE $CSTE(x,T) = \sum_{t \in T} A_t \cdot \left(\prod_{b,c \in R_t} \frac{V_{b,c}(t,\theta(x,T))}{|S_c|} \right)$

最大 HR $\theta(x,T) = \max$ その隣接ノードを通過する光パスで通信する資源ペア数の期待値 $T) \le x_{i,j})$

隣接ノード間を通過する光パスの数の期待値 $D_{i,j}(x,k,T) = \sum_{g \in V} \frac{\sum_{t \in T} \sum_{b,c \in R_t} Q_{i,j,g}(x,k,t,b,c)}{\kappa_g}$

その隣接ノードを通過 $Q_{i,j,g}(x,k,t,b,c) = \sum_{s \in J_b} \sum_{p \in C_{s,c}(t,k)} P^c_{i,j,g}(x,k,t,s,p,c)$
で通信する資源ペア

58

58

Chapter2: CSTE の定義

資源ペアがその隣接ノードを通過する光パスで通信する確率
- タスクの実行割合
- 資源ペアの通信確率
- PS ペアで確立される光パスを利用する確率
- その光パスが隣接ノードを通過する確率

通信する可能性のある資源ペアの組み合わせ

CSTE $CSTE(x,T) = \sum_{t \in T} A_t \cdot \left(\prod_{b,c \in R_t} \frac{V_{b,c}(t,\theta(x,T))}{|S_c|} \right)$

最大 HR $\theta(x,$

隣接ノード間を通過する光パスの数の期待 $D_{i,j}(x,k,T) = \sum_{g \in V} \frac{\sum_{t \in T} \sum_{b,c \in R_t} Q_{i,j,g}(x,k,t,b,c)}{\kappa_g}$

その隣接ノードを通過する光パスで通信する資源ペア数の期待値 $Q_{i,j,g}(x,k,t,b,c) = \sum_{s \in J_b} \sum_{p \in C_{s,c}(t,k)} P^c_{i,j,g}(x,k,t,s,p,c)$

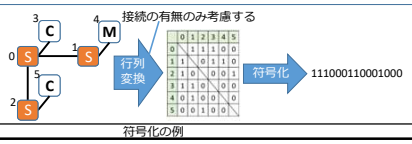
59

59

Chapter3: GA に基づくトポロジ生成

- ネットワークトポロジを符号化し、交叉、変異を繰り返すことでトポロジを生成
  - 符号化：ネットワークトポロジの接続を 0と1 によって表現
  - 変異：符号化された 2 トポロジの選択箇所の値を入れ替え
  - 変異：符号化されたトポロジで、ランダムに選択された箇所の 0 と 1 を反転

処理手順
1. 符号化、変異、交叉を行い、新たなトポロジを生成
2. 生成したトポロジに対して、各ノード間を通過する光パスの数を見積もり
3. 見積もり分だけ各ノードにリンクを追加
4. リンクを追加しても各スイッチのポート制限を超えない場合、CSTE を導出
5. 1~4を CSTE=1 となるトポロジが生成されるまで一定回数繰り返し



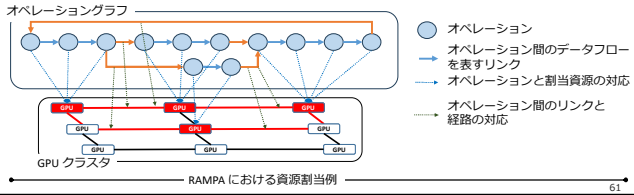
符号化の例

60

60

Chapter4: RAMPA における資源割当

- DL モデルの各層の関係を表すオペレーショングラフを定義
  - オペレーション：パイプライン化が可能な層の最小単位
- 各オペレーションに対して、実行する資源を割当
- 割当資源が異なるオペレーション間のリンクに対して経路を割当



RAMPA における資源割当例

61

61

Chapter4: ステージの実行時間の見積もり

- 割当資源における、DL モデルの各層の実行にかかる時間をもとに導出
  - 各層の実行時間は事前のプロファイルをもとに設定される

$$T^e_{a,v} = \sum_{g \in G} \chi(v,g) \left(\sum_{v' \in V_a} \chi(v',g) t^g_{v',g} \right)$$

そのステージに対応する層の集合

割当GPU と各層の組み合わせにおける実行時間

Chapter 4 の評価における各層の実行時間の設定方法

層のFLOPsをGPUのFLOPSで除算

$$t^g_{v,g} = \frac{FLOPs(v)}{f_g} + \frac{d_v}{Memory\_Band(g)}$$

処理データサイズをメモリ帯域幅で除算

62

62