

リソース分離型 マイクロデータセンターにおける ネットワーク構成の評価指標

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 2年
生駒 昭繁

2024/3/1 1

1

研究背景

- マイクロデータセンターを利用したエッジサービスの提供
 - 通信遅延の削減やトラフィックの削減を実現
 - 大規模データセンターと比べ保有資源に限りがある
- リソース分離型マイクロデータセンター (μDDC) が提案
 - 資源単位で構成されたマイクロデータセンター
 - 柔軟なスケーリングや資源利用率の向上の実現

サーバ中心のマイクロデータセンター μDDC

実行に必要なだけ資源割当 自由に資源を更新/追加

2

2

μDDC におけるタスクの実行

- タスク開始前に資源割り当ての要求が到着
- 要求に対して、随時、実行に必要な計算資源、メモリ資源、それらの間のネットワーク資源を割り当て

μDDC におけるタスクの実行

3

3

μDDC におけるタスクの実行

- タスク開始前に資源割り当ての要求が到着
- 要求に対して、随時、実行に必要な計算資源、メモリ資源、それらの間のネットワーク資源を割り当て

μDDC におけるタスクの実行

4

4

μDDC におけるタスクの実行

- タスク開始前に資源割り当ての要求が到着
- 要求に対して、随時、実行に必要な計算資源、メモリ資源、それらの間のネットワーク資源を割り当て

μDDC におけるタスクの実行

5

5

μDDC におけるタスクの実行

- タスク開始前に資源割り当ての要求が到着
- 要求に対して、随時、実行に必要な計算資源、メモリ資源、それらの間のネットワーク資源を割り当て

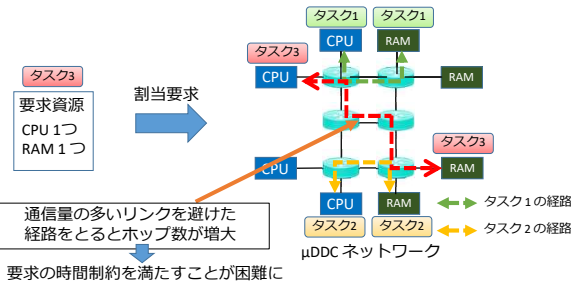
μDDC におけるタスクの実行

6

6

μDDC におけるタスクの実行

- タスク開始前に資源割り当ての要求が到着
- 要求に対して、随時、実行に必要な計算資源、メモリ資源、それらの間のネットワーク資源を割り当て



7

リソース分離の課題

- 資源間の通信遅延がサービスの性能に影響
 - 性能要件を満たすことができない可能性がある
- ある資源間の通信が他資源間の通信や資源割り当てを制限
- 多くの処理を実行するために、ネットワークが性能に及ぼす影響を考慮したμDDCに向けたネットワークの構成が必要
 - 資源間の通信が他の資源間の通信を阻害しない
 - 各資源間でできるだけ少ないホップで通信可能



- このような資源のペアが多ければ、多くのタスクを実行可能

8

μDDC ネットワーク評価指標の提案

- 研究の目的
 - 多くのタスクを実行可能な μDDC であるかを評価する指標の提案
- アプローチ
 - ネットワークの要件として、ホップ数とリンクの混雑度を考慮
 - ホップ数：各計算資源から一定ホップ内にどれだけ資源が存在するか
 - 混雑度：資源間の通信時に、輻輳が生じるリンクは存在しないか
 - 各計算資源に対する、必要な要件を満たす通信が可能な資源の数（候補資源数）を指標として定義
 - 候補資源数が多いほど、実行に必要な資源が枯渇しにくいことを表す
 - 既存のネットワークトポロジに対して指標を導出し、評価

9

μDDC ネットワークに求められる要件

- 計算資源・メモリ資源間の低遅延な接続
 - 計算資源とメモリ資源の通信遅延により、タスクの処理速度が低下
 - 計算資源とメモリ資源は、タスクの性能要件を満たすために十分近接していなければならない



タスクの性能要件を満たすホップ内に通信可能な資源ペアが多く存在

- 多量の計算資源・メモリ資源間の通信の収容
 - 実行タスクが増加すると、ネットワーク内の通信量が増加
 - 通信量の増加により、資源間の通信遅延が増大
 - 資源間の経路上のリンクは、タスクの性能要件を満たすために十分な帯域を確保しなければならない



資源間の同時通信の際に、一部のリンクに資源間の通信が集中しない

10

指標の考え方

- 多くのタスクを実行するために、性能要件を満たす遅延で通信できる資源ペアが多く存在することが必要
 - 各計算資源が通信できる資源の数を指標（候補資源数）として定義

 - 各計算資源が、当該資源から t 番目に近い資源とのみ最短経路で通信すると仮定
 - その状況下において、最大ホップ数と混雑度を導出
 - 最大ホップ数：通信する可能性のある資源間のホップ数の最大値
 - 混雑度：各リンクを利用する資源ペア数の期待値
 - 最大ホップ数・混雑度が要件を満たす t の最大値を導出
 - 各計算資源に対して、 t ホップ以内の資源数を導出

11

11

指標の導出

- 各計算資源が、当該資源から t 番目に近い資源とのみ通信する際の最大ホップ数・リンク混雑度を導出
- それぞれが制約を満たすときの t の最大値を導出

- 最大ホップ数

$$H(t) = \max_{n \in R} \max_{m \in P_n(t)} S_{n,m}$$

- リンク混雑度

$$D_e(t) = \sum_{n \in R} \sum_{m \in P_n(t)} \frac{R_{n,m}(e) C_n(t,m)}{R_{n,m} C_n(t) \cdot t}$$

- 候補資源数

$$t^{\max}(H^{\text{target}}, D^{\text{target}}) = \max t, \begin{cases} H(t) \leq H^{\text{target}} \\ D(t) \leq D^{\text{target}} \end{cases}$$

12

12

指標の導出

1. 各計算資源が、当該資源から t 番目に近い資源とのみ通信する際の最大ホップ数・リンク混雑度を導出
2. それぞれが制約を満たすときの t の最大値を導出

- 最大ホップ数

$$H(t) = \max_{n \in R} \max_{m \in P_n(t)} S_{n,m}$$

任意の計算資源とその資源から t ホップ内の資源との最短ホップの最大値

- リンク混雑度

$$D_e(t) = \sum_{n \in R} \sum_{m \in P_n(t)} \frac{R_{n,m}(e) C_n(t, m)}{R_{n,m} C_n(t) \cdot t}$$

- 候補資源数

$$t^{\max}(H^{target}, D^{target}) = \max t, \begin{matrix} H(t) \leq H^{target} \\ D(t) \leq D^{target} \end{matrix}$$

13

13

指標の導出

1. 各計算資源が、当該資源から t 番目に近い資源とのみ通信する際の最大ホップ数・リンク混雑度を導出
2. それぞれが制約を満たすときの t の最大値を導出

- 最大ホップ数

$$H(t) = \max_{n \in R} \max_{m \in P_n(t)} S_{n,m}$$

- リンク混雑度

$$D_e(t) = \sum_{n \in R} \sum_{m \in P_n(t)} \frac{R_{n,m}(e) C_n(t, m)}{R_{n,m} C_n(t) \cdot t}$$

任意の計算資源とその資源から t ホップ内の資源とがリンク e を通過する確率の総和

- 候補資源数

$$t^{\max}(H^{target}, D^{target}) = \max t, \begin{matrix} \text{リンク } e \text{ を通過する} & \text{資源ペアの通信確率} \\ \text{経路設定確率} & H(t) \leq H^{target} \\ D(t) \leq D^{target} \end{matrix}$$

14

14

指標の導出

1. 各計算資源が、当該資源から t 番目に近い資源とのみ通信する際の最大ホップ数・リンク混雑度を導出
2. それぞれが制約を満たすときの t の最大値を導出

- 最大ホップ数

$$H(t) = \max_{n \in R} \max_{m \in P_n(t)} S_{n,m}$$

- リンク混雑度

$$D_e(t) = \sum_{n \in R} \sum_{m \in P_n(t)} \frac{R_{n,m}(e) C_n(t, m)}{R_{n,m} C_n(t) \cdot t}$$

最大ホップ数、リンク混雑度の要件 H^{target} と、 D^{target} を満たす t の最大値

$$t^{\max}(H^{target}, D^{target}) = \max t, \begin{matrix} H(t) \leq H^{target} \\ D(t) \leq D^{target} \end{matrix}$$

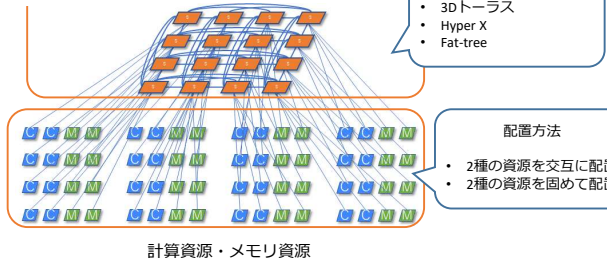
15

15

評価対象の μ DDC ネットワーク構成

- 既存のネットワークトポロジに対して、指標を導出し、資源割当シミュレーションを実施
- 全 8 個のトポロジに対して評価
 - トポロジの構造と資源の配置方法を変更

スイッチ間ネットワーク

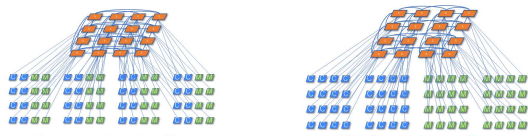


計算資源・メモリ資源

16

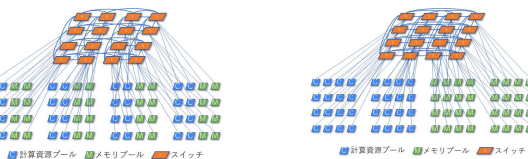
16

比較トポロジ (1/2)



3D トーラス (資源を交互に配置)

3D トーラス (資源を固めて配置)



HyperX (資源を交互に配置)

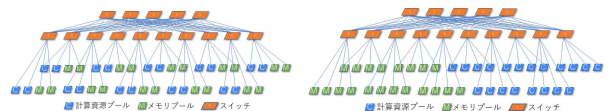
HyperX (資源を固めて配置)

17

17

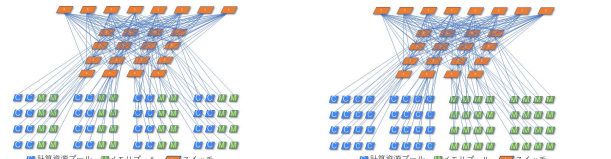
比較トポロジ (2/2)

- 他トポロジと資源数を同数にするために、スイッチ数が異なる2パターンを評価



Fat-Tree_A (資源を交互に配置)

Fat-Tree_A (資源を固めて配置)



Fat-Tree_B (資源を交互に配置)

Fat-Tree_B (資源を固めて配置)

18

18

各トポロジの指標の値

- ホップ数と混雑度の要件ごとの、指標が最大のトポロジを調査
- 全環境で、資源を交互に配置する場合は最大
 - 異種の資源が隣接するため、ホップ数を削減したため
- HyperX が多くの環境で最大
 - 各ノードに対する隣接ノード数が最大
 - 近接資源や柔軟な経路設定が可能

各要件における指標が最大のトポロジ

資源割当による評価

- 既存トポロジに対して、資源割当シミュレーションを実施
- どれだけの資源を割り当てるのが可能かを評価
 - 指標の有効性を確認
- 私たちが以前提案した資源割当手法[1]で実施
 - 将来要求されるタスクが必要な資源を残すように割当
- 4つの状況を想定し、シミュレーションを実施
 - タスクの実行に必要な資源の数と、時間制約を変更

	タスクの時間制約が長い	タスクの時間制約が短い
タスクの実行に必要な資源が少ない	Case 1	Case 2
タスクの実行に必要な資源が多い	Case 3	Case 4

各ケースにおける、必要資源数とタスクの時間制約の対応

[1] Disaggregated Micro Data Center: Resource Allocation Considering Impact of Network on Performance

資源割当結果

- 各ケースにおいて、HyperXの資源利用率が最大
 - 候補資源数が大きいほど多くの資源の割り当てに成功
- 資源を交互に配置することで資源利用率が高い
 - 候補資源数との対応関係と一致

まとめと今後の課題

- まとめ
 - μDDCにおけるネットワーク構成の評価指標 候補資源数 を定義
 - 各計算資源に対する、ホップ数・混雑度の要件を満たす資源の数
 - 既存トポロジに対して指標を導出し、資源割当を実施
 - 候補資源数が多いほど多くの資源の割当が可能であることを示した
- 今後の予定
 - ホップ数や混雑度の要件の決定方法の検討
 - 指標をもとにした最適なトポロジの設計手法の検討