

# 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

生駒 昭繁<sup>†</sup> 大下 裕一<sup>†</sup> 反町 将<sup>†</sup> 村田 正幸<sup>†</sup>

<sup>†</sup> 大阪大学 大学院情報科学研究科 〒565-0871 大阪府吹田市山田丘 1-5

E-mail: †{a-ikoma,y-ohsita,a-ikoma,murata}@ist.osaka-u.ac.jp

**あらまし** リソース分離型マイクロデータセンター ( $\mu$ DDC) において、多くのアプリケーションをその性能要件を満たしながら実行するためには、性能要件を満たすために必要な遅延時間内で通信することが可能な資源ペアが十分に存在することが必要である。そして、資源間の通信遅延は、通信経路のホップ数や通信量が大きく影響する。そこで、資源ペアの通信経路として利用するリンクに対して、通信がどれだけ混雑しやすいか、資源間の通信のために必要なホップ数の2点に注目し、各計算資源に対して、どれだけの資源を通信相手とできるかを指標として定義した。指標の有効性を示すために、複数のネットワークトポロジで構成された  $\mu$ DDC に対して資源割当シミュレーションを行い、指標の値を比較を行った結果、指標の値が大きいほどより多くのアプリケーションの割当が可能であることを示した。

**キーワード** マイクロデータセンター、リソース分離、ネットワークトポロジ

## Performance evaluation of remote memory utilization method in disaggregated micro data center

Akishige IKOMA<sup>†</sup>, Yuichi OHSITA<sup>†</sup>, Masashi SORIMACHI<sup>†</sup>, and Masayuki MURATA<sup>†</sup>

<sup>†</sup> Graduate School of Information Science and Technology, Osaka University

Yamadaoka 1-5, Suita, Osaka, 565-0871 Japan

E-mail: †{a-ikoma,y-ohsita,a-ikoma,murata}@ist.osaka-u.ac.jp

**Abstract** In a disaggregated micro data center ( $\mu$ DDC), ensuring the execution of numerous applications while satisfying their performance requirements necessitates many resource pairs capable of communicating within the required latency. The communication latency between resources significantly depends on factors such as the hop count between communicating resources and the traffic. We focus on two aspects-how congested the communication is on the links used as communication paths for resource pairs, and the hop count required for communication between resources. We define an metrics that measures how many resources each computing resource can communicate with, considering these factors. To evaluate the effectiveness of the defined metrics we conducted a resource allocation simulation. The results indicate that higher values of the indicator allow for the allocation of more applications, demonstrating its efficacy.

**Key words** Micro data center, Resource disaggregation, network topology

### 1. はじめに

近年、多くのクラウドサービスが展開されているが、クラウドを用いた処理は、ユーザ側から離れた場所にあるデータセンターとの通信を必要とするため、自動運転のようなタイムセンシティブな処理には不向きである。そこで、エッジに配置する比較的小規模のデータセンターであるマイクロデータセンターが提案されている [1]。ただし、マイクロデータセンターは、大

規模なデータセンターと比べて保有する資源量が限られているため、多種多様なサービスを展開するためには、資源利用効率の向上が重要となる。資源の効率的な利用を実現する方法の一つとして、CPU やメモリ等の資源を独立させ、それらをネットワークでつなぐリソース分離があげられる [2]。リソース分離を行ったデータセンターでは、資源単位での管理が可能であり、従来のサーバ中心のデータセンターと比較して、資源利用率の向上が可能である [3]。そこで、私たちは、リソース分離

を適用したマイクロデータセンターである、リソース分離型マイクロデータセンター（以降、 $\mu$ DDC）を検討している。しかし  $\mu$ DDC では、資源同士がネットワークによって接続するため、資源間の通信遅延により、実行アプリケーションの性能が低下する [4]。特に、CPU 等の計算資源とメモリ資源との遅延の影響が大きく、各資源や資源間の通信経路の割当方法にアプリケーションの実行性能が大きく依存する。

私たちは、ネットワークがアプリケーションの処理性能に及ぼす影響を考慮した、資源割り当て制御手法 [5] をすでに提案しており、その有効性を確認している。ただし、資源割当はネットワークに大きく依存しており、特定のリンクに通信が集中してしまうなど、柔軟な資源割当が困難な  $\mu$ DDC ネットワークトポロジの場合、資源の利用効率は大きく低下してしまう可能性がある。そのため、より効率的な資源利用のためには、柔軟な資源割当が可能なネットワークトポロジについての検討が必要である。

文献 [6] では、リソース分離型のアーキテクチャに向けたネットワークトポロジが提案されており、複数の木構造を並列に構成した Monet というネットワークトポロジが提案されている。このトポロジは、資源間のホップ数を削減することで、低消費電力で資源間を低遅延で接続することができることが示されているが、規模が大きく、小規模の  $\mu$ DDC においては、不向きであるといえる。そのため、 $\mu$ DDC に向けたトポロジの提案が必要となる。

本研究では、多種多様なサービスを提供する  $\mu$ DDC に最適なネットワークを構成するために必要な、 $\mu$ DDC ネットワークの評価指標を提案する。 $\mu$ DDC において、資源割当は実行サービスの性能を決定づけるため、様々なサービスを提供するためには、柔軟な資源割当を行うことが可能なトポロジの構成が必要不可欠である。このとき、 $\mu$ DDC において処理を実行する際、実行資源と、実行資源間の通信経路を設定する。そのため、通信可能な資源ペアの数が多ければ多いほど、柔軟な資源割当が可能にあるといえる。本研究では、処理を実行する CPU や GPU のような計算資源を中心に、各計算資源に対して、通信可能な資源の数を指標として定義し、その指標の有効性を議論する。

本稿の構成は以下の通りである。2 章で提案する指標の定義を行い、3 章において、定義した指標が  $\mu$ DDC ネットワークの評価のために有効であるかを既存トポロジとの比較によって示す。最後に、4 章で今後の課題とまとめについて示す。

## 2. 評価指標

$\mu$ DDC において、多くのアプリケーションをその性能要件を満たしながら実行するためには、性能要件を満たすために十分な遅延で通信できる資源ペアが多く存在することが必要である。そこで、各計算資源が通信できる資源の数を指標として定義する。本稿では、この指標を通信候補資源数と呼ぶ。実行アプリケーションの性能要件を満たすためには、資源間のホップ数が十分小さく、資源間の経路上のトラヒックが多すぎないことが必要である。そこで、資源間のホップ数と各リンクがどれ

だけ混雑しやすいかに注目して通信候補資源数を定義する。

本研究では、通信候補資源数の導出のために、各資源が当該資源から近い順に一定数の資源を候補とし、それぞれが最短経路で通信するような状況を仮定する。そして、その状況下における各リンクを経由して通信する資源ペアの数の期待値の最大値と通信する可能性のある資源間のホップ数の最大値を導出する。それぞれ、リンク混雑度、最大ホップ数とする。そして、リンク混雑度、最大ホップ数をもとに、通信候補資源数を導出し、各計算資源に対して十分な数の通信可能資源が存在するかを調査することで、 $\mu$ DDC ネットワークを評価する。

### 2.1 記号定義

通信候補資源数の導出に利用する記号を表 3 に示す。

### 2.2 通信候補資源数の導出

$\mu$ DDC 上の各資源が、当該資源から  $t$  番目に近い資源までを通信候補とするときの最大ホップ数  $H(t)$  とリンク混雑度  $D(t)$  を導出し、それらと実行アプリケーションの性能要件をもとに、通信候補資源数を導出する。

#### a) 最大ホップ数 $H(t)$ の導出

資源  $n, m \in R$  間の最短ホップ数を  $S_{n,m}$  とし、最大ホップ数は、資源  $n \in R$  と当該資源から  $t$  番目に近い資源候補  $m \in P_n(t)$  の各資源ペアの最短ホップ数の最大値より、

$$H(t) = \max_{n \in R} \max_{m \in P_n(t)} S_{n,m} \quad (1)$$

である。

#### b) リンク混雑度 $D(t)$ の導出

まず、各リンク  $e \in E$  に対する、当該リンクを経由して通信する資源ペアの数の期待値を  $D_e(t)$  とする。本研究において、各資源ペアの経路は、最短経路の中からランダムに選択されるとする。よって、

資源  $n \in R$  から近い順に  $t$  個選択した資源候補  $m \in P_n(t)$  の最短経路数を  $R_{n,m}$ 、リンク  $e$  を経由する最短経路数  $R_{n,m}(e)$  とし、リンク  $e$  が資源  $n, m$  間の経路になる確率を  $\frac{R_{n,m}(e)}{R_{n,m}}$  とする。また、各資源は通信候補の中からランダムに通信相手を選択するとする。よって、資源  $n$  から  $t$  番目に近い資源までを通信候補としたときの資源候補の組み合わせの数を  $C_n(t)$ 、そのうち資源  $m$  を含む組み合わせの数を  $C_n(t, m)$ 、として、資源  $n$  と  $m$  の通信確率は、 $\frac{C_n(t, m)}{C_n(t) \cdot t}$  とできる。よって、各リンクを経由して通信する資源ペアの数の期待値  $D_e(t)$  は、

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

である。そして、この混雑度をもとに、 $\mu$ DDC の各リンク  $e \in E$  の混雑度の最大値がリンク混雑度であるので、

$$D(t) = \max_{e \in E} D_e(t) \quad (2)$$

である。

#### c) 最大ホップ数・リンク混雑度に基づく通信候補資源数 $t^{\max}(H^{\text{target}}, D^{\text{target}})$ の導出

$\mu$ DDC 上の各資源が、当該資源から  $t$  番目に近い資源までを通信候補とするときにパラメータ  $t$  は各資源が通信可能な資源の

表 1: 通信候補資源数の導出に利用する記号の定義

記号	内容
$R$	$\mu$ DDC 上の資源の集合
$E$	$\mu$ DDC 上のリンクの集合
$H(t)$	最大ホップ数
$D(t)$	リンク混雑度
$S_{n,m}$	資源 $n, m \in R$ 間の最短ホップ数
$P_n(t)$	資源 $n \in R$ と当該資源から $t$ 番目に近い資源の集合
$D_e(t)$	各リンク $e \in E$ に対する、当該リンクを経由して通信する資源ペアの数の期待値
$R_{n,m}$	資源 $n, m \in R$ 間の最短経路数
$R_{n,m}(e)$	$R_{n,m}$ のうち、リンク $e \in E$ を通過する経路の数
$C_n(t)$	資源 $n$ が $t$ 番目に近い資源までを通信候補としたときの資源候補の組み合わせの数
$C_n(t, m)$	$C_n(t)$ のうち、資源 $m$ を含む資源候補の組み合わせの数
$H^{target}$	実行アプリケーションの性能要件を満たすためのホップ数の制約
$D^{target}$	実行アプリケーションの性能要件を満たすためのリンク混雑度の制約
$t^{max}(H^{target}, D^{target})$	ホップ数制約 $H^{target}$ 、リンク混雑度制約 $D^{target}$ の際の通信候補資源数

数とみなすことができ、パラメータ  $t$  が大きいほど各計算資源は多くの通信候補を持つといえる。そこで、実行アプリケーションの性能要件を満たすためのホップ数の制約  $H^{target}$  と混雑度の制約  $D^{target}$  をもとに、通信候補資源数  $t^{max}(H^{target}, D^{target})$  を導出する。これは、最大ホップ数とリンク混雑度が制約をみたすときの  $t$  の最大値であるので、

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

である。この値が大きいほど、各計算資源が通信相手とできる資源が多いため、より柔軟な資源割当が可能である。

### 3. 評価

既存のネットワークについて、通信候補資源数を導出し、資源割当シミュレーション結果と比較することで、評価指標の有効性を示す。

#### a) 比較ネットワーク

評価のために、一般的に広く用いられている、3D トーラストポロジ、FatTree トポロジと、ノード間のホップ数をより小さく抑えることができる HyperX [7] トポロジを利用する。なお、比較のために、計算資源とメモリ資源の数を統一してネットワークを構成する。各ネットワーク構成のスイッチ数と資源数を表 2 に示す。ただし、FatTree は、構成のために必要とするスイッチの数が他トポロジとは異なるため、資源プール内に存在する資源数を増加させることで保有資源数を統一した場合と、資源プール内に存在する資源数を固定し、構成するスイッチの数を増やすことで保有資源数を統一した場合の 2 パターンについて評価する。それぞれ、FatTree\_1、FatTree\_2 とする。さらに、これらのネットワークについて、一つのスイッチに計算資源とメモリ資源の両方が接続する資源配置  $A$  と、1 つのスイッチに 1 種類の資源しか接続しない資源配置  $B$  の 2 パターン計測する。それぞれ図 1a から図 1h に示す。

#### 3.1 既存ネットワークごとのネットワーク指標とパラメータの値の関係

各トポロジにおける通信候補資源数の比較のために、最大ホップと混雑度の制約  $H^{target}$  と  $D^{target}$  を 1 から 20 まで変化させたときに、その制約における通信候補資源数が最大であるネットワークを図 2 に示す。

図 2 から、どの環境においても、A 型の資源配置のトポロジが B 型の資源配置のトポロジよりも通信候補資源数の値が大きくなっていった。これは、交互に資源を配置することで、資源同士が隣接し、より少ないホップ数で通信可能になるためである。例えば、 $D^{target}$  が 10 であるとき、 $H^{target}$  が 2 や 3 であるときは、A 型の資源配置のトポロジの通信候補資源数が最大となっている。また、 $H^{target} = 2$  かつ、 $8 \leq D^{target}$  の範囲以外において、HyperX(A) は  $t^{max}$  の値が最も大きいネットワークとなっており、より多くの環境において、通信候補資源数の観点から優れているといえる。HyperX は、各ノードに対して、資源と直接接続している隣接ノードの数が最も多いため、近接する資源ペアも多い。さらに、あるノードから 1 ホップで到達できるノードが多いため、選択できる経路の数も多くなる。そのため、少ないホップでの通信を可能としながら、リンク混雑度も小さく抑えることができたといえる。

#### 3.2 ネットワーク指標と性能要件や実行資源数との関係調査

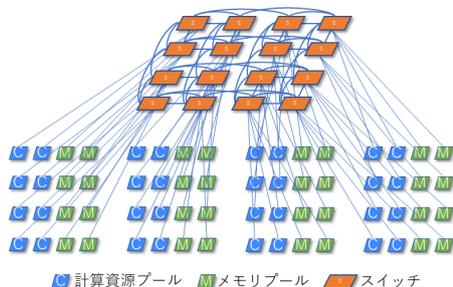
通信候補資源数によるトポロジ評価の有効性を示すために、私たちが提案した、 $\mu$ DDC ネットワークがアプリケーションの性能に与える影響を考慮した資源割当手法 (RA-CNP) [5] を利用して、各トポロジにおいてアプリケーションの資源割当シミュレーションを行い、各トポロジでどれだけのアプリケーションの資源割当が可能であるかを計測し、その結果と通信候補資源数の値を比較する。

##### 3.2.1 計測環境

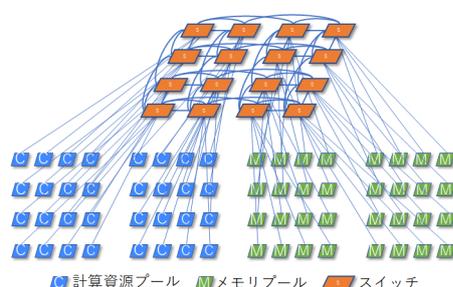
シミュレーションのために、 $\mu$ DDC ネットワークや割当アプリケーションのパラメータを設定する。このとき、図 2 に示すように、通信候補資源数の値は実行サービスの性能要件によっ

表 2: 各ネットワーク構成のスイッチ数と資源数

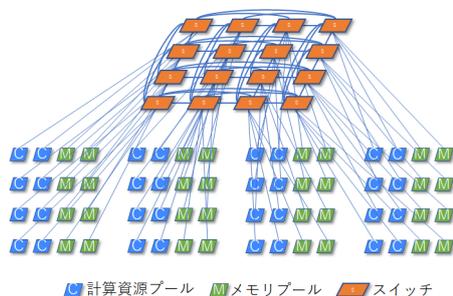
トポロジ	スイッチ数	計算資源プール数	計算資源プール内 計算資源数	メモリ資源プール数	メモリ資源プール内 メモリ資源数	総計算資源数	総メモリ資源数
FatTree_1	15	20	8	20	8	160	160
FatTree_2	24	32	5	32	5	160	160
3D トーラス	16	32	5	32	5	160	160
HyperX	16	32	5	32	5	160	160



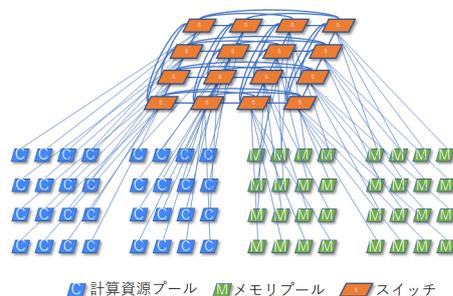
(a) 3D トーラス (A)



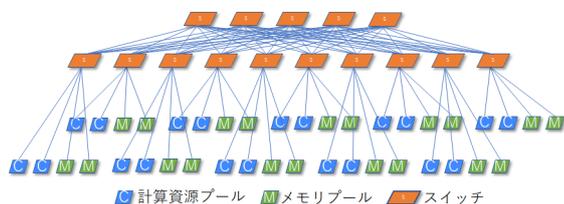
(b) 3D トーラス (B)



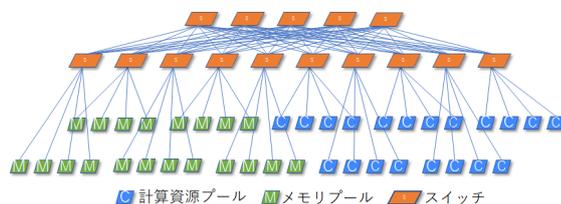
(c) HyperX (A)



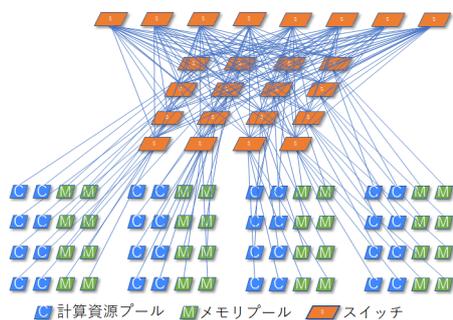
(d) HyperX (B)



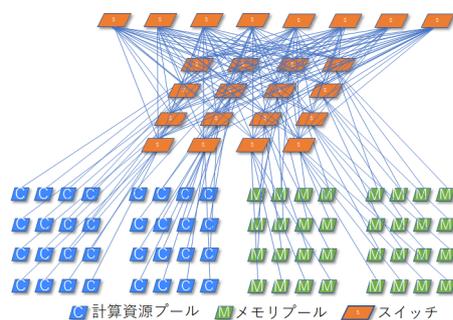
(e) FatTree\_1 (A)



(f) FatTree\_1 (B)



(g) FatTree\_2 (A)



(h) FatTree\_2 (B)

図 1: ネットワーク構成

で変化する。そこで、割当アプリケーションの性能要件や実行に必要な資源の数が異なる複数のケースについて計測を行う。

a) ネットワーク

ネットワークのパラメータについて、表 3 に示す。なお、評価を行うすべてのネットワークにおいて、パラメータは同一で

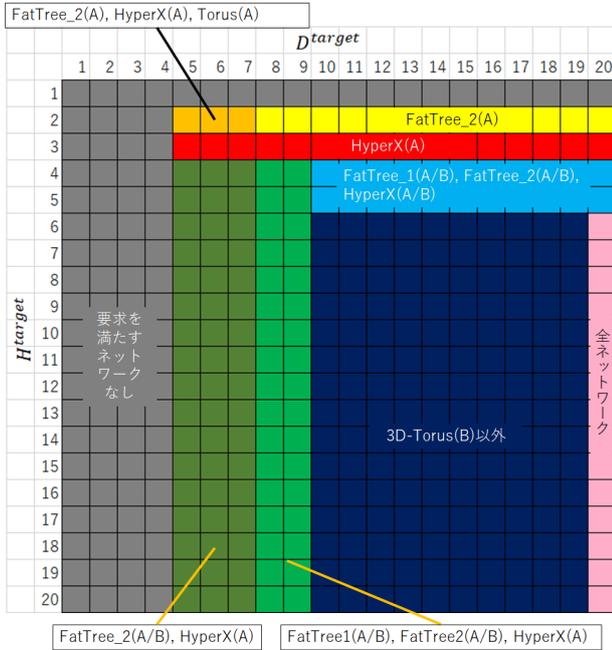


図 2:  $t^{max}$  の値が最も高いネットワーク構造

ある。

表 3:  $\mu$ DDC ネットワークパラメータの設定

Parameters	Value
CPU の FLOPS	108.8GFLOPS
各コアの周波数	3.4GHz
伝搬遅延	0.025 $\mu$ s
スイッチング遅延	3 $\mu$ s
カットスルー時のスイッチング遅延	300 ns
ページサイズ	4 KB
帯域幅	50 Gbps
各ノード間のリンク数	2

### b) 割当アプリケーション

性能要件の異なる 3 種類のアプリケーションについて資源割当を行う。エッジで実行される典型例として ResNet [8]、YOLO [9] を用いた画像分類処理を考える。各アプリケーションの実行要求は同一の確率で生成されるとし、生成されたすべての実行要求が割り当てられた場合、これ以上の実行要求の割当が不可能になる。つまり、生成されたすべての要求を割り当てた場合、そのトポロジは割当可能な上限までアプリケーションを割り当てたことを示す。

また、評価のために、各資源要求の性能要件と必要資源数を変更して以下の 4 つの場合について資源割当を行う。

- Case1 実行アプリケーションの時間制約が長く、実行に必要な資源数が少ない場合
- Case2 実行アプリケーションの時間制約が短く、実行に必要な資源数が少ない場合
- Case3 実行アプリケーションの時間制約が長く、実行に必要な資源数が多い場合
- Case4 実行アプリケーションの時間制約が短く、実行に必要な資源数が多い場合

各アプリケーションの実行性能のパラメータ設定について、表 4 に示す。なお、アプリケーション 1 が ResNet を用いた画像

分類処理、その他のアプリケーションが YOLO を用いた画像分類処理である。

表 4: Parameter settings for the resource requests

	アプリケーション 1	アプリケーション 2	アプリケーション 3
時間制約 (Case 1,3/Case 2,4)	700/350	600/250	500/150
実行に必要な資源数 (Case 1,2/Case 3,4)	3/6	3/6	3/6
Process 1			
クロック数	0.063	0.063	0.063
メモリへの書き込みレート (/ms)	0.000495	0.003	0.0075
メモリからの読み込みレート (/ms)	0.000495	0.003	0.0075
メモリ資源数 (Case 1,2/Case 3,4)	1/1	1/1	1/1
計算資源数 (Case 1,2/Case 3,4)	1/1	1/1	1/1
Process 2			
クロック数	0.097	0.097	0.097
メモリへの書き込みレート (/ms)	0	0	0
メモリからの読み込みレート (/ms)	0.000495	0.000495	0.0075
メモリ資源数 (Case 1,2/Case 3,4)	2/5	2/5	2/5
計算資源数 (Case 1,2/Case 3,4)	1/1	1/1	1/1
Process 3			
クロック数	4268.4	3528.6	3528.6
メモリへの書き込みレート (/ms)	2.805	2.85	2.85
メモリからの読み込みレート (/ms)	5.565	5.145	5.145
ページフォールト発生回数	121577.9	101990.3	101990.3
1 ページフォールトあたりの読み込みページ数	9.5	8.7	8.7
メモリ資源数 (Case 1,2/Case 3,4)	2/5	2/5	2/5
計算資源数 (Case 1,2/Case 3,4)	2/5	2/5	2/5

### 3.2.2 評価指標

資源割当シミュレーションによって、通信候補資源数が大きいほど、より多くのアプリケーションの実行が可能であることを示す。そこで、ネットワーク上の計算資源の総数に対する、割当計算資源の数を計算資源利用率として計測する。この値が大きいほど、より多くの資源ペアの通信を行うことができ、より多くのアプリケーションの実行が可能であるといえる。なお、評価のために、性能要件を満たすような資源割当に失敗した場合は、その要求を棄却する。

### 3.2.3 資源割当結果

各 Case について、それぞれ 5 種類の要求列を生成し、それをもとに評価を行う。結果を図 3 に示し、図 3 の縦軸は資源割当終了時の計算資源利用率を表す。

すべての Case において HyperX は最も多くのアプリケーションを収容することができているが、図 2 に示す結果とも一致しており、様々な場合で各資源が十分な資源候補を持っているために、より多くのアプリケーションの割当に成功したといえる。また、どの Case においても、A 型の資源配置のトポロジの方が、B 型の資源配置の同一トポロジよりも計算資源利用率が高く、これも図 2 の結果と一致する。結果から、通信候補資源数が大きいほど、柔軟な資源割当が可能であり、より多くのアプリケーションを実行可能であるといえる。

## 4. おわりに

本稿では、リソース分離型マイクロデータセンター ( $\mu$ DDC) におけるネットワークの評価指標として計算資源候補数を提案した。本指標は、多種多様なアプリケーションを実行可能な  $\mu$ DDC ネットワークであるかを評価するための指標であり、通信を行う可能性のある資源間は、どれだけのホップ以内で通信可能であるか、ネットワーク上の各リンクはどれだけの資源間の経路として利用される可能性があるかの 2 点に注目し、それぞれが実行アプリケーションの性能要件を満たすときの、各計

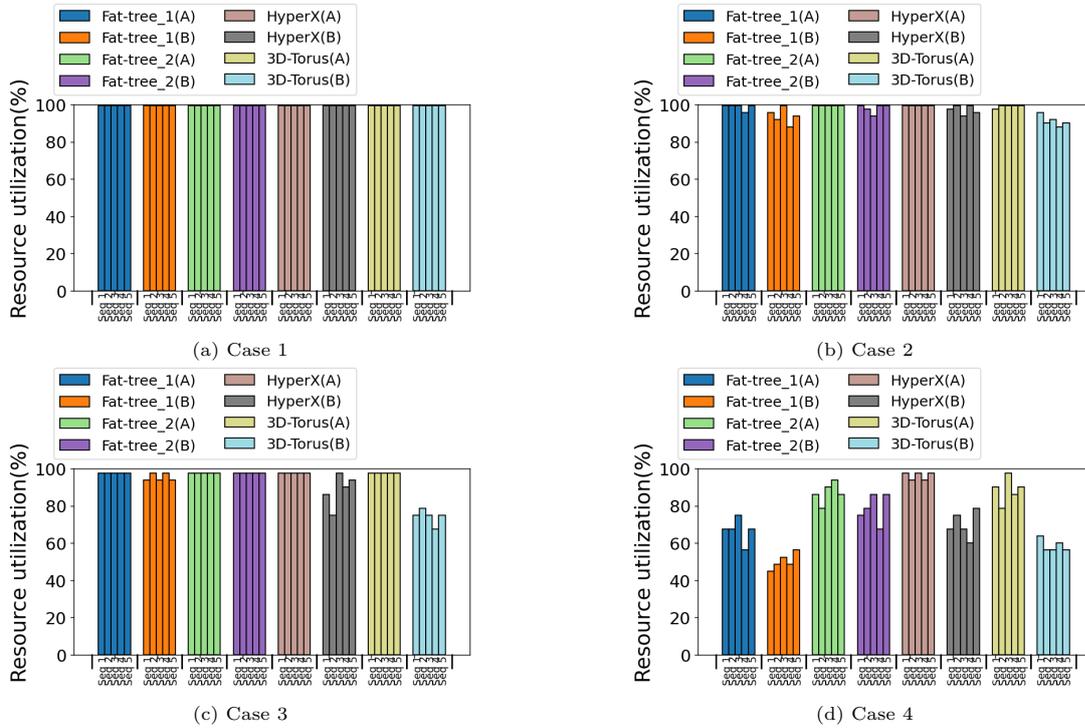


図 3: 各 Case における、各ネットワークトポロジの計算資源利用率

算資源が通信できる資源の個数として導出している。この指標の値が大きいほど、資源割当の際に、多くの実行資源ペアの候補から実行資源を選択できることを表し、柔軟な資源割当にによる、実行アプリケーション数の向上が期待できる。

いパニック的に利用される複数の既存ネットワークトポロジに対して、計算資源候補数を導出し、資源割当を成功した実行要求の数と比較した結果、計算資源候補数が多いネットワークトポロジであるほど、より多くの要求の割当に成功しており、 $\mu$ DDC ネットワークの評価指標として、計算資源候補数の有効性を示した。

今後の予定として、アプリケーションの性能要件の決定方法の検討が挙げられる。本稿では、ホップ数やリンク混雑度の制約はあらかじめ判明している前提で計測を行ったが、実際にアプリケーションを処理する際は、資源の実行性能や、要求量等、制約には様々な要因が影響する。これらについて、適切に考慮した制約の決定方針が必要となる。また、本稿では既存トポロジに対して評価を行ったのみであり、 $\mu$ DDC に最適なトポロジが何であるかは判明していない。指標をもとにした、ネットワークトポロジの構築手法の提案も今後検討が必要な点である。

## 謝 辞

本研究の一部は、NICT 委託研究・JPJ012368C00101「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」によるものである。ここに記して謝意を表す。

## 文 献

- [1] K. Bilal, O. Khalid, A. Erbad, and S. U. Khan, “Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers,” *Computer Networks*, vol. 130, pp. 94–120, Jan. 2018.
- [2] S. Han, N. Egi, A. Panda, S. Ratnasamy, G. Shi, and

S. Shenker, “Network support for resource disaggregation in next-generation datacenters,” in *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks*, pp. 1–7, Nov. 2013.

- [3] G. Zervas, H. Yuan, A. Saljoghei, Q. Chen, and V. Mishra, “Optically disaggregated data centers with minimal remote memory latency: Technologies, architectures, and resource allocation [invited],” *Journal of Optical Communications and Networking*, vol. 10, no. 2, pp. A270–A285, 2018.
- [4] P. X. Gao, A. Narayan, S. Karandikar, J. Carreira, S. Han, R. Agarwal, S. Ratnasamy, and S. Shenker, “Network requirements for resource disaggregation,” in *Proceedings of 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, (Savannah, GA), pp. 249–264, USENIX Association, Nov. 2016.
- [5] A. Ikoma, Y. Ohsita, and M. 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, 2023.
- [6] V. Mishra, J. L. Benjamin, and G. Zervas, “Monet: Heterogeneous memory over optical network for large-scale data centre resource disaggregation,” *Research Article Journal of Optical Communications and Networking*, vol. 1, 2021.
- [7] J. H. Ahn, N. Binkert, A. Davis, M. McLaren, and R. S. Schreiber, “Hyperx: topology, routing, and packaging of efficient large-scale networks,” in *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, pp. 1–11, 2009.
- [8] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.
- [9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.