

# リソース分離型マイクロデータセンターのネットワーク構成

反町 将<sup>†</sup> 生駒 昭繁<sup>†</sup> 大下 裕一<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: †{m-sorimachi,a-ikoma,y-ohsita,murata}@ist.osaka-u.ac.jp

**あらまし** リアルタイム性の高いデータ処理を行うため、ユーザにより近いエッジに配置可能な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 ( $\mu$ DC) の検討が進められている。 $\mu$ DCでは、資源が限られており、多種多量のアプリケーションを実行するためには、限られた資源を有効利用することが求められる。そこで、CPU・GPU・メモリ・ストレージといった資源を分離してネットワークに接続し、各アプリケーションに必要な分だけ資源を割り当てることで資源利用効率を上げることができる、リソース分離型マイクロデータセンター ( $\mu$ DDC) が提案されている。しかしながら、 $\mu$ DDC内のネットワークが $\mu$ DDCで実行されるアプリケーションの処理性能に大きな影響を与える。そのため、本稿では、 $\mu$ DDC内のネットワークに適したネットワーク構造について議論を行う。本稿では、まず、 $\mu$ DDC内のネットワークに求められる要件について議論する。その後、その要件を満たしているのかを評価可能な指標を導入する。そして、既存のデータセンター向けのネットワーク構造に計算資源とメモリ資源を接続した $\mu$ DDC内ネットワーク構成に対して、導入した指標を適用することにより、ネットワークの特性を調べ、 $\mu$ DDCに適したネットワーク構造について議論する。

**キーワード** マイクロデータセンター、リソース分離、ネットワーク構成

## Network Structure for Micro Disaggregated Data Centers

Masashi SORIMACHI<sup>†</sup>, Akishige IKOMA<sup>†</sup>, Yuichi OHSITA<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: †{m-sorimachi,a-ikoma,y-ohsita,murata}@ist.osaka-u.ac.jp

**Abstract** Micro data centers ( $\mu$ DCs), which are small data centers located close to users, have been proposed to execute time-sensitive applications. Efficient resource utilization is important in a  $\mu$ DCs because a  $\mu$ DCs has a limited amount of resources compared with cloud data centers but is required to run many applications. One approach to utilizing resources efficiently is to disaggregate resources such as CPU, GPU, memories and storage. In a disaggregated data center (DDC), resources are connected via network. By allocating only required resources to each application, we can efficiently utilize the resources in DDC. However, in a DDC, the network has a large impact on the performance of the applications. Therefore, in this paper, we discuss network structures suitable for micro disaggregated data centers ( $\mu$ DDCs). We first discuss the requirements for network structures in a  $\mu$ DDC and introduce metrics. Then, we apply the metrics to the network structures constructed based on existing data center network structures. Based on the metric, we investigate their properties and discuss the suitable network structures.

**Key words** Micro data center, Disaggregation, Network structure

## 1. はじめに

クラウドを用いた数多くのサービスが提供されるようになった。しかしながら、クラウドによる処理は、ユーザ側から離れた場所にある大規模なデータセンターとの通信が必要となるため、データセンターとの通信による遅延が生じ、リアルタイム性が求められる処理には不向きである。そのため、リアルタイム性の高いデータ処理を行うため、ユーザにより近いエッジに配置可能な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 ( $\mu$ DC) の検討が進められている [1]。

$\mu$ DC では、クラウドデータセンターとは異なり、資源が限られており、多種多量のアプリケーションを実行するためには、限られた資源を有効利用することが求められる。そこで、CPU・GPU・メモリ・ストレージといった資源を分離してネットワークに接続し、各アプリケーションに必要な分だけ資源を割り当てることで資源利用効率を上げることができる、リソース分離型マイクロデータセンター ( $\mu$ DDC) が提案されている [2]。 $\mu$ DDC は各タスクに必要な分だけ資源を割り当てていくことが可能であり、資源の柔軟な利用が可能である。さらに、 $\mu$ DDC では資源が独立しているため、資源ごとに技術の進化に合わせて資源を入れ替えることや柔軟なスケーリングも可能となる。

$\mu$ DDC では、資源間の情報のやり取りは  $\mu$ DDC 内のネットワークを経由して行われる。そのため、 $\mu$ DDC 内のネットワークが  $\mu$ DDC で実行されるアプリケーションの処理性能に大きな影響を与える [3]。特に、CPU とメモリ間の通信遅延の影響は大きい。そのため、 $\mu$ DDC では、通信遅延を考慮した資源割り当てを行うことが必要であり、我々も、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、 $\mu$ DDC のための資源割り当て制御手法を提案している [4]。

しかしながら、 $\mu$ DDC における柔軟な資源割り当てを実現するためには  $\mu$ DDC 内のネットワーク構造が重要であり、ネットワーク構造によっては特定のリンクに通信が集中せざるを得ず、低遅延で資源間を接続することが困難となることもあり得る。

文献 [5] においても、リソース分離型データセンターのネットワーク構成について検討がされており、複数の木構造を並列に構成することにより、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。しかしながら、提案したネットワーク構造と従来型の Leaf-Spine 型のネットワーク構造の比較が行われているのみであり、 $\mu$ DDC の性能に大きく影響を与える計算資源やメモリ資源の配置についての検討は行われていない。

そこで、本稿では、計算資源やメモリ資源といった資源の接続方法も含め、 $\mu$ DDC 内のネットワークに適したネットワーク構成について議論を行う。本稿では、まず、 $\mu$ DDC 内のネットワークに求められる要件について議論する。その後、その要件を満たしているのかを評価可能な指標を導入する。そして、既存のデータセンター向けのネットワーク構造に計算資源とメモリ資源を接続した  $\mu$ DDC 内ネットワーク構成に対して、導入した指標を適用することにより、ネットワークの特性を調べ、

$\mu$ DDC に適したネットワーク構成について議論する。また、各ネットワークにおける資源をアプリケーションの要求に対して割り当てるシミュレーションを行い、ネットワークの特性と収容可能なアプリケーションの数の関係についても確認する。

HPC 環境やデータセンターで一般的に用いられている 3 つのトポロジを  $\mu$ DDC 内のネットワーク構成に用いた場合のネットワーク特性やアプリケーション収容能力を評価することで、 $\mu$ DDC に適したネットワーク構成について考察する。

本稿の構成は以下の通りである。2 章では  $\mu$ DDC 内のネットワーク構成に求められる要件について説明する。3 章で  $\mu$ DDC 内ネットワーク構成に対する指標を導入し、4 章ではネットワーク構成の特性と収容可能なアプリケーション数の関係について考察を行う。最後に、5 章でまとめと今後の課題について述べる。

## 2. $\mu$ DDC 内のネットワーク構成に求められる要件

$\mu$ DDC では、アプリケーションの要求に応じて、必要となる計算資源やメモリ資源と、その間のネットワーク資源を割り当てることにより、限られた資源で、より多くのアプリケーションを収容する。収容されたアプリケーションでは、割り当てられた資源を用いてアプリケーションを実行する。その際、計算資源やメモリ資源間では、メモリの読み書き等によって生じる多量の通信が発生する。また、処理によっては、メモリからのデータ転送が完了を待つ必要があるなど、ネットワークが  $\mu$ DDC の処理性能に与える影響は大きい。

そこで、本稿では、 $\mu$ DDC に適したネットワーク構成について議論する。まず、本節では、 $\mu$ DDC 内のネットワーク構成に求められる要件について整理する。

### 2.1 要件 1：計算資源・メモリ資源間の低遅延な接続

上述のように、 $\mu$ DDC においては、アプリケーション実行時には、計算資源とメモリ資源間の通信が  $\mu$ DDC 内のネットワーク経由で発生する。その際、計算資源が必要な情報をメモリ資源から取得するのに時間がかかる場合は、情報の取得を待つ時間が必要となり、アプリケーションの処理速度の低下の原因となり得る。そのため、通信が発生しうる計算資源とメモリ資源間については、必要な情報取得にかかる時間を短くすることが求められ、そのためには、それらの資源間を低遅延で接続するネットワーク構成が必要となる。

### 2.2 要件 2：多量の計算資源・メモリ資源間の通信の収容

$\mu$ DDC では、複数のアプリケーションを同時に実行する。その際、各アプリケーションにおいて、計算資源とメモリ資源間で多量の通信が発生する。 $\mu$ DDC 内のネットワークでは、このように複数のアプリケーションが同時に実行された際に多量の通信が発生した際であっても、それらを収容するだけの帯域が確保できることが求められる。

### 2.3 要件 3：低コスト

$\mu$ DDC のネットワーク構成として、上記の性能に関する要件に加えて、低コストに構築できるネットワークであることが求められる。

### 3. $\mu$ DDC 内ネットワーク構成に対する指標

本節では、 $\mu$ DDC 内に適したネットワーク構成を検討するにあたり、上述の要件に対応する指標を導入し、各指標をもとに、適切なネットワーク構成の探索を行う。

#### 3.1 指標1：各計算資源から $k$ ホップ以内のメモリ資源数

2.1 節で述べたように、 $\mu$ DDC において、処理性能を確保するためには、各計算資源から低遅延でアクセス可能なメモリ資源が必要となる。この要件を確認するための指標として、本稿では、各計算資源から  $k$  ホップ以内のメモリ資源数という指標を導入する。本指標が大きいネットワーク構成では、各計算資源にとって、ホップ数が少なく低遅延で通信可能となるメモリ資源が多くなる。

#### 3.2 指標2：重み付きエッジ媒介中心性

2.2 節で述べたように、 $\mu$ DDC では、複数の同時に実行されるアプリケーションが存在した場合においても、それらが生じる通信のすべてに対して十分な帯域を確保して収容することが求められる。そのため、通信が発生しうる計算資源・メモリ資源間の通信が、特定のリンクに集中するということは避ける必要がある。

このような特定のリンクへ通信が発生しやすいかを調べる手法としては、媒介中心性が挙げられる。ただし、 $\mu$ DDC における資源の割り当てを考慮すると、各計算資源の通信相手となるメモリ資源は、計算資源の近くのメモリ資源となるような資源割り当てが行われることが自然である。そのため、すべての資源間で通信が発生するのではなく、距離の近い計算資源とメモリ資源間のみ通信が発生する場合を想定し、ボトルネックとなるリンクの有無について調べることが必要となる。

本稿では、上記を満たす指標として、以下で定義する重み付きエッジ媒介中心性を用いる。

$$c(e, k) = \sum_{s, t \in V} \left( \frac{\sigma(s, t | e)}{\sigma(s, t)} \cdot w_{s, t}(k) \right) \quad (1)$$

ここで、 $c(e, k)$  は 2 点  $s, t$  間の最短経路における辺  $e$  を含む経路の割合に重みを掛けた結果の総和であり、 $\sigma(s, t)$  は  $s, t$  間の最短経路数、 $\sigma(s, t | e)$  は辺  $e$  を通る  $s, t$  間の最短経路数を示す。また、 $w_{s, t}(k)$  は 2 点  $s, t$  の資源ペア間の通信の有無を表す値である。本稿では、以下のように、この値を資源ペアの順位  $k$  によって決まる値として、以下のように定義した。

$$w_{s, t}(k) = \begin{cases} w'_{type(s)type(t)} & \text{if } rank_s(t) \leq k; \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

ここで、 $rank_s(t)$  は資源ペアの相対的な近さの順位、すなわち  $s$  にとって  $t$  が何番目に近い  $type(t)$  のノードであるかを示す。 $w'_{type(s)type(t)}$  は  $type(s)$  と  $type(t)$  の資源間の通信の有無を示し、 $type(s)$  と  $type(t)$  の間に通信が生じうる場合には 1、生じえない場合には 0 とする。

上記定義より、 $c(e, k)$  は、各資源が通信が発生しうる種類の資源のうち、 $k$  番目に近い資源とのみ通信した際に、リンク  $e$  が使われる確率に相当する。 $c(e, k)$  が大きいリンクの有無によ

り、距離の近い計算資源とメモリ資源間のみ通信が発生する場合に、ボトルネックとなるリンクが存在するかを確認することができる。

#### 3.3 指標3：スイッチコスト・リンクコスト

本稿では、ネットワークにかかるコストを調べるにあたり、計算資源とメモリ資源1つあたりのスイッチ数（スイッチコスト）、リンク数（リンクコスト）をコストとして定義し、ネットワーク構築にかかるコストの比較を行う。

### 4. ネットワーク構造の特性と収容可能なアプリケーション数の関係

本節では、既存のデータセンター向けのネットワーク構造を用い、計算資源とメモリ資源を接続した  $\mu$ DDC 内ネットワーク構成を構築し、その特性を調べるとともに、文献[4]の手法を用い、アプリケーションの要求に対して資源を割り当てを行った場合について評価し、適切なネットワーク構成について議論を行う。

#### 4.1 評価対象ネットワーク

$\mu$ DDC 内のネットワークの構成は、スイッチで構成されるネットワークの構造のみならず、計算資源とメモリ資源が接続するスイッチの選択も大きな影響を与える。

本稿では、スイッチ間のネットワークの構成としては、HPC 環境やデータセンターで一般的に用いられている、Torus、Flattened butterfly [6]、Fat-tree [7] の 3 種類の構成を用いた。また、計算資源とメモリ資源の接続の方法としては、計算資源が接続されているスイッチと隣接するスイッチはメモリ資源が接続されるような構成（交互に配置）と、計算資源が接続されているスイッチ同士が相互に接続、メモリ資源が接続されているスイッチ同士が相互に接続されるような構成（固めて配置）の 2 種類の構成について調査する。交互に配置することにより、いずれの計算機資源においても、自身に近いメモリ資源が存在する一方で、接続先のメモリ資源によってはホップ数が非常に長くなる。

図1に比較したネットワーク構造を示す。本稿では、いずれのネットワークにおいても、CPU 資源数を 100、メモリ資源数を 100 として評価を行った。また、各スイッチは、マルチコアファイバ 10 本が接続可能なパケットスイッチであるとし、各マルチコアファイバは 4 本のファイバーコアを持つものとした。各ファイバーコアは資源割り当ての際には独立したリンクとみなすことができるものとした。

#### 4.2 ネットワーク構造の特性

図2に、各ネットワーク構成における各計算資源から  $k$  ホップ以内のメモリ資源数の平均値のグラフを示す。図より、交互に配置した場合の方が、固めて配置した場合よりも、多くのメモリ資源に短いホップ数でアクセス可能であることが分かる。また、スイッチ間ネットワーク同士を比較すると、Flattened butterfly では、 $k=3$  で、多くのメモリ資源にアクセスができるのに対し、3D Torus や Fat-tree では、 $k=3$  の場合では、それぞれ、40、10 程度のメモリ資源にしかアクセスすることができないことが分かる。そのため、低遅延にアクセス可能なメモリ資源の候補の数という観点では、Flattened butterfly 構造において

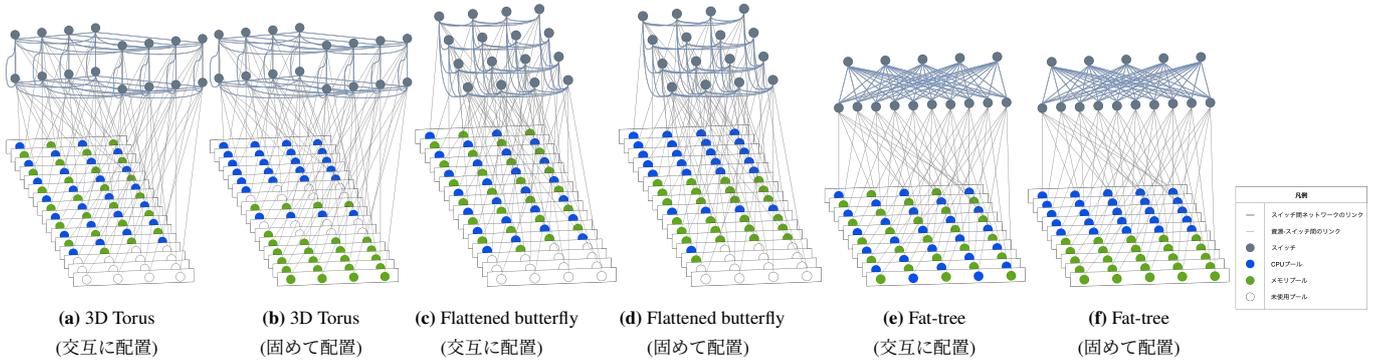


図 1: 比較したネットワーク構造

計算資源とメモリ資源を交互に配置した場合が優れていると言える。

図 3 に、各ネットワーク構成における重み付きエッジ媒介中心性の最大値を示す。図より、交互に配置した場合の方が、固めて配置した場合よりも、重み付きエッジ媒介中心性が大きくなっていることが分かる。これは、交互に計算資源とメモリ資源を配置することにより、計算資源とメモリ資源間の通信が短いホップ数の経路に収容されるようになり、各リンクを経由する通信量を抑えることができるためである。また、スイッチ間ネットワーク同士を比較すると、Fat-tree 構成は、 $k$  が大きい場合においても、重み付きエッジ媒介中心性を低く抑えることができているものの、 $k$  が小さく、通信が発生する計算資源・メモリ資源の組み合わせが近隣のものに限定されている場合は、3D Torus、Flattened butterfly と比べて、重み付きエッジ媒介中心性が大きくなっている。これは、階層的にスイッチを配置する必要のある Fat-tree では、各ノード間に多数の経路が確保されているものの、ホップ数の短いノードペアが限られていることが原因である。それに対して、Torus や Flattened butterfly では、計算資源とメモリ資源を交互に配置した場合には、 $k$  が小さいと、重み付きエッジ媒介中心性も非常に小さな値となる。特に、上述の様に、Flattened butterfly では、各計算資源に近いメモリ資源の数も多いため、 $k$  が 50 までであれば、重み付きエッジ媒介中心性を低く維持することが可能である。このため、計算資源とメモリ資源の多量の通信を収容するという観点においても、Flattened butterfly 構造において、計算資源とメモリ資源を交互に配置した場合が優れていると言える。

最後に、表 1 に、各ネットワーク構成におけるスイッチコストとリンクコストの値を示す。表より、全スイッチが計算資源やメモリ資源と接続する 3D Torus や Flattened butterfly と比べ、Fat-tree はスイッチコスト、リンクコストともに高くなる。以上より、コスト面では、3D Torus や Flattened butterfly が優れていると言える。

表 1: リンクコスト・スイッチコスト

	3D Torus	Flattened butterfly	Fat-tree
リンクコスト	0.4375	0.4375	0.5000
スイッチコスト	0.0625	0.0625	0.0800

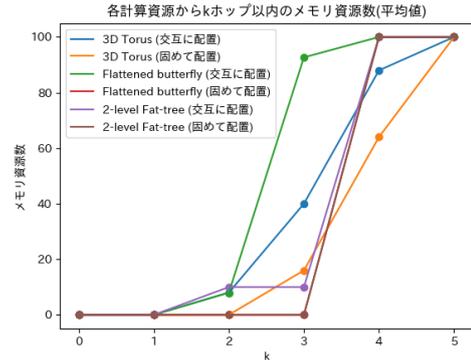


図 2: 各計算資源から  $k$  ホップ以内のメモリ資源数

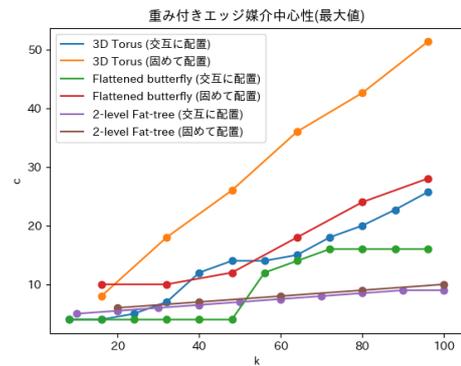


図 3: 重み付きエッジ媒介中心性

### 4.3 資源割り当てシミュレーションによる評価

本節では、上述の特性をもとにした、 $\mu$ DDC に適したネットワーク構造に関する考察と、実際に同時に収容可能なアプリケーション数の関係についても調査する。本調査においては、評価対象の 6 つのネットワーク構造について、文献 [4] の資源割り当て手法を適用する。文献 [4] の手法では、アプリケーションが必要とする計算資源とメモリ資源の量、アプリケーション実行にかかる計算処理とメモリ資源へのアクセス量に加えて、目標処理時間が与えられる。アプリケーションの実行開始の要求の要求は随時到着し、目標処理時間内に処理が完了することが見込まれる場合には、資源の割り当てを行い、目標処理時間内の処理が困難な場合は、アプリケーションの開始要求を棄却する。本節では、アプリケーションの開始要求を生成し、棄却

表 2: 評価用ネットワークのパラメータ設定

パラメータ	値
CPU の FLOPS	13.619GFLOPS
伝播遅延	0.025 $\mu$ s
スイッチの処理遅延	3 $\mu$ s
カットスルー時のスイッチ処理遅延	300ns
ページサイズ	4KB
帯域幅	50Gbps

表 3: 資源要求のパラメータ設定

	要求 1	要求 2	要求 3
目標時間	250ms	200ms	150ms
必要計算資源量	3	3	3
必要メモリ資源量	3	3	3
プロセス 1			
クロック数	0.035	0.035	0.035
1ms あたりのメモリ書き込み時のパケット到着率	0.00033	0.002	0.005
1ms あたりのメモリ読み込み時のパケット到着率	0.00033	0.002	0.005
プロセス 2			
クロック数	0.054	0.054	0.054
1ms あたりのメモリ書き込み時のパケット到着率	0	0	0
1ms あたりのメモリ読み込み時のパケット到着率	0.00033	0.002	0.005
プロセス 3			
クロック数	2371.33	1960.36	1960.36
1ms あたりのメモリ書き込み時のパケット到着率	1.87	1.9	1.9
1ms あたりのメモリ読み込み時のパケット到着率	3.71	3.43	3.43
ページフォールトの発生回数	67543.25	56661.29	56661.29
ページフォールトあたりに通信するページ数	5.27	4.84	4.84

された要求数を比較することにより、各ネットワーク構造で収容可能なアプリケーション数の比較を行う。

#### 4.3.1 評価環境

##### a) ネットワーク内の資源

本評価では、評価対象のいずれのネットワークにおいても、計算資源やスイッチの処理能力は、表 2 に示す値を用いる。

##### b) 資源要求

本評価では、要求される処理速度の異なる 3 種類のアプリケーションの開始要求が到着するものとした。表 3 に各要求の設定を示す。この環境において、各要求が 90 分で解放されるとして、300 分間のシミュレーションを行った。なお、要求が到達する頻度は、300 分間中に 170 個の要求が到達するとすし、要求到着時には、その要求が要求 1、要求 2、要求 3 である確率をそれぞれ 0.35、0.30、0.35 と設定した。

##### c) 資源割り当て手法

文献 [4] の手法では、計算資源、メモリ資源、ネットワーク上のリンクに対してコストを割り当てる。そして、アプリケーションの実行にかかる時間を、計算資源がネットワーク経由でメモリ資源から処理に必要な情報を取得する時間と、計算資源内での処理時間の和として求め、アプリケーションの処理に必要な時間が、要求された時間内に終わるような資源割り当てのうち、コストを最小化するような資源割り当てを求める。本手法では、計算資源、メモリ資源ともに、高速・大容量な資源ほどコストが高く、また、高速・大容量な計算資源・メモリ資源間を接続する際に必要となるリンクほどコストが高くなるよう

表 4: 資源割り当て手法のパラメータ設定

パラメータ	値
資源探索を行うエージェント数	20
経路探索を行うエージェント数	20
探索の繰り返し回数	20
フェロモン減衰率	0.1
フェロモン強化率	100
フェロモンの重み	2
コストの重み	1
フェロモンの初期値	1000

なコストの割り当てを行うことで、将来のアプリケーション要求の収容において、必須となる可能性の高い資源に高いコストを割り当てる。その結果、将来のアプリケーション要求への対応に必要な資源を温存しつつ、要求された時間内に処理を行うことができるような資源割り当てを実行する。

ただし、コストが最小となる資源割り当てを探索するには、膨大な資源割り当て候補の探索が必要となる。そのため、文献 [4] では、条件を満たしつつコスト最小な資源割り当ての探索にあたり、ACO (アントコロニー最適化) を用いている。この手法では、以下の動作を行うエージェントを複数生成することにより、資源割り当ての探索を行う。本稿では、表 4 に示したパラメータを設定し、資源割り当ての探索を実行した。

#### 4.3.2 結果

図 4 に、各要求列と必要資源量の組み合わせにおける棄却数のグラフを示す。図より、計算資源・メモリ資源を固めて配置した場合の方が、交互に配置した場合よりも、アプリケーション開始要求の棄却数が多いことが分かる。これは、上述のネットワークの特性の分析において、計算資源・メモリ資源を固めて配置した場合は、計算資源から  $k$  ホップ以内のメモリ資源数、重み付き媒介中心性のいずれも悪化していることと合致する結果であり、計算資源・メモリ資源を固めて配置することにより、計算資源・メモリ資源間のホップ数が多くなり、遅延の増大のみならず、未使用のリンクの数も減り、十分な低遅延でのアプリケーションの実行が可能な資源割り当てができなくなるためである。

また、計算資源・メモリ資源を交互に配置した 3 つのネットワーク構造を比較すると、3D Torus、Fat-tree においては、棄却が発生しているものの、Flattened butterfly においては棄却が発生していない。これも、上述のネットワーク特性分析で示されている、Flattened butterfly においては、各計算資源から短いホップ数で接続可能なメモリ資源の数が多く、重み付き媒介中心性も低いという結果と合致している。すなわち、計算資源・メモリ資源を交互に配置した場合、Flattened butterfly 構造では、各計算資源において、十分な帯域・低遅延で接続可能なメモリ資源の候補が多く、柔軟な資源配置が可能であり、多数のアプリケーション要求が到達した際にも、要求処理時間を満たす割り当てが可能であったと考えられる。

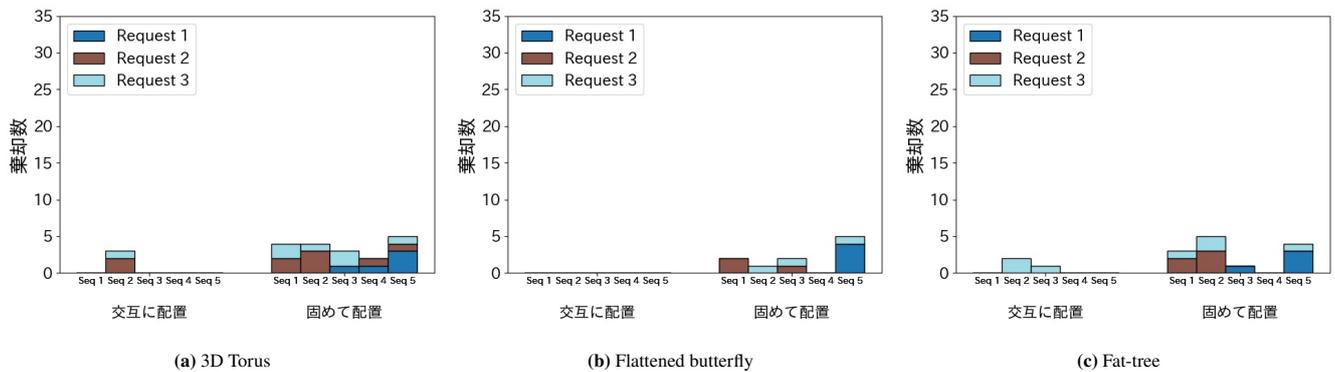


図 4: 要求列と必要資源量ごとの各ネットワーク構成の棄却数

## 5. まとめと今後の課題

本稿では、 $\mu$ DDC 内のネットワーク構成について、要件と、その要件に関連するネットワーク構成の特性を表す指標について議論をした。また、従来、HPC 環境やデータセンターで用いられているネットワーク構造をもとにした  $\mu$ DDC 内のネットワーク構成について、検討したネットワーク特性について評価するとともに、当該ネットワークに対する資源割り当てのシミュレーションを行い、検討した特性を持つネットワークが、より多くのアプリケーション要求を収容可能であることを示した。

本稿では、既存のネットワーク構成をもとに、 $\mu$ DDC 内のネットワーク構成について議論を行っているが、 $\mu$ DDC で求められる要件は、従来型の全サーバ間が通信を行うことを前提とした、データセンターとは大きく異なるため、より低コストで多数の要求を収容可能な新たなネットワーク構造が存在することも考えられる。そのため、今後、本稿で検討を行った指標をもとに、より  $\mu$ DDC 内のネットワークに適した構造について、検討を進める予定である。

## 謝 辞

本研究の一部は NICT 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」によるものである。ここに記して謝意を示す。

## 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, 2018.
- [2] S. Han, N. Egi, A. Panda, S. Ratnasamy, G. Shi, and S. Shenker, "Network support for resource disaggregation in next-generation datacenters," *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks*, pp.1–7, 2013.
- [3] A. Ikoma, Y. Ohsita, and M. Murata, "Impact of remote memory and network performance on execution performance of disaggregated micro data centers," *Proceedings of ICETC 2021*, pp.●●–●●, Dec. 2021.
- [4] 生駒昭繁, 大下裕一, 村田正幸, "リソース分離型マイクロデータセンターにおける将来の資源要求に備えた資源割り当て手法," *電子情報通信学会技術研究報告 (IN2021-36)*, pp.31–36, March 2022.
- [5] H. Yuan, A. Saljoghei, A. Peters, and G. Zervas, "Disaggregated optical data center in a box network using parallel ocs topologies," *Optical Fiber Communication Conference Optical Society of America*, pp.W1C–2 2018.

- [6] J. Kim, W.J. Dally, and D. Abts, "Flattened butterfly: a cost-efficient topology for high-radix networks," *Proceedings of the 34th annual international symposium on Computer architecture*, pp.126–137, 2007.
- [7] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," *ACM SIGCOMM computer communication review*, vol.38, no.4, pp.63–74, 2008.