

障害発生を考慮した仮想マシン配置によるエッジコンピューティング設計

平田 孝志*, 三浦 秀芳 (関西大学), 木村 共孝 (同志社大学)

Design of edge computing systems by virtual machine allocation considering failure occurrences
Kouji Hirata*, Hideyoshi Miura (Kansai University), Tomotaka Kimura (Doshisha University)

Abstract

This paper proposes a virtual machine allocation method considering failure occurrences in edge computing environments. The proposed method uses multiple routing configurations (MRC) algorithm to recovery a single node/link failure. The MRC algorithm prepares multiple backup routing configurations in advance, and realizes fast failure recovery by switching to an appropriate configuration when a failure occurs. The proposed method allocates virtual machines to edge servers so as to minimize network and edge server loads on the routing configurations. To derive the optimal solution, the proposed method provides Mixed Integer Programming. In this paper, we show the effectiveness of the proposed method through numerical experiments.

キーワード：エッジコンピューティング, 障害復旧, 複数ルーティング構成, 仮想マシン割当
(edge computing, failure recovery, multiple routing configurations, virtual machine allocation)

1. はじめに

近年, モノのインターネット (Internet of Things: IoT) 技術が急速に発展しており, 多くの IoT サービスがユーザに提供され, 今後そのデータ量は増加していくと予想される⁽¹⁾. 一般に IoT サービスでは, IoT デバイスから収集されたデータを, クラウドデータセンタにおいて処理することでユーザにサービスを提供することが想定される⁽²⁾. しかし, 将来的にますます増加する大量のデータを, ネットワークを通じてクラウドデータセンタに送信することは, ネットワーク内の輻輳の発生や, クラウドデータセンタへの負荷の集中の要因となる. そこで, このような問題に対処するために, エッジコンピューティング環境が検討されている⁽³⁾.

エッジコンピューティングではクラウドデータセンタに加えて, ユーザや IoT デバイスに近いネットワークエッジ (ネットワークの端) にデータを処理するエッジサーバを配置する. IoT デバイスが生み出すデータをエッジサーバとクラウドデータセンタによる処理を連携させることで, より良いサービスをユーザに提供することができる. 各エッジサーバ上では, ユーザが要求するサービスに応じた複数のアプリケーションが仮想マシン (Virtual Machine: VM) 上で動作し, 必要に応じてデータの処理がなされる⁽⁴⁾. これにより, ネットワーク内の輻輳発生やクラウドデータセンタにかかる負荷集中の回避が期待できる.

エッジコンピューティングシステムを設計する際には, それぞれのサービスを提供する異なる種類の VM を, どのエッジサーバに割当ててのかを決定する問題である, エッジサーバへの VM 割当て問題⁽⁵⁾ を考慮することが重要である. エッジサーバの容量には限りがあるため, 異なるサービスを提供する全ての VM を同一のエッジサーバに割当てることはできない. そのため, 容量制約やユーザ-VM 間の

トラフィック量等を考慮した割当てが必要となる.

本稿ではエッジコンピューティング環境において, 障害発生を考慮した VM 割当て最適化手法を提案する. 提案手法では, 高速障害復旧技術の一つである複数ルーティング構成 (Multiple Routing Configurations: MRC) 技術⁽⁶⁾ の使用を想定する. MRC では, あらかじめ複数の予備ルーティング構成を用意する. 通常ルーティング構成におけるデータ転送中に単一ノードもしくはリンク障害が発生した場合, MRC は障害箇所を使用しない予備ルーティング構成の一つを使いデータ伝送を継続することで, 高速な障害復旧が実現できる. しかし, MRC では構成毎に使用可能なノードやリンクが異なり, その結果, 予備ルーティング構成において特定のリンクやエッジサーバに負荷が集中する恐れがある. そこで提案手法では, 各ルーティング構成においてリンク負荷やエッジサーバ負荷を分散するための VM 割当てを, 混合整数計画 (Mixed Integer Programming: MIP) 問題として定式化する. この MIP を解くことで, MRC によって構成される各ルーティング構成の負荷を分散する VM 割当てを実現する. 本稿では数値実験により提案手法の有効性を示す.

2. 複数ルーティング構成

MRC は K ($K > 1$) 個の予備ルーティング構成を予め用意することで高速障害復旧を実現する. 単一ノードもしくはリンク障害が発生した場合でも, MRC は障害発生箇所を使用しない予備ルーティング構成を用いることで, データ転送を継続可能である. K 個の予備ルーティング構成において, 各ノードは通常ノードと分離ノードのいずれかに分類される. また, 各リンクは通常リンク, 分離リンク, 制限リンクのいずれかに分類される. 通常ノード/リンクは障害発生時にデータ転送に直接使用されるノード/リンクを表

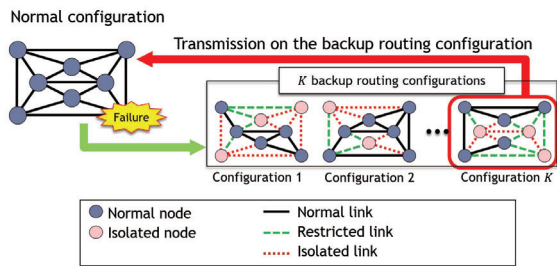


図1 複数ルーティング構成
 Fig. 1. Multiple routing configurations.

す. 分離ノード/リンクは, 障害発生時において使用しないことが想定されるノード/リンクを表す. 制限リンクはデータ転送の最初もしくは最後のホップにおいてのみ使用されるリンクを示す. これは, 制限リンクに接続されている分離ノードにおいて障害が発生していない場合を想定して使用されるリンクである.

図1に, 通常ルーティング構成とそれから作成される K 個の予備ルーティング構成の例を示す. MRC では, 通常ルーティング構成と予備ルーティング構成は以下の制約に従い構成される.

- (1) 各ノードと各リンクは少なくとも一つ以上の予備ルーティング構成において, 分離ノード, 分離リンクになる必要がある.
- (2) 各予備ルーティング構成において, 全ての通常ノードと, 通常リンクで構成される連結グラフが存在する.
- (3) 制限リンク, もしくは分離リンクは分離ノードに接続され, 通常リンクは分離ノードに接続されない.
- (4) 分離ノードに接続するリンクのうち少なくとも一つは制限リンクである.
- (5) 制限リンクにより接続される二つのノードは, 一つが通常ノードで, 他方が分離ノードである.
- (6) 分離リンクにより接続される二つのノードのうち一つは分離ノードである.

予備ルーティング構成の一つを用いることで, MRC は高い可用性を維持することができる. 具体的には, 単一ノード障害発生時において, MRC はそのノードが分離ノードである予備ルーティング構成を選択する. 同様に, 単一リンク障害が発生した場合においては, MRC はそのリンクが分離ノードとして選択されている予備ルーティング構成を使用する. なお, 選択された予備ルーティング構成において障害が発生していないノードもしくはリンクが, 分離ノードや分離リンクとして扱われている場合がある. 例えば, 図1において, 通常ルーティング構成において単一リンク障害が発生し, 予備ルーティング構成1が使用された場合, 障害が発生していない二つの分離ノードと七つの分離リンクが存在することとなる. これらの障害が発生していない分離ノードは制限リンクを利用してデータの送受信

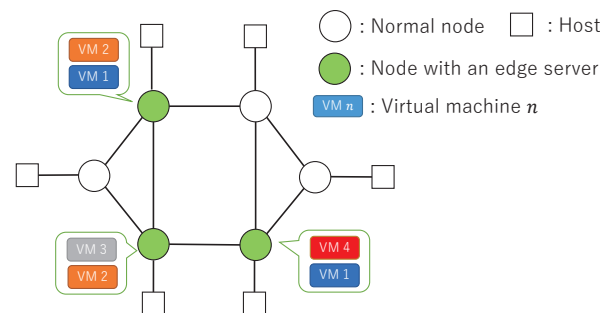


図2 システムモデル
 Fig. 2. System model.

が可能となる.

3. 障害発生を考慮した仮想マシン配置

〈3・1〉 システムモデル 図2に本稿で想定するシステムモデルを, 表1に使用する記号とその意味を示す. 各エッジサーバ $j \in \mathcal{V}_S$ では, ユーザが要求するそれぞれのサービスに対応する VM が割当てられ稼働する. 本稿では簡単化のため, 各 VM のサイズは同一とし, 最大 C_j 個の VM を配置できるものとする. 同じサービスを提供する VM を複数のエッジサーバに複製することで, ネットワークリンクとエッジサーバにかかる負荷の分散が可能となる. 各ホストは必要なサービスに応じて, いずれかのエッジサーバにある, 対応する VM と通信を行う. また, 各ホスト $i \in \mathcal{V}_H$ と各 VM $n \in \mathcal{N}_i$ 間の通信要求量 $\lambda_{i,n}$ は予め与えられており, 既知であるとする. また, MRC による予備ルーティングの形成は予め行っているものとし, 各構成において使用する経路は予め計算されたホップ数に関する最短経路を用いるものとする. 以下ではネットワークリンクの負荷をそのリンクを通るホスト-VM 間通信の要求量の総和とし, また, エッジサーバ負荷をそのエッジサーバを使用するホストの通信要求量の総和とする.

このシステムモデルを想定して, 提案手法では MRC を適用した場合における VM の最適配置を検討する. MRC を適用した場合, エッジサーバがあるノードが特定の予備ルーティング構成において分離ノードとして選択され, 通信経路が制限される. つまり, 分離ノードは制限リンクを通じてのみ, 他のノードと通信を行うことが可能であり, 制限リンクにトラフィックが集中する恐れがある. また, 各予備ルーティング構成においては, 分離リンクを通信に使用することができないために使用できる経路が限られ, その結果, 輻輳が発生しやすい状況が生まれる. どのリンクが制限リンクや分離リンクとなるかは, 予備ルーティング構成毎に異なるため, 各予備ルーティング構成間において, これらトラフィック集中等の問題が, 特定の予備ルーティング構成において大きく出ることが考えられる. そこで提案手法では VM の適切なエッジサーバへの配置により, こ

表 1 記号リスト
Table 1. List of symbols.

Symbol	Meaning
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Network consisting of the set \mathcal{V} of nodes and the set \mathcal{E} of links
$\mathcal{V}_H \subseteq \mathcal{V}$	Set of hosts (users)
$\mathcal{V}_S \subseteq \mathcal{V}$	Set of nodes with edge servers
\mathcal{N}	Set of distinct virtual machines
$\mathcal{N}_i \subseteq \mathcal{N}$	Set of distinct virtual machines required by host $i \in \mathcal{V}_H$
\mathcal{R}	Set of routing configurations including a normal routing configuration and backup routing configurations
$d_{i,j}^{[r]}$	Number of hops from node $i \in \mathcal{V}$ to node $j \in \mathcal{V}$ along the shortest path on routing configuration $r \in \mathcal{R}$
$\lambda_{i,n}$	Amount of traffic transmitted between host $i \in \mathcal{V}_H$ and virtual machine $n \in \mathcal{N}_i$
C_j	Capacity of edge server $j \in \mathcal{V}_S$
$z_{i,j,n}^{[r]}$	Binary variable that is equal to 1 if host $i \in \mathcal{V}_H$ communicates with virtual machine $n \in \mathcal{N}_i$ in an edge server $j \in \mathcal{V}_S$ on routing configuration $r \in \mathcal{R}$; otherwise, 0
$\delta_{n,j}$	Binary variable that is equal to 1 if virtual machine $n \in \mathcal{N}$ is allocated to edge server $j \in \mathcal{V}_S$; otherwise, 0
α	Non-negative real variable that indicates the maximum network load
β	Non-negative real variable that indicates the maximum edge server load
W_1, W_2	Weight parameters
A_r, B_r	Weight parameters for routing configuration $r \in \mathcal{R}$

の問題に対処する。さらに、ホストが各構成においてどの VM と通信するのかを同時に考えることで、ネットワーク負荷の偏りだけでなく、エッジサーバへの通信負荷の偏りについても改善することを狙う。

〈3・2〉 定式化 提案手法では、以下に示す目的関数及び制約式で与えられる MIP を解くことで、MRC における各ルーティング構成間のネットワーク負荷およびエッジサーバ負荷の偏りをなくすることができる VM 割当て及び VM 選択を行う。

Objective function

$$\min W_1\alpha + W_2\beta \quad (1)$$

Subject to

$$\forall r \in \mathcal{R}; \quad \alpha \geq \sum_{i \in \mathcal{V}_H} \sum_{j \in \mathcal{V}_S} \sum_{n \in \mathcal{N}_i} A_r \lambda_{i,n} d_{i,j}^{[r]} z_{i,j,n}^{[r]} \quad (2)$$

$$\forall j \in \mathcal{V}_S, r \in \mathcal{R}; \quad \beta \geq \sum_{i \in \mathcal{V}_H} \sum_{n \in \mathcal{N}_i} B_r \lambda_{i,n} z_{i,j,n}^{[r]} \quad (3)$$

$$\forall j \in \mathcal{V}_S; \quad \sum_{n \in \mathcal{N}} \delta_{n,j} \leq C_j, \quad (4)$$

$$\forall i \in \mathcal{V}_H, n \in \mathcal{N}_i, r \in \mathcal{R}; \quad \sum_{j \in \mathcal{V}_S} z_{i,j,n}^{[r]} = 1, \quad (5)$$

$$\forall n \in \mathcal{N}; \quad \sum_{j \in \mathcal{V}_S} \delta_{n,j} \geq 2, \quad (6)$$

$$\forall i \in \mathcal{V}_H, j \in \mathcal{V}_S, n \in \mathcal{N}_i, r \in \mathcal{R}; \quad \delta_{n,j} \geq z_{i,j,n}^{[r]}. \quad (7)$$

式 (1) は目的関数であり、最大ネットワーク負荷及び最大エッジサーバ負荷の重み付き和の最小化を行う。重みパラメータ W_1 と W_2 を調整することによりネットワーク負荷の平滑化、もしくは、エッジサーバ負荷の平滑化のどちらを優先するのかを決定することが可能となる。式 (2) は各ルーティング構成のネットワーク負荷、つまり各リンクにかかる負荷（そのリンクを通る通信の要求量）の総和が最大 α で抑えられることを示す。ただし、ここでは構成毎にパラメータ A_r で重み付けを行った値を用いる。式 (3) は、各ルーティング構成において、各エッジサーバにかかる負荷（そのエッジサーバへの通信要求量の総和）が最大で β 以下になることを保証するものである。ここでも同様に、パラメータ B_r で重み付けを行った値を用いる。式 (4) は各エッジサーバ $j \in \mathcal{V}_S$ に配置できる VM の数が容量 C_j 以下であることを示す。式 (5) は各ユーザは各ルーティング構成において、それぞれの VM に対して複製配置されているものの中から一つを選択して通信することを意味する。式 (6) は、各 VM が少なくとも 2 つ以上のエッジサーバに割当てられていることを示す。これは、単一ノード障害が発生した場合に、ネットワーク内で少なくとも 1 つ以上の VM がそれぞれ存在することを保証する。式 (7) は、ホストは対応する VM が割当てられているエッジサーバのみと通信できることを示す。この MIP を解くことで、VM をどのエッジサーバに割当ててるのか ($\delta_{n,j}$) と、各ルーティング構成においてホストが割当てられたどの VM と通信するのか ($z_{i,j,n}^{[r]}$) を決定する。

4. 評価

〈4・1〉 モデル 提案手法の有効性を評価するために数値実験を行う。使用するネットワークはノード数 $|\mathcal{V}| = 30$ で、Barabasi-Albert (BA) モデル⁽⁷⁾ に基づいて構築する。各ノードはホストと中間ノードの役割を果たすとし、全てのノードにエッジサーバを配置する ($\mathcal{V} = \mathcal{V}_H = \mathcal{V}_S$)。各ホスト i と各 VM n 間で発生する通信要求量 $\lambda_{i,n}$ は平均 5.0 [Mbps] として $[0, 10.0]$ からランダムに選択する。またこの時、リンクがすべてのトラフィックフローを収容できるように、各リンク容量は十分に大きいと仮定する。また、各エッジサーバ $j \in \mathcal{V}_S$ の容量 C_j は $C_j = C = 10$ とする。ネットワーク内に配置する異なる VM の数 $|\mathcal{N}|$ は 30 とする。各ルーティング構成に対する重みパラメータ A_r 及び B_r の値は 1 とする。本稿では、提案手法の MIP で与えられる最適化問題を解くために IBM ILOG CPLEX⁽⁸⁾ を使

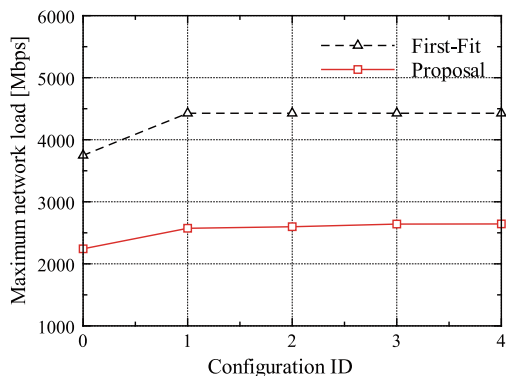


図3 最大ネットワーク負荷
Fig. 3. Maximum network load.

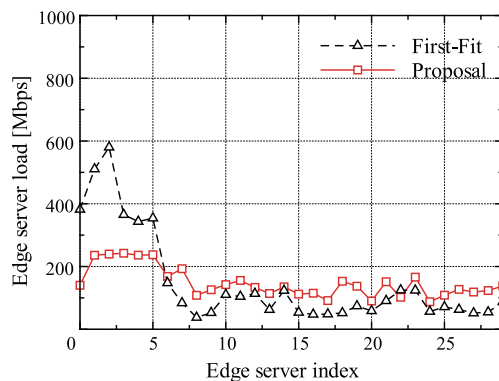


図5 エッジサーバ負荷
Fig. 5. Edge server load.

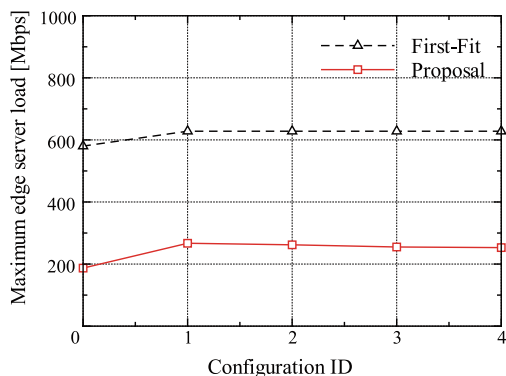


図4 最大エッジサーバ負荷
Fig. 4. Maximum edge server load.

用する。

〈4・2〉 結果 図3に、予備ルーティング構成数に対する最大ネットワーク負荷を示す。ここで、最大ネットワーク負荷とは、作成したルーティング構成（通常構成も含む）のうち、もっともネットワーク負荷が高かった構成のネットワーク負荷の値 (α) を表す。なお、横軸の0は通常構成のみの結果を表す。また、比較対象として、VM割当てをファーストフィット法を用いて行い、ホップ数に関して最も短いVMをユーザが選択する手法（First-Fit）による結果を併せて示す。図より、提案手法を用いることで大きくネットワーク負荷を削減できていることがわかる。

図4に、予備ルーティング構成数に対する最大エッジサーバ負荷 (β) を示す。図より、ネットワーク負荷と同様に、提案手法を用いることで大きくエッジサーバにかかる負荷を削減できていることがわかる。また、図5に、予備ルーティング構成数が4の場合の、通常構成における各エッジサーバの負荷を示す。図より、First-Fitでは、エッジサーバ負荷に偏りが大きいことがわかる。これは、次数が大きいエッジサーバが選択されやすくなるためである。一方、提案手法を用いることで、エッジサーバ負荷が平滑化されている。これにより、最大エッジサーバ負荷が抑制可能となる。

5. まとめ

本稿ではエッジコンピューティング環境において、障害発生を考慮したVM割当て最適化手法を提案した。本稿では数値実験により提案手法の有効性を示した。

謝辞

本研究は学術研究助成基金助成金基盤研究 (C) No. 18K1128 の支援を受けている。ここに記して謝意を表す。

参考文献

- (1) L. Atzori, A. Iera, and G. Morabito: "The Internet of Things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- (2) J. Gubbia, R. Buyyab, S. Marusica, and M. Palaniswamia: "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 27, pp. 1645–1660, 2013.
- (3) X. Sun and N. Ansari: "Edge IoT: Mobile edge computing for the Internet of Things," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 22–29, 2016.
- (4) Z. Tao, Q. Xia, Z. Hao, C. Li, L. Ma, S. Yi, and Q. Li: "A survey of virtual machine management in edge computing," *Proceeding of the IEEE*, vol. 107, no. 8, pp. 1482–1499, 2019.
- (5) X. Xu, X. Liu, Z. Xu, F. Dai, X. Zhang, and L. Qi: "Trust-oriented IoT service placement for smart cities in edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4084–4091, 2020.
- (6) A. Kvalbein, A. Hansen, T. Cicic, S. Gjessing, and O. Lysne, "Multiple routing configurations for fast IP network recovery," *IEEE/ACM Transactions on Networking*, vol. 17, no. 2, pp. 473–486, 2009.
- (7) A. Barabasi and R. Albert: "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.
- (8) IBM ILOG CPLEX
<https://www.ibm.com/products/ilog-cplex-optimization-studio>