

## 特別研究報告

題目

生物の縮退特性によるロバスト性を利用した  
データセンタの低消費電力化のための仮想マシン配置法

指導教員

松岡 茂登 教授

報告者

添 亮太

平成 26 年 2 月 14 日

大阪大学 基礎工学部 情報科学科

生物の縮退特性によるロバスト性を利用した  
データセンタの低消費電力化のための仮想マシン配置法

添 亮太

## 内容梗概

近年、クラウドコンピューティングやウェブアプリケーションなどの、ネットワークを介して提供されるサービスが大規模化かつ多様化し、データセンタ内で処理されるデータ量が増加の一途をたどっている。データ量の増加に伴い、データセンタ内のサーバ数も急速に増加しており、大規模なデータセンタにおいては、数万から数十万台のサーバが収容されるようになってきている。サーバ数が増大することで、サーバそのものの消費電力だけでなく、ネットワーク機器、及び空調機器の消費電力も増加するため、データセンタ全体の消費電力の増大が深刻な問題となっている。従って、データセンタの低消費電力化は、早急に取り組まなければならない課題である。

一方で、近年のクラウドサービスやウェブアプリケーションには、高いサービス品質が求められている。つまり、データセンタは、障害や環境変動、システムの再構成などの変動に対して、高い信頼性及び頑強性を要求されている。データセンタの信頼性及び頑強性を向上させるには、予備系機器の配置が不可欠となる。予備系機器の増加は信頼性及び頑強性の向上につながるが、同時に消費電力の増大にもつながる。そのため、データセンタの信頼性及び頑強性を維持しつつ、消費電力の低減を図ることが求められる。本報告においては、データセンタにおけるサーバへの仮想マシンの配置に着目し、信頼性及び頑強性を損なうことなく、データセンタの低消費電力化を実現するような、主系及び予備系のサーバ群に対する仮想マシン配置問題を取り扱う。

一般に、システムの冗長化を図るには、1つの構成要素に対して専用の予備機器を用意する1対1の冗長化を行うことが考えられるが、実現のコストが高いため、特に大規模なデータセンタには向いていない。一方、複数の構成要素が1つの予備機器を共有することでコストは下がるが、共有関係の構築方法によっては、信頼性が下がる可能性がある。しかしながら、1つの予備機器を共有する冗長化手法を用いて、システムの構成機器の故障だけでなく、環境変動や、システムの再構成に対しても、システムの頑強性を維持しつつ、消費電力低減を達成する手法は確立されていない。

そこで、本報告においては、古来様々な環境変動や障害に対して耐えてきた生物の仕組みからヒントを得ることによって、上記の問題を解決する。つまり、生物ネットワークのトポロジに見られる特徴である縮退特性を応用した、データセンタの仮想マシン配置における頑強かつ低消費電力な冗長化手法を提案する。提案手法は、従来のデータセンタに見られる1対1や1対多の冗長化ではなく、データセンタ全体として必要な冗長資源を保持する多対多の冗長化を実現する。具体的には、縮退特性をもつ生物ネットワークに見られる、ネットワーク化バッファリングと呼ばれる性質を利用し、多対多の冗長化を少ない予備機器によって実現する。ネットワーク化バッファリングを用いることで、ロバスト性を保持しつつ、予備機器を削減できるため、提案手法はデータセンタの低消費電力化にもつながる。提案手法をシミュレーションによって評価し、単純な冗長化手法に比べて、同等の頑強性を維持しつつ、最大75%の冗長資源を削減できることを示した。

## 主な用語

データセンタ, 仮想マシン配置, 生物ネットワーク, 縮退特性, 頑強性, 消費電力低減

# 目次

<b>1</b>	<b>はじめに</b>	<b>6</b>
<b>2</b>	<b>生物ネットワークの縮退特性とデータセンタへの応用</b>	<b>9</b>
2.1	縮退特性 . . . . .	9
2.2	ネットワーク化バッファリング . . . . .	9
2.3	データセンタにおける仮想マシン配置への適用 . . . . .	11
<b>3</b>	<b>データセンタモデルと障害発生モデル</b>	<b>12</b>
3.1	データセンタモデル . . . . .	12
3.2	データセンタにおける障害発生モデル . . . . .	15
3.3	障害からの回復 . . . . .	15
<b>4</b>	<b>提案手法</b>	<b>16</b>
4.1	仮想マシン配置手法 . . . . .	16
4.2	環境変動や障害発生後の VM 実行切り替えアルゴリズム . . . . .	28
<b>5</b>	<b>性能評価</b>	<b>30</b>
5.1	評価環境 . . . . .	30
5.2	均一な環境における提案手法の評価 . . . . .	30
5.3	不均一な環境における提案手法の評価 . . . . .	41
<b>6</b>	<b>まとめと今後の課題</b>	<b>44</b>
	<b>謝辞</b>	<b>45</b>
	<b>参考文献</b>	<b>46</b>

## 目次

1	ネットワーク化バッファリングの概念図	10
2	データセンタモデル	13
3	既存手法 1	17
4	既存手法 2	19
5	規則的部分重複手法 1	21
6	規則的部分重複手法 2	23
7	無作為部分重複手法 1	25
8	無作為部分重複手法 2	27
9	障害復旧の際に用いる指標が回復性能に与える影響	33
10	均一な環境における物理サーバの故障台数と障害回復割合の関係	35
11	均一な小規模データセンタを想定した環境における冗長度を変化させた時の サーバ故障数と障害回復割合の関係	36
12	均一な中規模データセンタを想定した環境における冗長度を変化させた時の サーバ故障数と障害回復割合の関係	37
13	均一な環境のラック障害における物理サーバの故障台数と障害回復割合の関係	38
14	均一な環境における冗長度と障害回復割合の関係	40
15	不均一な環境のサーバ障害における物理サーバの故障台数と障害回復割合の 関係	42
16	不均一な環境における冗長度と障害回復割合の関係	43

## 表目次

1	均一な環境におけるパラメータ設定 . . . . .	31
2	不均一な環境におけるパラメータ設定 . . . . .	32

## 1 はじめに

近年、クラウドコンピューティングやウェブアプリケーションなどの、ネットワークを介して提供されるサービスが大規模化かつ多様化し、データセンタ内で処理されるデータ量が増加の一途をたどっている [1, 2]. データ量の増加に伴い、データセンタ内のサーバ数も急速に増加しており、大規模なデータセンタにおいては、数十万から百万台のサーバが収容されるようになってきている [3]. サーバ数が増大することで、サーバそのものの消費電力だけでなく、ネットワーク機器、及び空調機器の消費電力も増加するため、データセンタ全体の消費電力の増大が深刻な問題となっている。従って、データセンタの低消費電力化は、早急に取り組まなければならない課題であり、様々なアプローチでデータセンタの低消費電力化を実現する研究がなされている [4-11].

一方で、近年のクラウドサービスやウェブアプリケーションには、高いサービスの品質が求められている。つまり、データセンタは、障害や環境変動、システムの再構成などの変動に対して、高い信頼性及び頑強性を要求されている。一般的に、データセンタの故障予測は困難であるため [12], データセンタの信頼性や頑強性を向上させるには、予備系機器の配置が不可欠となる。このとき、予備系機器の増加は信頼性及び頑強性の向上につながるが、同時に消費電力の増大にもつながる。そのため、データセンタの信頼性及び頑強性を維持しつつ、消費電力の低減を図ることが求められる。本報告においては、データセンタにおけるサーバへの仮想マシンの配置に着目し、信頼性や頑強性を損なうことなく、データセンタの低消費電力化を実現するような、主系及び予備系のサーバ群に対する仮想マシン配置問題を取り扱う。

データセンタにおける、仮想マシン配置問題についての多くの既存研究が存在し [10, 13, 14], データセンタの低消費電力化のための仮想マシン配置に関する研究も盛んに行われている [15-17]. しかし、これらの研究は、低消費電力化のみに着目しており、データセンタの頑強性については考慮されていない。また、ネットワーク負荷を考慮に入れた低消費電力化の研究も行われているが [18], スループットなどのネットワークの指標を用いて、データセンタのサービス品質を評価しているため、データセンタに大規模な障害が発生した際に、高品質のサービスが保証されるものではない。例えば、ラックのスイッチが故障した際に、そのラック中のサーバが稼働させている仮想マシンは、すべて他のラックのサーバへ移動させなければならないため、移動にかかる時間だけサービスが中断する。

一般に、システムの冗長化を図るには、1つの構成要素に対して専用の予備機器を用意する1対1の冗長化を行うことが考えられるが、実現のコストが高いため、特に大規模なデータセンタには向かない。一方、複数の構成要素が1つの予備機器を共有することでコストは下がるが、共有関係の構築方法によっては、信頼性が下がる可能性がある。例えば、ある構

成要素が予備機器を使用している際に、その予備機器を共有している他の構成要素に何らかの変動があり、その構成要素が予備機器を必要としても、すでに予備機器は使用されているため、その構成要素は復旧できなくなる。1つの予備機器を共有する冗長化手法を用いて、システムの構成機器の故障だけでなく、環境変動や、システムの再構成に対しても、システムの頑強性を維持しつつ、消費電力低減を達成する手法は確立されていない。

ところで、生物ネットワークの特性を情報ネットワークアーキテクチャへ活かす観点での研究が近年活発に行われている。例えば、ネットワークのトポロジカルな特徴であるスケールフリー性 [19] やスモールワールド性 [20] を情報ネットワークのトポロジ設計などに適用・応用する検討が近年活発に行われている [21-23]。しかし、それらの検討はネットワーク層以下のネットワークの理解や構築に主に着目しており、より上位レイヤのネットワークやアプリケーションサービスへの適用例は多くない。情報システム全体の頑強性や持続的成長性を確保するためには、よりアプリケーションやユーザに近いレイヤのネットワークシステムに対しても生物ネットワークの特性を取り込むことが重要であると考えられる。

また、遺伝子ネットワークや神経回路ネットワークなどの生物ネットワークの解析が近年進んでいる。この分野では、遺伝子ネットワークにおいて各遺伝子が、ある環境では異なる機能や出力を行うが、異なる環境では等しい、あるいは非常に似た出力を行う、縮退特性 [24] という性質が注目されている。近年、この性質が生物ネットワークの頑強性、環境変動への耐性、環境適応性、持続的成長性などを説明するものとして説明されている [25-28]。

その一例として、[29]において、ネットワーク化バッファリングと呼ばれる性質が注目されている。ネットワーク化バッファリングとは、縮退性をもつ遺伝子で構成されるネットワークにおいて、ある発現パターンでは定められたタンパク質を発現できない状態が発生した際に、そのタンパク質の発現とは関係のない遺伝子の発現パターンがネットワークを介して連鎖的に変化することで、結果として求めてられていたタンパク質を発現できるという性質のことである。このような性質により、遺伝子ネットワークなどの生物ネットワークは、高い頑強性や効率のよい冗長性を保有していると考えられている。これは、ある同じ機能を果たす機器を複数台用意することにより冗長構成を行い、頑強性を高めている従来のデータセンタやその他の情報システムでは考えられていないものである。

そこで、本報告においては、古来様々な環境変動や障害に対して耐えてきた生物の仕組みからヒントを得ることによって、少ない冗長資源で高い頑強性を有する、仮想マシン配置手法を提案する。つまり、生物ネットワークのトポロジに見られる特徴である縮退特性を応用した、データセンタの仮想マシン配置手法を提案する。提案手法は、従来のデータセンタに見られる1対1や1対多の冗長化ではなく、データセンタ全体として必要な冗長資源を保持する多対多の冗長化を実現する。具体的には、縮退特性をもつ生物ネットワークに見られる、ネットワーク化バッファリングと呼ばれる性質を利用し、多対多の冗長化を少ない予備機器



によって実現する。また、提案手法をシミュレーションによって評価し、従来の単純な手法に比べて、少ない追加資源で高い頑強性を持つことを示す。冗長資源量を削減することで、冗長なサーバに係る消費電力を低減させることができると考えられるため、提案手法は高い頑強性を維持しつつ、同時にデータセンタの低消費電力化を実現することができる。また、本報告においては、仮想マシンをインストールする物理サーバの決定に着目しているため、仮想マシンの実行場所とネットワークの構成方法に関して、既存の研究を取り入れることにより、ネットワーク負荷も考慮した仮想マシン配置が行えると考えられる。

本報告の構成は以下のとおりである。2章において、本報告において着目する生物ネットワークの縮退特性、およびネットワーク化バッファリングと呼ばれる性質について説明し、データセンタの仮想マシン配置への応用について議論する。3章において、本報告で扱うデータセンタモデルを示す。4章において、本報告で用いるデータセンタの仮想マシン配置手法と、環境変動や障害発生などへの対応手法について説明を行う。5章において、シミュレーションによる提案手法の性能評価結果を示す。最後に、6章で本報告のまとめと今後の課題について述べる。

## 2 生物ネットワークの縮退特性とデータセンタへの応用

### 2.1 縮退特性

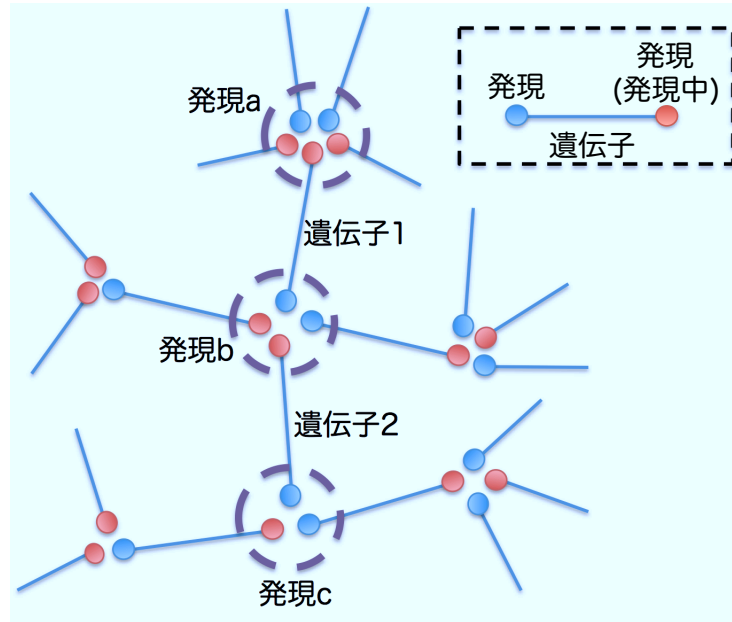
生物分野における縮退特性とは、システムの構成要素の提供する資源や機能が部分的に重複しており、その各構成要素が、ある環境では異なる資源や機能を提供するが、別の環境においては等しい、あるいは非常に近い資源や機能を提供する性質のことであり、生物システムの至る所でみられる性質である [30]. 例えば、出芽酵母のアドヘシンにみられる特徴として、初期段階では単一の機能しか提供できないのだが、発現レベルが上がると、別の機能を提供していたアドヘシンどうしが互いの機能を提供できるようになり、ここに縮退特性が見られる [31]. また、大腸菌の NADPH 代謝において、可溶性および膜結合トランスヒドロゲナーゼである UdhA と PntAB は、必要に応じて互いの機能をかわりに果たすことができる [32]. その他にも、遺伝子発現によるタンパク質の合成や、脳の損傷における補償活動などの分野において、縮退特性が観察されることが知られている [33, 34].

### 2.2 ネットワーク化バッファリング

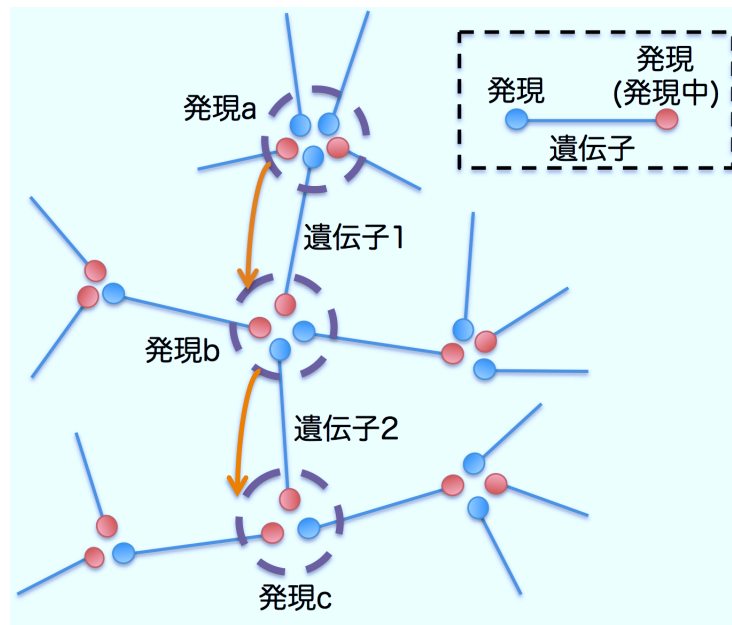
[29] で示されている、縮退特性がもたらす特性の一つであるネットワーク化バッファリングとは、複数の遺伝子の発現パターンが部分的に重複し、ネットワーク全体として冗長資源を保持しているということを表している。これにより、ある発現パターンによる形質発現が不足した際に、その発現に直接的に対応した遺伝子が不足していたとしても、その発現とは直接は関係のない遺伝子を含めた複数の遺伝子の発現パターンが適応的に変化することによって、結果として必要な発現パターンを確保することができる。この性質は、遺伝子ネットワークなどの生物ネットワークシステムが強い頑強性や環境変動への耐性を保有している 1 つの根拠とされている。

遺伝子とその発現パターンの関係を表したネットワークモデルを図 1(a) に示す。図においては、1 本の線分が 1 つの遺伝子を表し、両端が、その遺伝子における可能な発現パターンを表している。この例では、各遺伝子が 2 種類の発現パターンを持つものとしている。また、点線の円で囲まれた部分はある 1 つの発現パターンを表す。その中に複数の遺伝子の発現パターンが存在することは、複数の遺伝子の発現パターンが重複していることを意味している。このネットワークの特徴として、各遺伝子の発現パターンは完全に重複しているのではなく、部分的に重複し、その重複関係がネットワークを構築していることが挙げられる。

図 1(a) において、発現 a-c がそれぞれ 2 つ以上の遺伝子によって発現されていることを求められている場合を考える。図 1(a) の状態においては、発現 c が不足している。このとき、遺伝子 2 の発現を b から c に変化させると、発現 b が不足する。このように、発現に直接関



(a) 遺伝子ネットワークモデル



(b) 発現パターンの連鎖的な変化による不足の解消

図 1: ネットワーク化バッファリングの概念図

わる遺伝子の発現パターンを変化させるだけでは環境変動に対応できない場合がある。しかし、図1(a)のような発現パターンが重複しているネットワークにおいては、ネットワークを介して適応的に発現パターンを変更することにより、不足を解消できることがある。この例においては、遺伝子1の発現パターンがbに変化し、遺伝子2の発現パターンがcに変化することで、不足が解消される。その様子を図1(b)に示す。

このようにして、部分的に重複するネットワークを構成することで様々な環境変動による発現の不足に対して柔軟に対応することができるようになる。この性質が、生物ネットワークの頑強性、環境変動への耐性、環境適応性、持続的成長性の1つの根拠として説明されている。

### 2.3 データセンタにおける仮想マシン配置への適用

遺伝子ネットワークにみられるネットワーク化バッファリングの性質は、情報ネットワークアーキテクチャへ応用できるとされている [35,36]。データセンタの仮想マシン配置における冗長構成もその例外ではない。すなわち、限られた複数の仮想マシンを実行できる物理マシンを用意し、それらの実行できる仮想マシンを部分的に重複させることによって、ネットワークを構築し、データセンタに環境変動や障害が発生した場合に、ネットワーク内の複数の物理マシンの仮想マシンの実行を適応的に切り替えることによって、仮想マシンの実行不足を解消することができる。

従来のデータセンタにおいて、これまで考えられてきた冗長化手法においては、1つの物理サーバに対して専用の冗長資源を確保する1対1の冗長化手法や、高性能な物理サーバを導入し、必要とされるすべての仮想マシンを実行可能な冗長資源を用意するという手法が取られてきた。しかし、前者の手法では、個々の冗長資源は固定的な機能を提供すればよいが、故障や環境変動をあらかじめ予測して、それに応じた冗長資源を用意しなければならないため、想定していないような故障や環境変動が発生すると、冗長資源の過不足が発生する。また、後者の手法では、高い性能を持つ物理サーバを多数必要とするため、実現のコストが大きくなる。

それに対し、ネットワーク化バッファリングに基づく冗長化手法は、各冗長資源は限られた数の仮想マシンを実行できればよく、ある仮想マシンの実行が不足した際には、ネットワークを介して各物理サーバの実行する仮想マシンを適応的に切り替えることによって、実行不足を解消できるため、従来の冗長化手法に見られる欠点を克服できると考えられる。

### 3 データセンタモデルと障害発生モデル

本章では、本報告における提案手法の性能評価のために用いるデータセンタモデルと、想定する障害発生に関するモデルの説明を行う。

#### 3.1 データセンタモデル

本報告で用いるデータセンタのモデルを以下のように定義する。物理サーバはラックに収容されており、ネットワークスイッチに接続されている。一般的にデータセンタにおけるネットワークスイッチの配置方法にはEoR (End of Row) と ToR (Top of Rack) の2種類があるが、現在の主流はToRである [15] ので、本報告が対象とするモデルについてもToRを採用する。つまり、各ラック中のすべての物理サーバはそのラックに設置された1台のToRスイッチに接続されている。物理サーバはストレージを持っており、物理サーバ上で動作する仮想マシン (VM) を複数インストールできる。また、物理サーバはマルチコアCPUを持つものとし、同時に実行可能なVMの上限は物理サーバのCPUコア数に等しいとする。

データセンタを用いてVMを実行するテナント構成として、マルチテナント型データセンタを想定する。マルチテナント型データセンタとは、複数の異なるテナントが同一のデータセンタ環境を共同で利用するものである。ここでは、テナントをVMの部分集合であると定義し、テナントは、自身に含まれるそれぞれのVMについて必要とする実行数だけ実行することを物理サーバに要求する。物理サーバはテナントからの要求に従って、どのVMをインストールするか、かつ、そのうちのどのVMを実行するかを決定する。そのとき、データセンタ全体で、テナントが要求している実行数を満たすようにVMを実行する。マルチテナント型データセンタにおける、ラックへのサーバ収容、およびサーバのVMのインストールと実行の関係を図2に示す。図中において、青色の矢印はサーバがVMをインストールしているが実行はしていない状態を示し、赤色の矢印はサーバがVMをインストールしてかつ実行もしている状態を示している、また黄色の矢印はラックがサーバを収容している状態を示している。

データセンタモデルにおける変数定義は以下のとおりである。ラックの総数を  $N$  とし、 $i$  番目のラックを  $r_i$  とする。ラック  $r_i$  に収容されたサーバ数を  $M_i$  とし、そのうち  $j$  番目のサーバを  $s_{ij}$  とする。サーバの集合を  $Z = \{s_{11}, s_{12}, \dots, s_{1M_1}, s_{21}, \dots, s_{NM_N}\}$  とする。また、VMの総数を  $V$  とし、その集合を  $Y = \{v_1, v_2, \dots, v_V\}$  とする。テナントの総数を  $O$  とし、 $k$  番目のテナントがもつVMの集合を  $Y_k \subseteq Y$  とする。ただし、 $(\forall i, \forall j Y_i \cap Y_j = \phi)$  かつ  $(Y_1 \cup Y_2 \cup \dots \cup Y_O = Y)$  とする。また、物理サーバ  $s_{ij}$  のCPUコア数は  $C_{ij}$ 、インストールできるVMの数を  $I_{ij}$  と定義する。

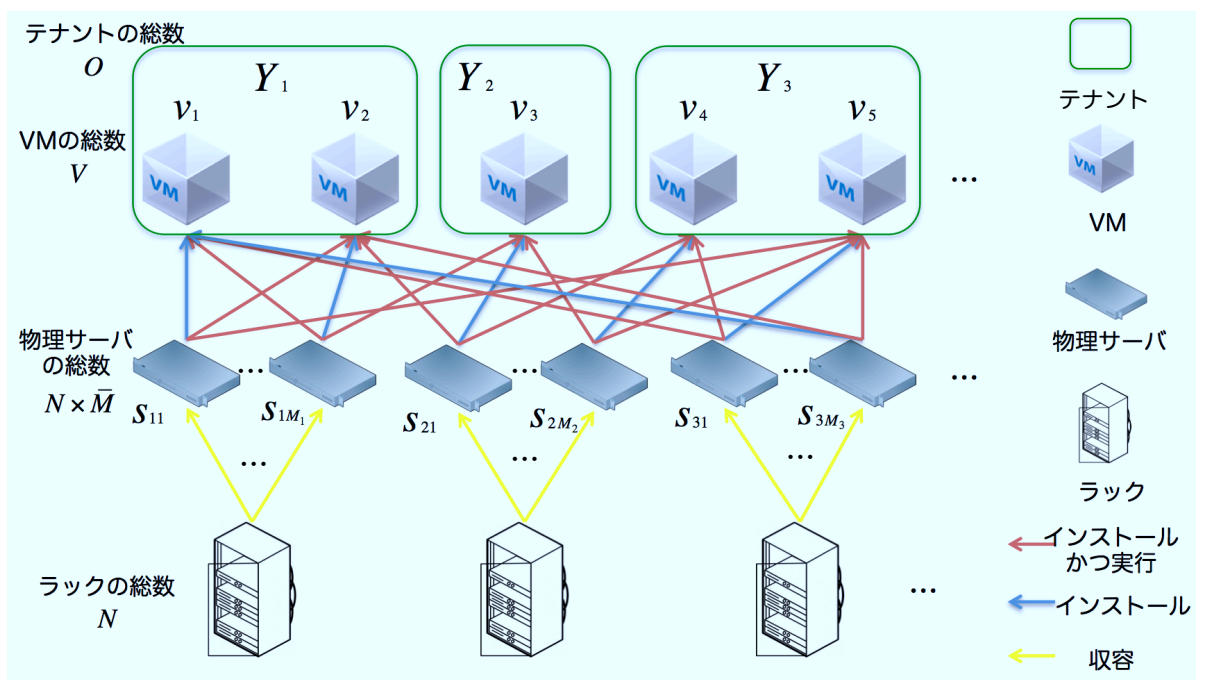


図 2: データセンタモデル

サーバ  $s_{ij}$  が VM  $v_k$  をインストールしているかどうかを表す変数を  $\delta_{ijk}$  とし、以下のよう  
に定義する。

$$\delta_{ijk} = \begin{cases} 1 & \text{サーバ } s_{ij} \text{ が VM } v_k \text{ をインストールしている} \\ 0 & \text{サーバ } s_{ij} \text{ が VM } v_k \text{ をインストールしていない} \end{cases}$$

同様に、サーバ  $s_{ij}$  が VM  $v_k$  を実行しているかどうかを表す変数を  $\gamma_{ijk}$  とし、以下のよう  
に定義する。

$$\gamma_{ijk} = \begin{cases} 1 & \text{サーバ } s_{ij} \text{ が VM } v_k \text{ を実行している} \\ 0 & \text{サーバ } s_{ij} \text{ が VM } v_k \text{ を実行していない} \end{cases}$$

インストールできる VM の数、及び同時に実行可能な VM 数に関する制約は以下のよう  
なる。

$$\forall i, \forall j \left( \left( \sum_{k=1}^V \delta_{ijk} \leq I_{ij} \right) \wedge \left( \sum_{k=1}^V \gamma_{ijk} \leq C_{ij} \right) \right)$$

ある環境において、VM  $v_k$  を実行している物理サーバの台数を  $T_k^P$  とすると、 $T_k^P$  は以下  
の式により求められる。

$$T_k^P = \sum_{s_{ij} \in Z} \gamma_{ijk}$$

また、VM  $v_k$  が必要とされる実行数を  $T_k^E$  とすると、VM  $v_k$  における不足数  $\theta_k$  は、

$$\theta_k = \begin{cases} 0 & T_k^P > T_k^E \\ T_k^E - T_k^P & \text{else} \end{cases}$$

となる。これより、すべての VM の実行数の不足状況を示す指標  $F$  を、

$$F(T^P) = \sum_{v_k \in Y} \theta_k$$

と定義する。またテナント  $Y_i$  ごとの VM の実行数の不足状況を示す指標  $F_i$  を、

$$F_i(T^P) = \sum_{v_k \in Y_i} \theta_k$$

と定義する。

### 3.2 データセンタにおける障害発生モデル

すべての VM の実行数の不足状況を示す指標  $F$  について、 $F = 0$  のとき、データセンタ全体が健全であると定義する。また、テナント  $Y_i$  についても同様に、 $F_i = 0$  のとき、そのテナントは健全であると定義する。データセンタにおける障害とは、データセンタ全体として健全な状態が保てなくなることを意味している。具体的には、データセンタで発生する障害として、サーバ障害とラック障害の2種類を想定する。サーバ障害とは、ディスクや CPU などのサーバの構成物の故障を想定している。一方、ラック障害とは、ToR スイッチの故障により、ラックに収容されているすべての物理サーバのネットワークへの接続が切断される障害を想定している。5章における性能評価においては、サーバ障害を想定する場合には、指定した台数のランダムに選択した物理サーバを停止させる。その結果、停止した物理サーバが実行している VM は停止する。一方、ラック障害を想定する場合には、指定した数のランダムに選択したラックを停止させる。物理サーバが接続されているラックが停止した場合には、その物理サーバが実行している VM は停止する。VM の実行が停止した結果、必要とされている実行数を満たしていない VM があれば、その VM が所属するテナントの健全性が保てなくなってしまう。また、障害の度合いを評価する指標として、全テナント数のうち健全なテナント数の割合を  $F_{num}$ 、各テナント中の VM のうち必要実行数を満たしている VM の割合の最小値を  $F_{min}$  と定義する。前者の指標は、障害の影響がなかったテナントの規模を、後者は、テナント間の公平性を表すものである。

### 3.3 障害からの回復

3.2節で示した障害発生の結果、テナントが持つ VM の実行数が不足すると、データセンタは稼働していない物理サーバや CPU コアなどの冗長資源を利用して、VM の実行不足を解消する。その際、物理サーバが事前にインストールしていない VM を起動するには、VM の物理サーバ間の移動を必要とするため、起動までの時間として、VM のデータ容量や移動元と移動先の物理サーバ間の距離などによって変動するが、数十秒から数百秒を必要とする [37]。そのため、ここでは、障害からの回復を迅速に行うため、物理サーバは事前にインストールした VM のみを実行するものとする。従って、障害に対する頑強性を高めるには、事前にどの VM をどの物理サーバへインストールするかが重要になる。本報告では、2章に示した、生物ネットワークの縮退特性を応用した手法を提案する。



## 4 提案手法

本章では，生物ネットワークの縮退特性を応用した仮想マシン配置手法，および，障害からの回復手法について説明する．

### 4.1 仮想マシン配置手法

ここでは，物理サーバへの VM の配置手法についての詳しい説明を行う．各手法を表す図においては，ラック数  $N$  は 2，ラックに収容できるサーバ数  $M$  は 4，各サーバの VM のインストール可能上限数  $I_{ij}$  は 2 としている．

#### 既存手法 1 (Existing Method 1)

既存の冗長化手法として，同じ種類の VM をインストールした物理サーバを複数台用意することで頑強性を高めるものがある．例えば，各 VM をそれぞれ 1 つのラックに割り当て，そのラック中の複数のサーバへインストールするという手法が考えられる．この手法を既存手法 1 とし，これを実現する具体的な手順を以下に示す．まず，VM を  $v_1, v_2, \dots, v_V$  と並べる．並べた VM をラック数である  $N$  で等分し，VM のグループ分けを行う．ラックも同様に  $r_1, r_2, \dots, r_N$  と並べ，順番に VM のグループを割り当てる．そして各 VM を，所属するグループの VM の番号が低い順に，割り当てられたラックに収容されているサーバを 1 つランダムに選択し，そのサーバへインストールしていく．グループ内のすべての VM をインストールし終わったら，グループ内の最初の番号の VM から同様にインストールしていく．これを，ラックに収容されているすべてのサーバのインストール上限数に達するまで繰り返す．こうすることで，グループ内の VM のインストール数を均一にすることができる．既存手法 1 によるサーバへの VM 配置の様子を図 3 に示す．

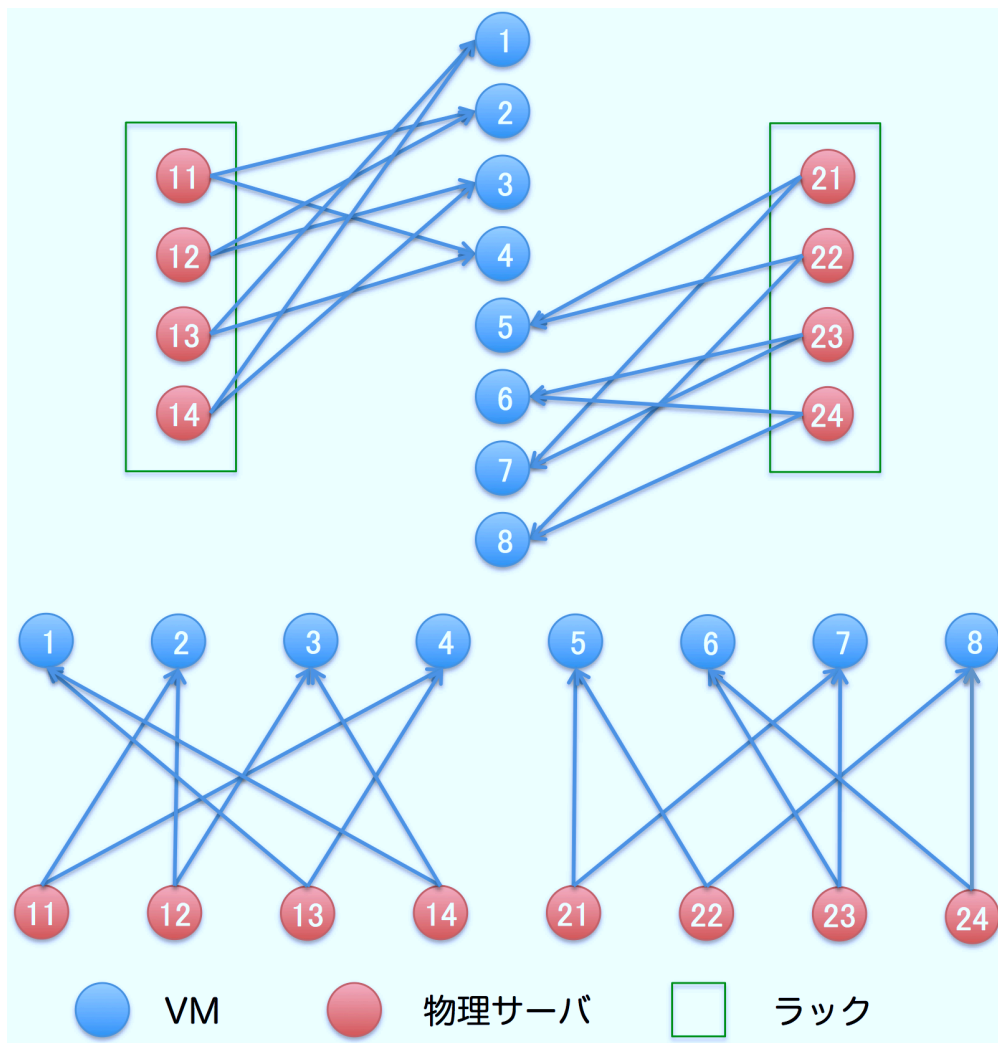


図 3: 既存手法 1

## 既存手法 2 (Existing Method 2)

既存手法 1 では、ラック障害の影響を考慮していない。そのため、ラック障害の際には、そのラックに割り当てられている VM を実行できる物理サーバが他に存在せず、障害からの回復が不可能となる。そこで既存手法 2 では、ラックごとに VM を割り当てるのではなく、異なるラックに収容されたサーバでグループを構成し、そのグループに対して VM を割り当てる。具体的な手順を以下に示す。まず、VM を  $v_1, v_2, \dots, v_V$  と並べる。並べた VM を、ラックに収容できるサーバ数の最小値である  $M_{min}$  で等分し、VM のグループ分けを行う。次に、物理サーバを  $s_{11}, s_{12}, \dots, s_{1M_1}, s_{21}, \dots, s_{NM_N}$  とならべ、 $s_{11}$  は VM の 1 番目のグループへ、 $s_{12}$  は VM の 2 番目のグループへ、というように、順に VM のグループへ割り当てていく。VM の最後のグループまで割り当てたら、最初のグループに戻り、続けて物理サーバを割り当てていく。これをすべての物理サーバに関して行う。物理サーバをグループへ割り当てたら、各 VM を、所属するグループの VM の番号が低い順に、割り当てられたサーバのグループのうち 1 つランダムに選択し、そのサーバへインストールしていく。VM のグループ内のすべての VM をインストールし終わったら、VM のグループ内の最初の番号の VM から同様にインストールしていく。これを、割り当てられたすべてのサーバのインストール上限数に達するまで繰り返す。既存手法 2 によるサーバへの VM 配置の様子を図 4 に示す。

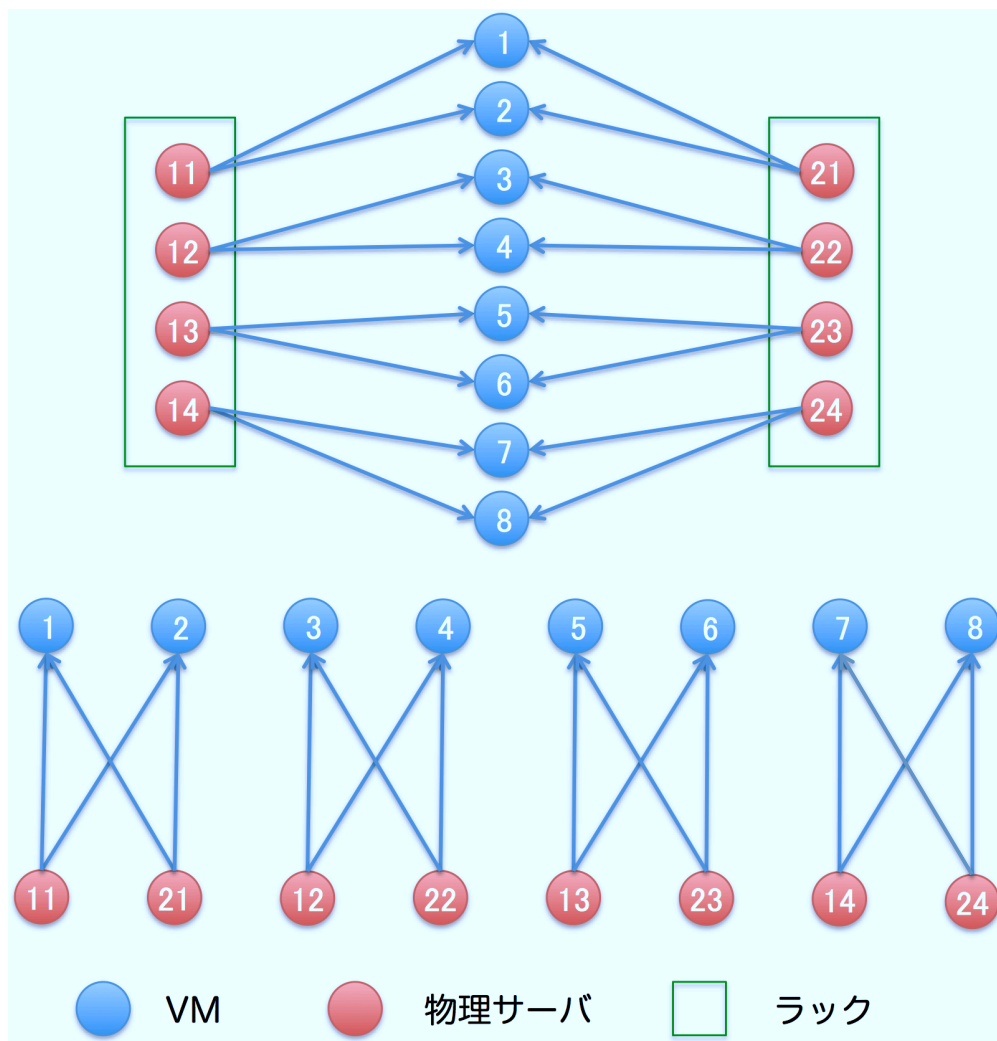


図 4: 既存手法 2

### 規則的部分重複手法 1 (Regular Degenerate Method 1)

部分的重複関係を構築する手段として、各物理サーバがインストールする VM を、規則的にずらす方法が考えられる。具体的な手順を以下に示す。まず、物理サーバを  $s_{11}, s_{12}, \dots, s_{1M_1}, s_{21}, \dots, s_{NM_N}$  と並べ、VM を  $v_1, v_2, \dots, v_V$  と並べる。続いて、各物理サーバに対して用意した VM を物理サーバへ並べた順にインストールしていく。その際、 $s_{11}$  に関しては  $v_1$  から並べた順に  $I_{11}$  個、 $s_{12}$  に関しては  $v_{1+I_{11}/2}$  から並べた順に  $I_{12}$  個というように、 $I_{ij}/2$  だけずらして、インストール可能上限数だけ VM を用意する。このようにして用意した VM を各物理サーバへインストールしていく。こうすることで、規則的に部分重複関係を構築することができる。規則的部分重複手法 1 によるサーバへの VM 配置の様子を図 5 に示す。

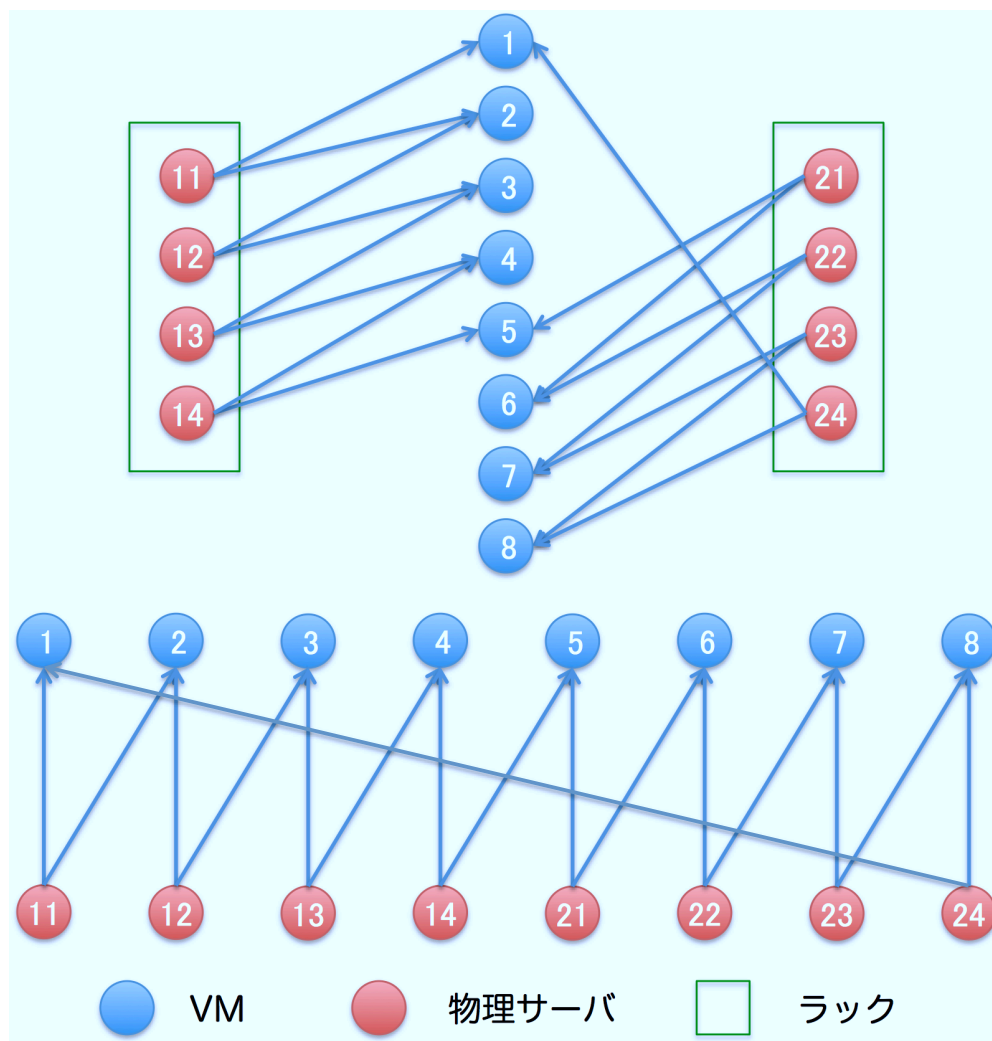


図 5: 規則的部分重複手法 1

## 規則的部分重複手法 2 (Regular Degenerate Method 2)

規則的部分重複手法 1 では、物理サーバのラックへの収容状況を考慮していないため、1つのラック中に収容されている各物理サーバがインストールしている VM に偏りが生じる。そのため、ラック障害が発生すると、特定の VM をインストールしている物理サーバが少なくなり、障害からの回復が行えない場合が発生する。そこで、規則的部分重複手法 2 では、重複関係を異なるラックに収容された物理サーバ間で構築する事により、上記の問題を解決する。具体的な手順を以下に示す。まず、物理サーバを  $s_{11}, s_{21}, \dots, s_{N1}, s_{12}, \dots, s_{NM_N}$  と並べ、VM を  $v_1, v_2, \dots, v_V$  と並べる。規則的部分重複手法 1 と同様に各物理サーバに対して VM を用意し、それらの VM を物理サーバへインストールしていく。こうすることで、部分的な重複関係をラック内で構築するのではなく、他のラックと構築できるため、ラックの収容を考慮できていると考えられる。規則的部分重複手法 2 によるサーバへの VM 配置の様子を図 6 に示す。

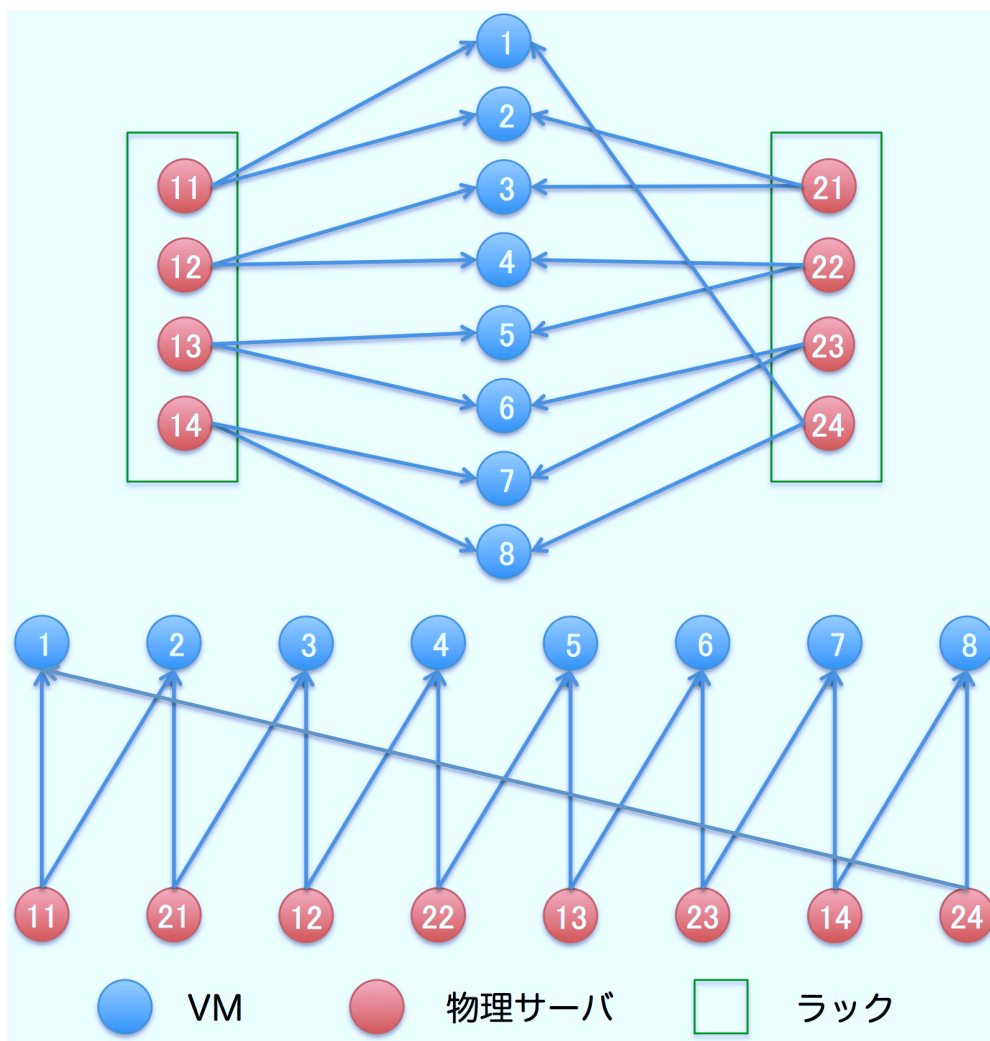


図 6: 規則的部分重複手法 2



### 無作為部分重複手法 1 (Random Degenerate Method 1)

部分的重複関係を構築する手段として、VMをランダムに選択したサーバへインストールする方法が考えられる。具体的な手順を以下に示す。VMを $v_1, v_2, \dots, v_V$ の順に、物理サーバ $s_{11}$ へインストールしていく。物理サーバ $s_{11}$ のインストール可能数の上限に達すると、 $s_{12}, \dots, s_{1M_1}, s_{21}, \dots, s_{NM_N}$ の順で、続けてVMをインストールしていく。 $v_V$ までインストールが終わったら、再び $v_1$ からインストールを再開する。これを、物理サーバ $s_{NM_N}$ のインストール可能数の上限に達するまで繰り返す。そのうち、2つの物理サーバをランダムに選択し、それぞれの物理サーバがインストールしているVMを1つランダムに選択し、それらを交換する。これを十分に大きな回数だけ繰り返す。こうすることで、各VMのインストール数が均一であり、かつVMをランダムなサーバにインストールすることができる。無作為部分重複手法1によるサーバへのVM配置の様子を図7に示す。

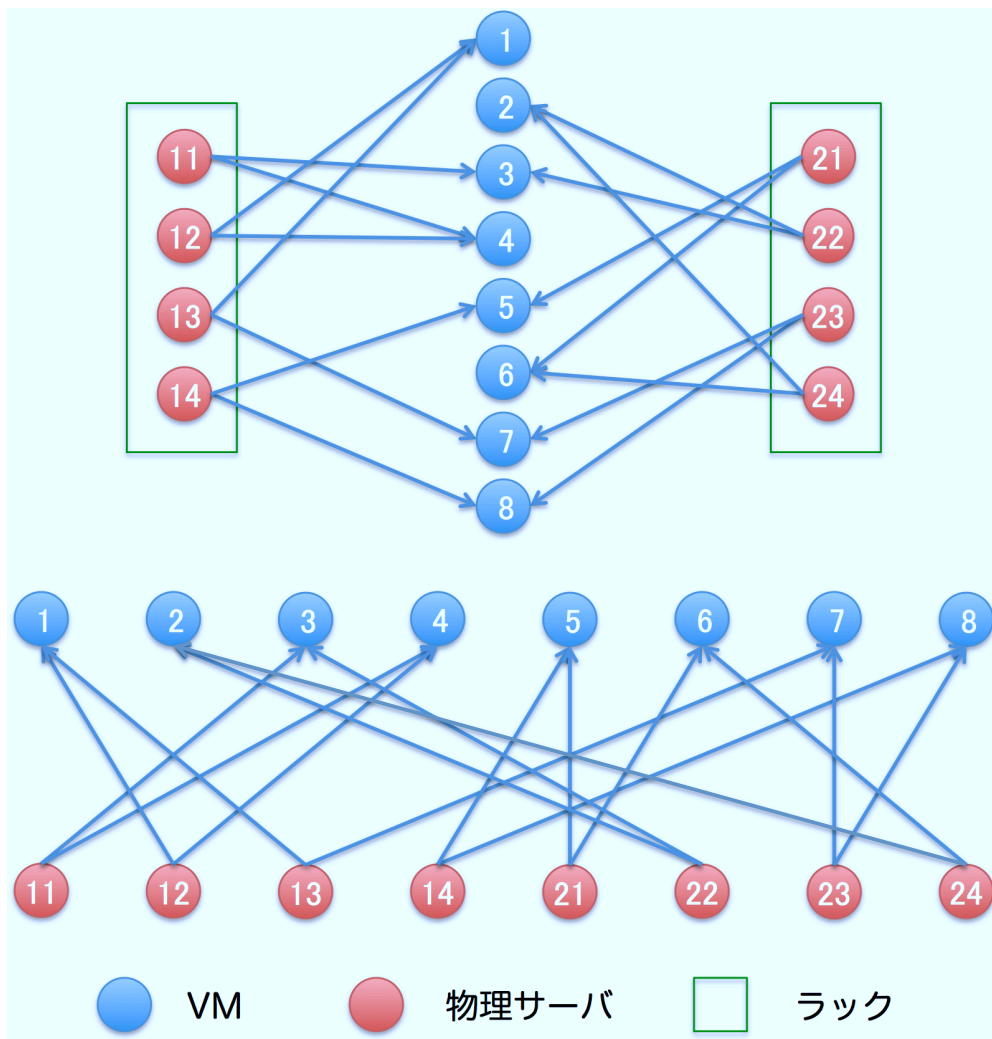


図 7: 無作為部分重複手法 1

## 無作為部分重複手法 2 (Random Degenerate Method 2)

無作為部分重複手法 2 では、無作為部分重複手法 1 が考慮していない、物理サーバのラックへの収容状況を考慮する。すなわち、ラックに収容されている物理サーバがインストールしている VM にばらつきがでないように、かつ、ランダムに VM をインストールする。具体的な手順を以下に示す。まず、VM を  $v_1, v_2, \dots, v_V$  と並べる。そして、VM が並べた順に、ラック  $r_1$  に収容されているサーバをランダムに 1 つ選択し、その VM を選択したサーバへインストールする。  $v_V$  までインストールしたら、  $v_1$  にもどり、同様にしてサーバへインストールしていく。これをラック  $r_1$  に収容されているすべてのサーバのインストール可能上限数に達するまで繰り返す。ラック  $r_1$  へのインストールが終わると、続いて  $r_2, \dots, r_N$  の順に同様にしてインストールしていく。こうすることで、ラック内により多くの VM をインストールすることができ、かつ VM をランダムなサーバにインストールすることができる。無作為部分重複手法 2 によるサーバへの VM 配置の様子を図 8 に示す。

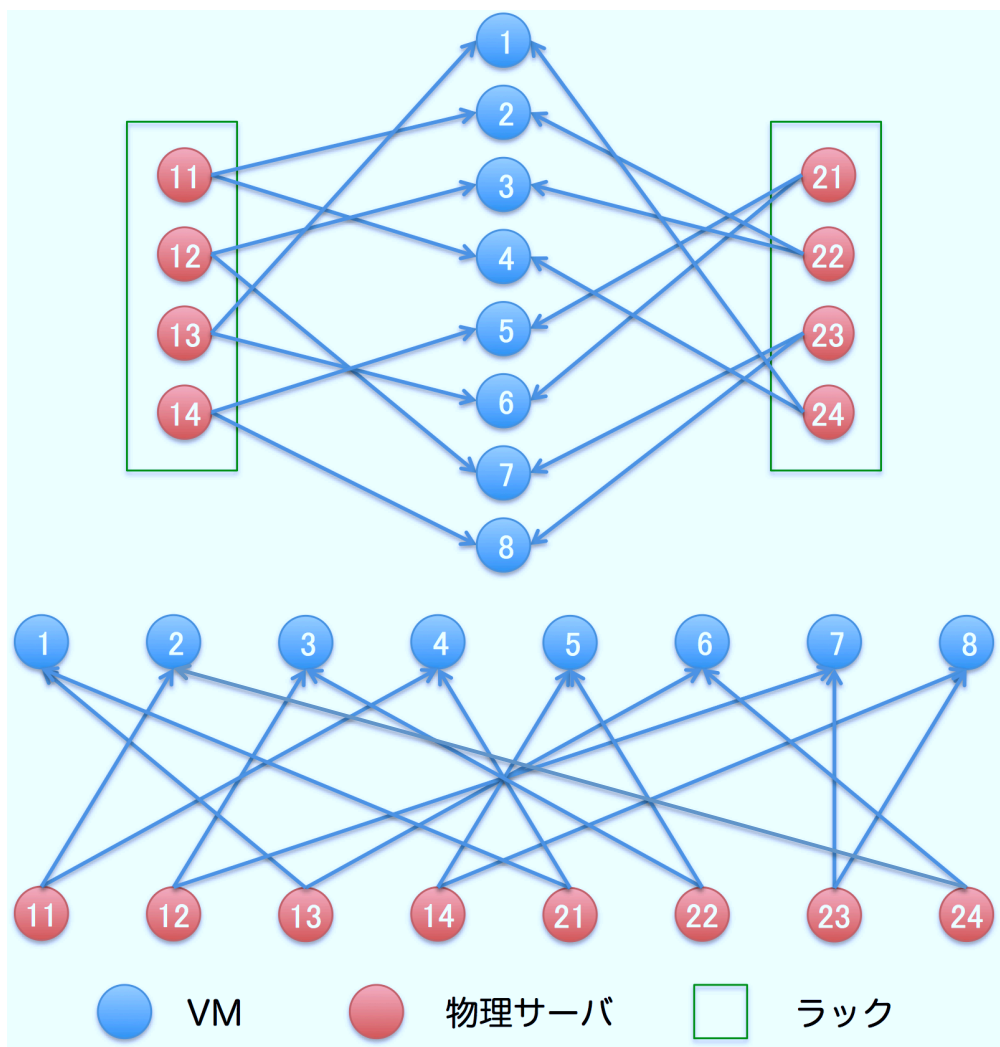


図 8: 無作為部分重複手法 2

## 4.2 環境変動や障害発生後の VM 実行切り替えアルゴリズム

本報告においては、データセンタにおける VM 配置において縮退特性を応用し、頑強な構成を構築することが目的である。従って、環境変動や障害発生後に VM 実行を切り替えるアルゴリズムに関しては、単純な手法を用いる。これにより、縮退特性を応用した構成とそうでない構成の性能比較を行う。[38] に提案されているような、化学反応式を応用した環境変動や障害発生への対応方法を適用することは今後の課題とする。

以下では、環境変動や障害発生によって生じる VM の実行不足を解消するための VM 実行切り替えアルゴリズムの説明を行う。このアルゴリズムにおいては、3.2 節に示した障害の規模を示す指標である  $F_{num}$ 、または  $F_{min}$  を用いる。前者はより多くのテナントを復旧したい場合に、後者はテナント間での公平性を保ちたい場合に用いる。また、各指標の理想値を、前者の指標においては、全テナント数と等しい値、後者の指標においては、1 であると定義する。

具体的なアルゴリズムを以下に示す。

1. 現在の指標  $F$  を評価し、 $F_{before}$  とする。 $F_{before}$  が理想値と等しい時、不足はないとして、アルゴリズムを終了する。
2. 実行数が不足している VM の集合である  $Y^s$  を求める。

$$Y^s = \{v_k \mid v_k \in Y, \theta_k > 0\}$$

$Y^s = \phi$  の場合、実行数が不足していて、かつ不足の解消ができる可能性のある VM は存在しないということになる。この時、不足の解消が不可能であり、かつ指標の値は現在の実行状況で一番良い値となると判断し、不足解消のアルゴリズムを終了する。

3.  $Y^s$  からランダムに選んだ VM  $v_k$  をインストールしていて、かつ現在実行をしていない物理サーバの集合を  $Z_k^p$  とし、この集合を求める。

$$Z_k^p = \{s_{ij} \mid s_{ij} \in Z, \exists v_k \in Y^s (\delta_{ijk} = 1 \wedge \gamma_{ijk} = 0)\}$$

$Z_k^p = \phi$  のとき、 $v_k$  の実行数を  $T_k^E$  と等しくすることは不可能となる。よって、 $v_k$  を  $Y^s$  から除外し、2. に戻る。

4.  $Z_k^p$  から余剰コアの最も多い物理サーバを選択する。複数ある場合には、その中からランダムに選択する。その物理サーバ  $s_{ij}$  に余剰コアが存在する場合、つまり  $\sum_{v_k \in Y} \gamma_{ijk} < C_{ij}$  のとき、6. に進む。そうでなければ、物理サーバ  $s_{ij}$  がインストールしていて、かつ現在実行している VM の集合  $Y_{ij}^p$  を求める。

$$Y_{ij}^p = \{v_k \mid v_k \in Y, \delta_{ijk} = 1 \wedge \gamma_{ijk} = 1\}$$

5.  $Y^p$  からランダムに選択した VM  $v_l$  の実行を中止する.

$$\gamma_{ijl} = 0$$

6. 現在不足している VM  $v_k$  を実行する.

$$\gamma_{ijk} = 1$$

7. この状態で指標  $F$  を評価し,  $F_{after}$  とする.  $F_{after}$  が理想値と等しい場合, 不足が解消されたということになり, アルゴリズムを終了する.  $F_{after}$  の値が, それまでの一番良い値である  $F_{temp}$  よりも大きい場合 ( $F_{after} > F_{temp}$ ),  $F_{temp}$  に  $F_{after}$  を代入する. また, 変更前の不足状況の指標  $F_{before}$  に  $F_{after}$  の値を代入する.
8. 2.に戻る. また, 2.-7.を繰り返し実行が一定回数  $t$  を超えると, 指標の値は現在の実行状況で一番良い値となると判断し, 不足解消のアルゴリズムを終了する.

## 5 性能評価

本章では、4.1節において提案した仮想マシン配置手法の性能評価を行う。具体的には、提案手法に基づいて仮想マシンの配置を行ったデータセンタに対して、3.2節に示した障害が発生した際の、障害回復能力の評価を行う。また、データセンタ資源の冗長度がデータセンタの障害回復能力に与える影響についても評価する。ここで、本章におけるデータセンタ資源の冗長度とは、データセンタを利用するすべてのテナントのVMを健全に稼働させるために最低限必要な物理サーバ数に対する、余剰な物理サーバの割合を意味する。

### 5.1 評価環境

3章に示したデータセンタモデルを用いて、提案手法の評価を行う。物理サーバ  $s_{ij}$  には、4.1節において提案した冗長化手法に基づき、 $I_{ij}$  個のVMがインストールされる。そのうち、 $C_{ij}$  個を上限に、各テナントが要求するVMの実行数を満たすように、VMを実行する。次に、3.2節に示したようなデータセンタの障害発生を想定し、指定した台数の物理サーバ、またはラックを停止させる。ラックが停止すると、そのラックに収容されているすべての物理サーバの動作が停止する。障害発生後、データセンタ全体が健全な状態でなければ、4.2節に示した障害への対応を行い、復旧を試みる。

以降では、各ラックに収容される物理サーバ数と、各サーバのCPUコア数がそれぞれ均一である環境と、それらがランダムに決定される不均一な環境のそれぞれに対して、2種類のデータセンタ規模を想定して評価を行う。なお、物理サーバ  $s_{ij}$  がインストールできるVMの数  $I_{ij}$  は、CPUコア数  $C_{ij}$  の2倍であるとしている。均一な環境における2種類のデータセンタ規模のパラメータ設定を表1に、不均一な環境における2種類のデータセンタ規模のパラメータ設定を表2に示す。性能評価指標には障害回復割合を用いる。これは各パラメータにおいて1,000回のシミュレーションを行い、4.2節に示したVMの実行切り替えにより、データセンタ全体が健全な状態に回復した割合を示す。

### 5.2 均一な環境における提案手法の評価

まず、VMの実行切り替えアルゴリズムで用いる指標が正しく機能していることを確認する。図9は、小規模データセンタを想定した環境における、冗長度を100%とした場合の、障害が発生したサーバ数と障害回復割合の関係を示している。ここでは、既存手法1と規則的部分重複手法1についての、指標  $F_{num}$  と指標  $F_{min}$  を用いた場合の評価結果をプロットしている。図9(a)に全テナントのうち健全な状態に回復したテナントの割合を、図9(b)に各テナントの回復割合の最小値を、図9(c)にデータセンタ全体の障害回復割合をそれぞれ

表 1: 均一な環境におけるパラメータ設定

(a) 小規模データセンタを想定したパラメータ設定

ラック数 ( $N$ )	8
ラックに収容する物理サーバ数 ( $M$ )	8
VM の総数 ( $V$ )	43,51,64,73,85,102,109,116,122
テナント数 ( $O$ )	6
物理サーバの CPU コア数 ( $\forall i \forall j C_{ij}$ )	4
インストールできる VM の数 ( $\forall i \forall j I_{ij} = C_{ij} \times 2$ )	8
各 VM の必要稼働数 ( $\forall k T_k^E$ )	2

(b) 中規模データセンタを想定したパラメータ設定

ラック数 ( $N$ )	12
ラックに収容する物理サーバ数 ( $M$ )	30
VM 数 ( $V$ )	240,288,360,411,480,576,615,655,686
テナント数 ( $O$ )	36
物理サーバの CPU コア数 ( $\forall i \forall j C_{ij}$ )	4
インストールできる VM の数 ( $\forall i \forall j I_{ij} = C_{ij} \times 2$ )	8
VM の必要稼働数 ( $\forall k T_k^E$ )	2



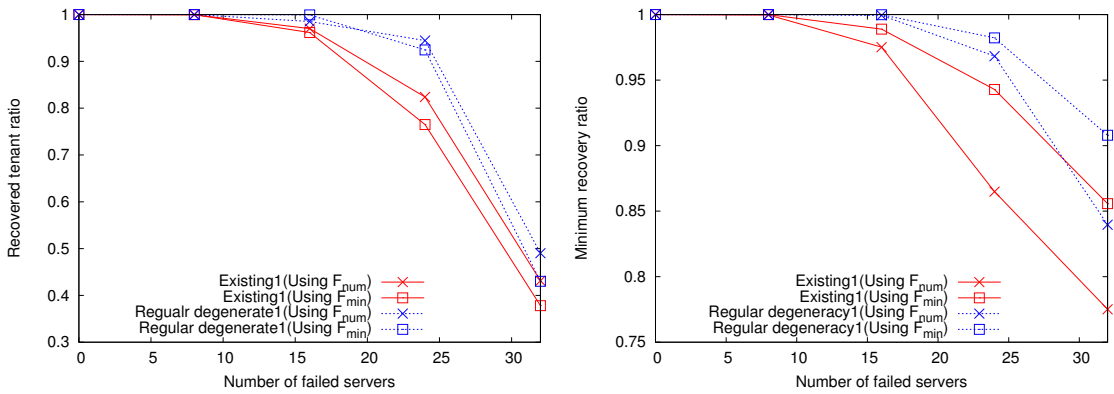
表 2: 不均一な環境におけるパラメータ設定

(a) 小規模データセンタを想定したパラメータ設定

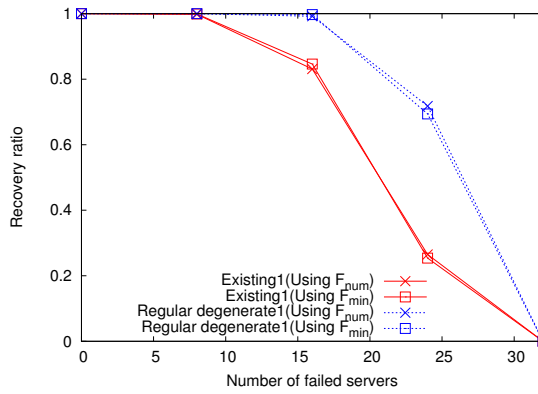
ラック数 ( $N$ )	8
ラックに収容する物理サーバ数 ( $M$ )	6-10 からランダムに決定
VM の総数 ( $V$ )	43,51,64,73,85,102,109,116,122
テナント数 ( $O$ )	6
物理サーバの CPU コア数 ( $\forall i \forall j C_{ij}$ )	2,4,6 からランダムに決定
インストールできる VM の数 ( $\forall i \forall j I_{ij} = C_{ij} \times 2$ )	4,8,12
各 VM の必要稼働数 ( $\forall k T_k^E$ )	2

(b) 中規模データセンタを想定したパラメータ設定

ラック数 ( $N$ )	12
ラックに収容する物理サーバ数 ( $M$ )	20-40 からランダムに決定
VM 数 ( $V$ )	240,288,360,411,480,576,615,655,686
テナント数 ( $O$ )	36
物理サーバの CPU コア数 ( $\forall i \forall j C_{ij}$ )	2,4,6 からランダムに決定
インストールできる VM の数 ( $\forall i \forall j I_{ij} = C_{ij} \times 2$ )	4,8,12
VM の必要稼働数 ( $\forall k T_k^E$ )	2



(a) 全テナント数のうち健全なテナント数の割合 (b) 各テナント中の VM のうち必要実行数を満たしている VM の割合の最小値



(c) データセンタ全体の障害回復割合

図 9: 障害復旧の際に用いる指標が回復性能に与える影響

示している。図より、 $F_{num}$  を指標として用いた場合は、 $F_{min}$  を用いた場合に比べて、全テナントのうち健全な状態に回復したテナントの割合が高くなり、 $F_{min}$  を指標として用いた場合は、 $F_{num}$  を用いた場合に比べて、各テナントの回復割合の最小値が高くなることから、用いる指標に応じた VM の切り替えが期待通りに行われていることがわかる。また、データセンタ全体の障害回復割合は、指標によってほぼ変化しないことがわかる。従って、以降の評価においては、アルゴリズムが用いる指標として、 $F_{min}$  を用いる。

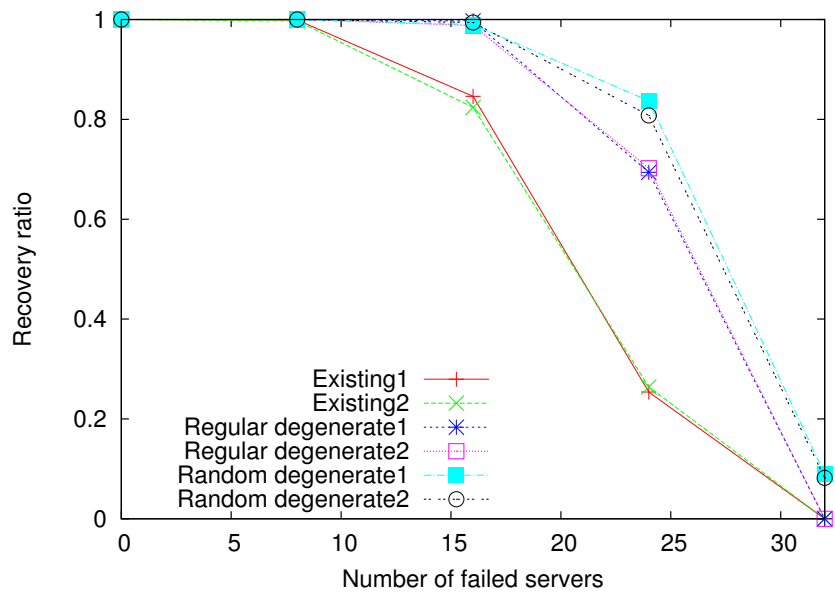
図 10 に、均一な小規模及び中規模データセンタを想定した環境における、サーバ障害が発生した場合の、サーバの故障台数と障害回復割合の関係を示す。ここでは、冗長度は 100%としている。図より、データセンタの規模に関わらず、既存手法に比べて、すべての部分重複手法の性能が優れていることがわかる。これは、ネットワーク化バッファリングに基づく仮

想マシン配置によって、様々な障害発生パターンに対して復旧が可能になっていることを示している。一方、図 10(b) から、既存手法 1 の性能が、部分重複手法にはわずかに及ばないが、優れた性能を持つことがわかる。これは、中規模データセンタを想定した環境では、ラックが収容するサーバ数が多いため、ラック内において部分冗長化がある程度実現したためであると考えられる。また、部分重複手法と比較して、既存手法 1 の障害回復割合が若干低いのは、既存手法 1 が、グループ分けされたサーバと VM における部分的な多対多の冗長化であるのに対して、部分重複手法はネットワーク全体で多対多の冗長構成を行っているため、より多くの障害パターンに対応することができるためである。

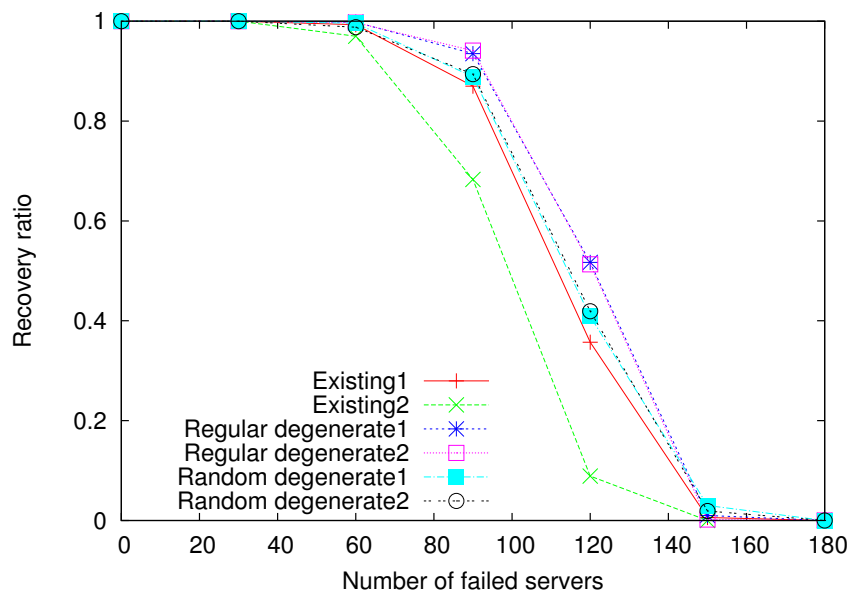
次に、データセンタ資源の冗長度が回復性能に与える影響を評価する。図 11 に小規模データセンタ、図 12 に中規模データセンタをそれぞれ想定した環境における、冗長度を変化させた場合の、サーバの故障台数と障害回復割合の関係を示す。ただし、図 11(a) と図 12(a) は冗長度 100%、図 11(b) と図 12(b) は冗長度 75%、図 11(c) と図 12(c) は冗長度 50%、図 11(d) と図 12(d) は冗長度 25%としている。図より、規則的部分重複手法は、冗長度が小さくなると障害回復割合が大きく低下する事がわかる。これは、冗長度が小さくなり、余剰資源量が減少することで、各 VM の物理サーバへのインストール数にばらつきが発生するためである。一方、冗長度が非常に大きい場合を除き、ほぼすべての場合において、無作為部分重複手法の性能が最も優れていることがわかる。これは、ランダムに VM を物理サーバにインストールすることで、異なる組み合わせの VM をインストールしている物理サーバが増加するためである。このことから、ネットワーク化バッファリングに基づく仮想マシン配置を行う際には、ランダムに VM を物理サーバにインストールすることが効果的であるといえる。

次に、サーバ障害とラック障害の違いが性能に与える影響を評価する。図 13 に小規模及び中規模データセンタを想定した環境における、ラック障害が発生した場合の、サーバの故障台数と障害回復割合の関係を示す。ここでは、冗長度は 100%としている。図 13(a) において、既存手法 2 の性能が障害規模にかかわらず非常に高いことがわかる。これは、小規模データセンタを想定した環境のパラメータ設定にその原因があり、非常に特殊な状況であるため、既存手法 2 の性能を適切に示したものではない。図 13 より、ラック障害を想定した場合においても、既存手法の性能は、部分重複手法に比べて、性能が劣っていることがわかる。一方、部分重複手法に関しては、図 10 に示したサーバ障害の場合は、無作為部分重複手法 1 と 2 の性能が最も優れていたが、ラック障害の場合は、ラック収容を考慮した無作為部分重複手法 2 の性能が最も優れている。これは、無作為部分重複手法 2 はラックの収容を考慮して、無作為部分重複を実現しているためである。

図 14 に、冗長度と障害回復割合の関係を示す。図 14(a) に小規模データセンタを、図 14(b) に中規模データセンタをそれぞれ想定した環境の、8%のサーバに障害が発生した場合の評

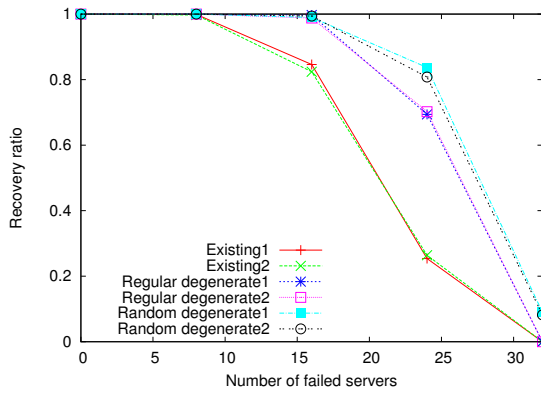


(a) 小規模データセンタを想定した環境の場合

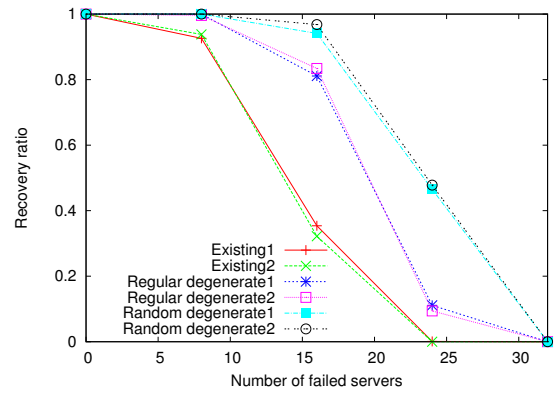


(b) 中規模データセンタを想定した環境の場合

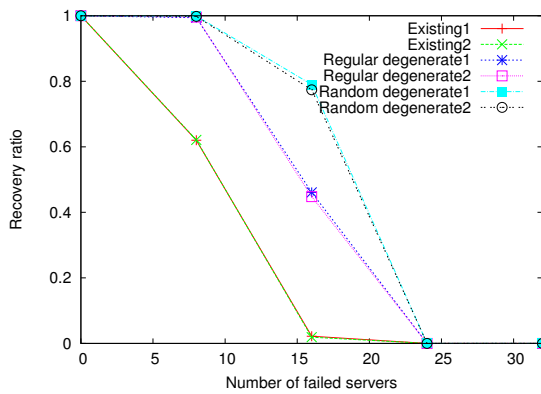
図 10: 均一な環境における物理サーバの故障台数と障害回復割合の関係



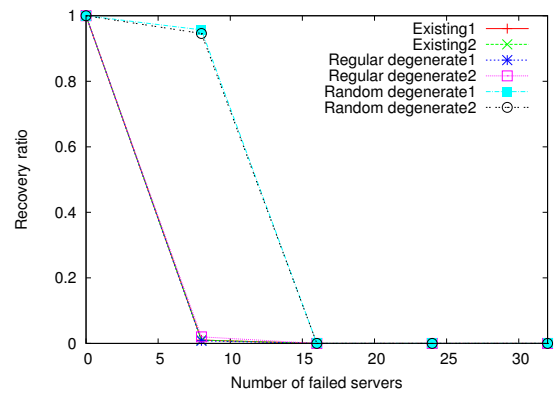
(a) 冗長度 100%の場合



(b) 冗長度 75%の場合

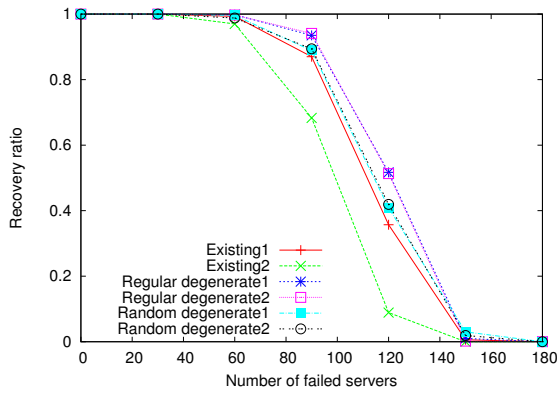


(c) 冗長度 50%の場合

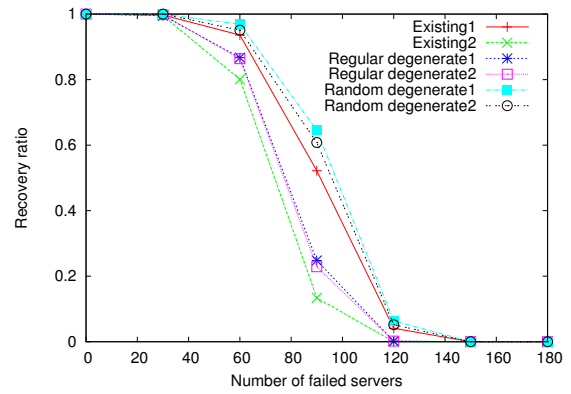


(d) 冗長度 25%の場合

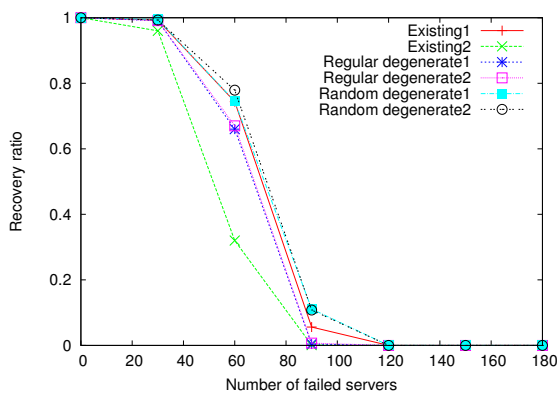
図 11: 均一な小規模データセンタを想定した環境における冗長度を変化させた時のサーバ故障数と障害回復割合の関係



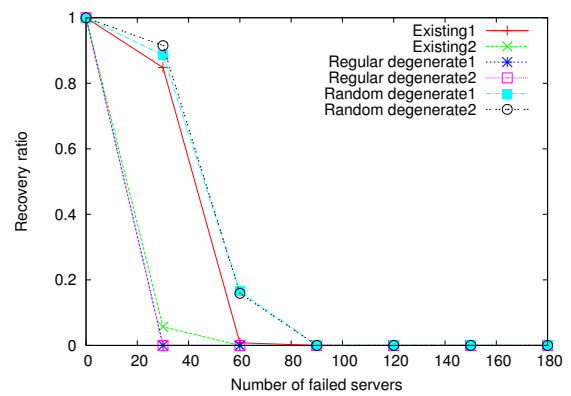
(a) 冗長度 100%の場合



(b) 冗長度 75%の場合

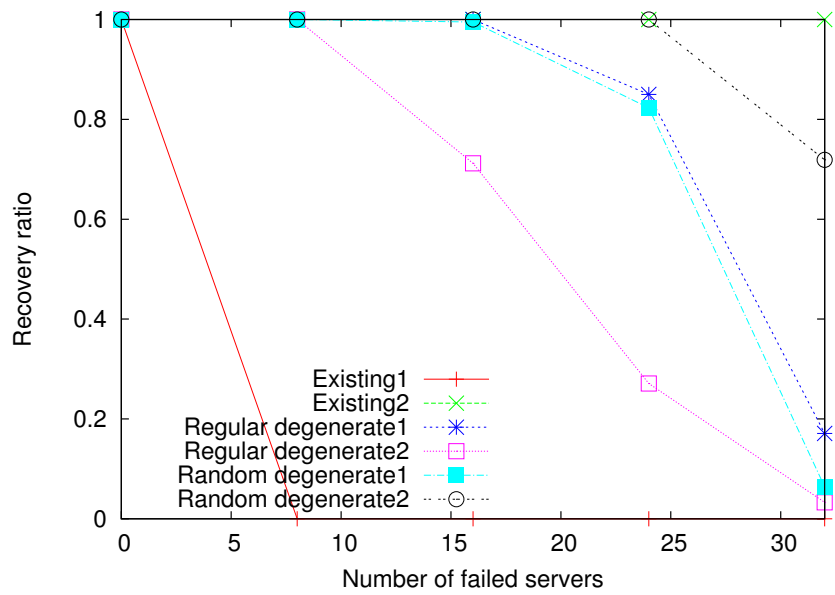


(c) 冗長度 50%の場合

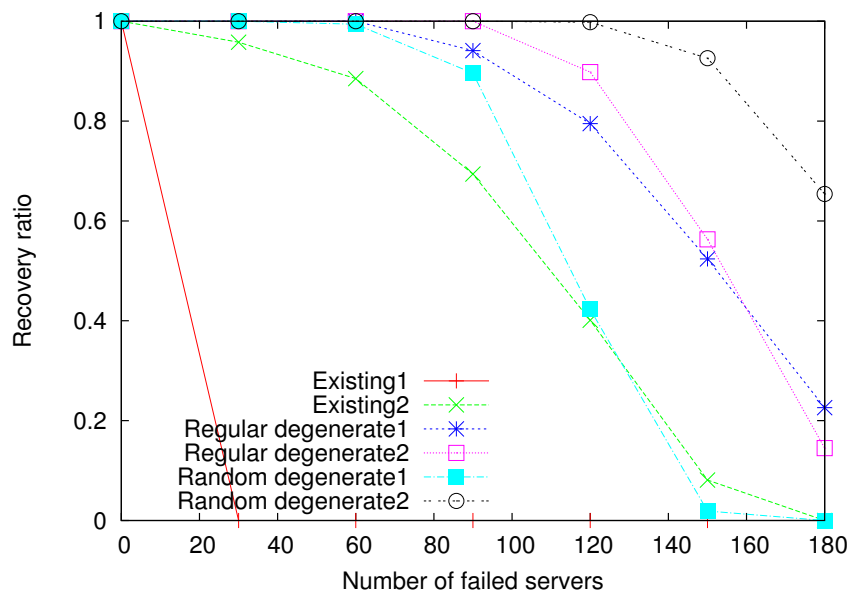


(d) 冗長度 25%の場合

図 12: 均一な中規模データセンタを想定した環境における冗長度を変化させた時のサーバ故障数と障害回復割合の関係



(a) 小規模データセンタを想定した環境の場合

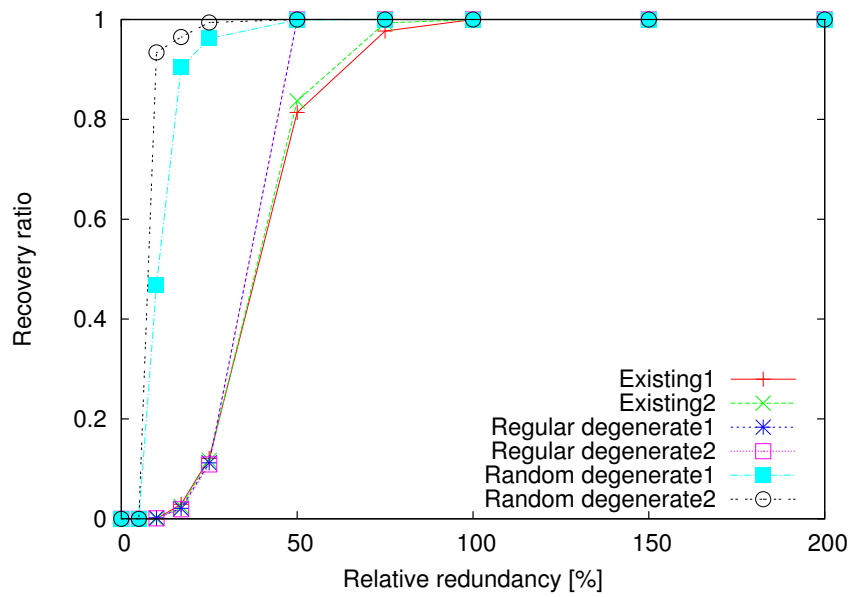


(b) 中規模データセンタを想定した環境の場合

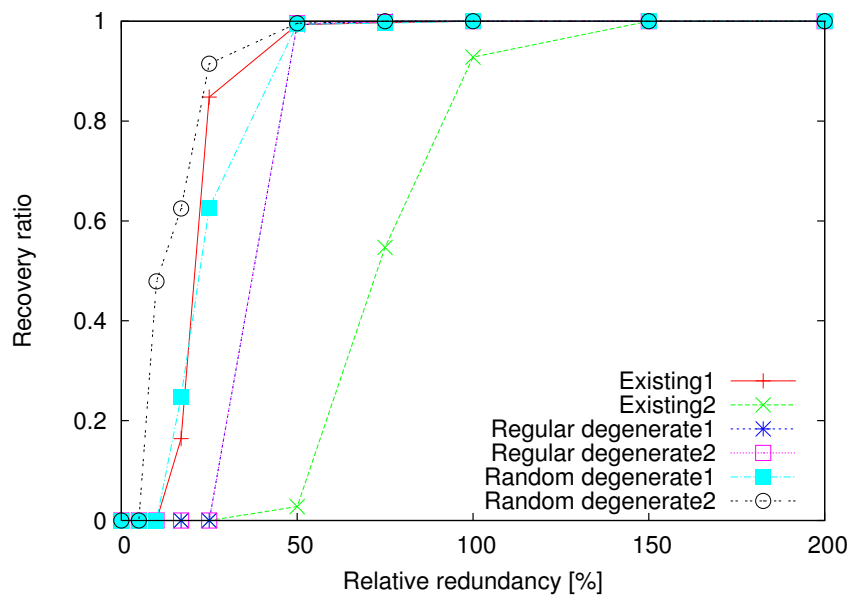
図 13: 均一な環境のラック障害における物理サーバの故障台数と障害回復割合の関係

価結果を表している。図より、無作為部分重複手法2が、他の仮想マシン配置手法と比べて、高い性能を示していることがわかる。また、障害回復割合を1.0とするために必要な冗長資源量に着目すると、小規模データセンタを想定した環境においては、無作為部分重複手法2を用いることで、既存手法1と2に比べて、冗長資源を75%削減できることがわかる。





(a) 小規模データセンタを想定した環境の場合



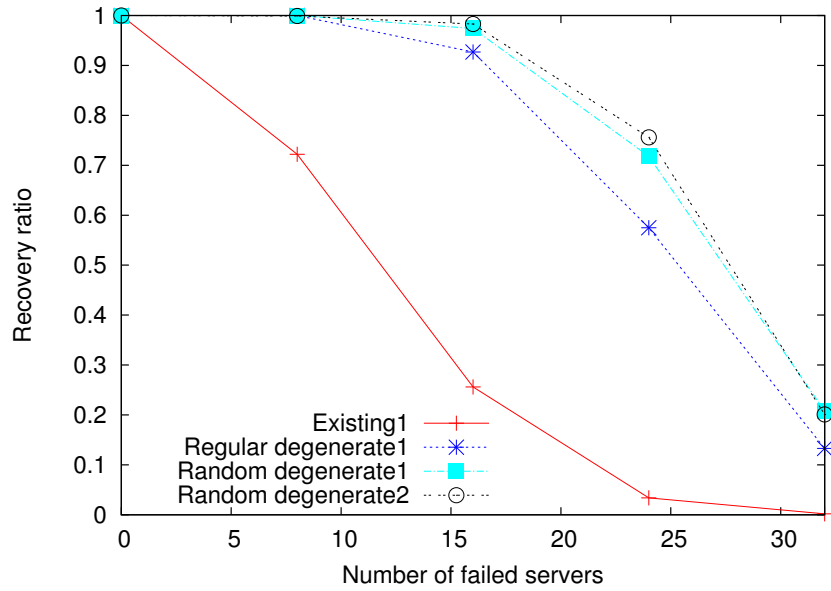
(b) 中規模データセンタを想定した環境の場合

図 14: 均一な環境における冗長度と障害回復割合の関係

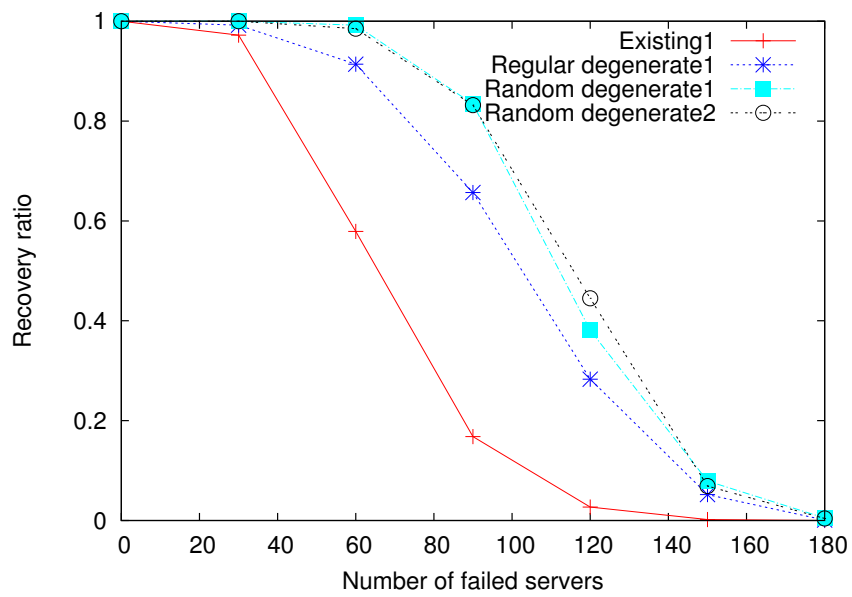
### 5.3 不均一な環境における提案手法の評価

図 15(a) に、不均一な小規模データセンタを想定した環境における、サーバ障害が発生した場合の、物理サーバの故障台数と障害回復割合の関係を示す。図 15(b) には、不均一な中規模データセンタを想定した環境における評価結果を示している。図より、無作為部分重複手法は、既存手法及び規則的部分重複手法に比べて、高い障害回復割合を示していることがわかる。また、既存手法 1 と規則的部分重複手法 1 は、均一な環境における評価に比べて、性能が低下している。既存手法 1 の性能が低下した理由は、ラックが収容するサーバ数が不均一であるために、各ラックの障害回復能力に差がでたためであると考えられる。また、規則的部分重複手法 1 の性能が低下した理由は、部分重複関係を構築する際に、均一な環境を仮定しているため、不均一な環境においては、VM のインストール数に偏りが生じるためであると考えられる。

次に、不均一な環境における、冗長度と障害回復割合の関係を、小規模データセンタを想定した環境に関しては図 16(a) に、中規模データセンタを想定した環境に関しては図 16(b) に示す。ただし、物理サーバの故障台数の割合は 8%としている。図より、無作為部分重複手法を用いることで、均一な環境と同様に、少ない冗長資源量でも、高い障害回復割合を維持できることがわかる。

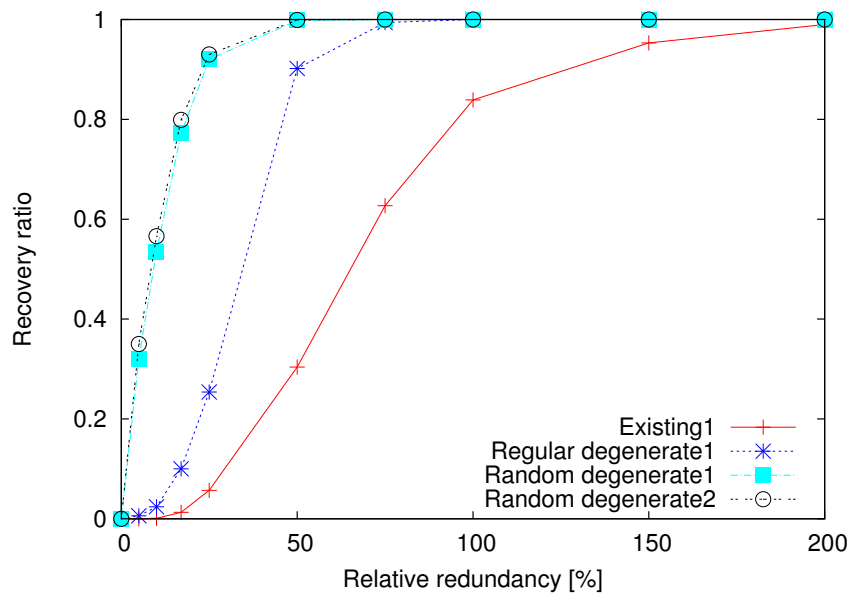


(a) 小規模データセンタを想定した環境の場合

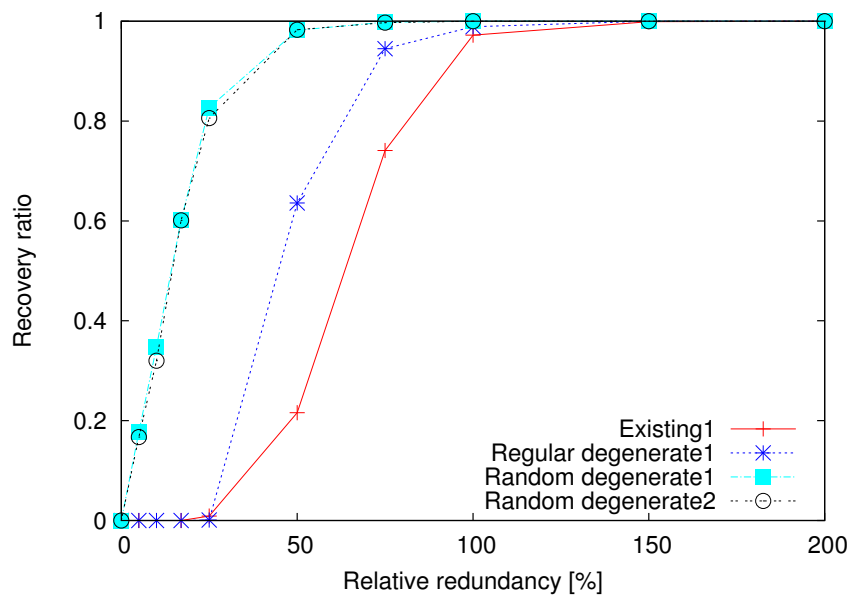


(b) 中規模データセンタを想定した環境の場合

図 15: 不均一な環境のサーバ障害における物理サーバの故障台数と障害回復割合の関係



(a) 小規模データセンタを想定した環境の場合



(b) 中規模データセンタを想定した環境の場合

図 16: 不均一な環境における冗長度と障害回復割合の関係

## 6 まとめと今後の課題

本報告では、生物ネットワークの縮退特性に着目し、データセンタの仮想マシン配置における、頑強かつ低消費電力な冗長化手法を提案した。提案手法では、縮退特性から得られるネットワーク化バッファリングの性質を利用し、高性能な物理サーバを用いることなく、限られた数の仮想マシンをインストールかつ実行することのできる物理サーバを準備し、それらの実行可能な仮想マシンの種類を部分的に重複させることによってネットワークを構築する。シミュレーション評価の結果、提案手法は従来の単純な冗長手法に比べて、環境変動や障害発生に対して、同じ障害復旧割合を示すために必要な冗長資源量を、最大で75%削減できることを示した。

今後の課題として、仮想マシン配置手法において、仮想マシンがどのテナントに属するかを考慮に入れることが挙げられる。また、4.2節において説明を行った障害回復手法は、集中制御を前提としているため、障害からの回復に時間がかかることが問題としてあげられる。そのため、各物理サーバが自律的に動作を行うような障害回復手法を検討し、障害からの回復にかかる時間の改善を図りたい。

## 謝辞

本報告を作成するにあたり、多くの方々の御指導、御協力を頂きました。指導教員である松岡茂登教授には、データセンタに関する基礎的な知識を御教授頂き、松岡研究室全体ミーティングにおいては、不確かな部分に鋭いご指摘を頂きました。ここに感謝の意を表します。そして、本報告を進めるにあたり、様々な御助言と御指導を頂き、研究者としての姿勢を学ばせて頂きました村田正幸教授には、大変感謝しております。また、この研究が特別研究報告として形になったのは、研究計画から報告書の作成に至るまで、終始丁寧かつ熱心な御指導を賜りました長谷川剛准教授のおかげです。心から感謝を申し上げます。最後に、研究生活において様々な知識やアドバイスを頂き、精神的にも支えて頂きました松岡研究室の皆様へ感謝いたします。本当にありがとうございました。

## 参考文献

- [1] M. Bari, R. Boutaba, R. Esteves, L. Granville, M. Podlesny, M. Rabbani, Q. Zhang, and M. Zhani, “Data center network virtualization: A survey,” *Communications Surveys Tutorials*, IEEE, vol. 15, no. 2, pp. 909–928, May 2013.
- [2] S. Sakr, A. Liu, D. Batista, and M. Alomari, “A survey of large scale data management approaches in cloud environments,” *Communications Surveys Tutorials*, IEEE, vol. 13, no. 3, pp. 311–336, Sept. 2011.
- [3] S. Bailmar, “Worldwide partner conference 2013 keynote.” available at <http://www.microsoft.com/en-us/news/speeches/2013/07-08wpcballmer.aspx>.
- [4] L. Ramos and R. Bianchini, “C-oracle: Predictive thermal management for data centers,” in *Proceedings of HPCA 2008*, pp. 111–122, Feb. 2008.
- [5] L. Wang, G. von Laszewski, J. Dayal, and T. Furlani, “Thermal aware workload scheduling with backfilling for green data centers,” in *Proceedings of Performance Computing and Communications Conference*, pp. 289–296, Dec. 2009.
- [6] B.-G. Chun, G. Iannaccone, G. Iannaccone, R. Katz, G. Lee, and L. Niccolini, “An energy case for hybrid datacenters,” *SIGOPS Operating System Review*, vol. 44, no. 1, pp. 76–80, Mar. 2010.
- [7] J. L. Berral, I. n. Goiri, R. Nou, F. Julià, J. Guitart, R. Gavaldà, and J. Torres, “Towards energy-aware scheduling in data centers using machine learning,” in *Proceedings of Energy-Efficient Computing and Networking*, pp. 215–224, Apr. 2010.
- [8] T. Mukherjee, A. Banerjee, G. Varsamopoulos, S. K. Gupta, and S. Rungta, “Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers,” *Computer Networks*, vol. 53, no. 17, pp. 2888–2904, Dec. 2009.
- [9] A. Beloglazov and R. Buyya, “Energy efficient resource management in virtualized cloud data centers,” in *Proceedings of Cluster, Cloud and Grid Computing*, pp. 826–831, May 2010.

- [10] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, “Elastic tree: Saving energy in data center networks,” in *Proceedings of Networked Systems Design and Implementation*, p. 17, Apr. 2010.
- [11] I. Cerutti, P. Raponi, N. Andriolli, P. Castoldi, and O. Liboiron-Ladouceur, “Designing energy-efficient data center networks using space-time optical interconnection architectures,” *Selected Topics in Quantum Electronics*, vol. 19, no. 2, p. 3700209, Mar. 2013.
- [12] U. Hoelzle and L. A. Barroso, *The datacenter as a computer: an introduction to the design of warehouse-scale machines*. Morgan and Claypool Publishers, 1st ed., 2009.
- [13] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, “The cost of a cloud: Research problems in data center networks,” *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 68–73, Dec. 2008.
- [14] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, and H. Liu, “Energy proportional datacenter networks,” *SIGARCH Comput. Archit. News*, vol. 38, no. 3, pp. 338–347, June 2010.
- [15] T. Benson, A. Akella, and D. A. Maltz, “Network traffic characteristics of data centers in the wild,” in *Proceedings of the IMC 2010*, pp. 267–280, Nov. 2010.
- [16] M. Chen, H. Zhang, Y.-Y. Su, X. Wang, G. Jiang, and K. Yoshihira, “Effective vm sizing in virtualized data centers,” in *Proceedings of Integrated Network Management (IM)*, pp. 594–601, May 2011.
- [17] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, “The nature of data center traffic: Measurements & analysis,” in *Proceedings of Internet Measurement Conference*, pp. 202–208, Nov. 2009.
- [18] S.-H. Wang, P. P.-W. Huang, C. H.-P. Wen, and L.-C. Wang, “Eqvmp: Energy-efficient and qos-aware virtual machine placement for software defined datacenter networks,” in *Proceedings of Information Networking*, pp. 220–225, Feb. 2014.
- [19] A. Barahasi and R. Albert, “Emergence of scaling in random networks,” *Science*, vol. 286, pp. 509–512, Oct. 1999.
- [20] D. Watts and S. Strogatz, “Collective dynamics of small-world networks,” *Nature*, vol. 393, no. 6684, pp. 440–442, June 1998.



- [21] S. Balasubramaniam, K. Leibnitz, D. Botvich, and M. Murata, “Biological principles for future Internet architecture design,” *IEEE Communications Magazine*, vol. 49, no. 7, pp. 44–52, July 2011.
- [22] M. Meisel, V. Pappas, and L. Zhanga, “A taxonomy of biologically inspired research in computer networking,” *Computer Networks*, vol. 54, no. 6, pp. 901–916, Apr. 2010.
- [23] A. Tero, S. Takagi, T. Saigusa, D. B. Kentaro Ito, M. Fricker, K. Yumiki, R. Kobayashi, and T. Nakagaki, “Rules for biologically inspired adaptive network design,” *Science*, vol. 327, no. 5964, pp. 439–442, Jan. 2010.
- [24] G. Tononi, O. Sporns, and G. Edelman, “Measures of degeneracy and redundancy in biological networks,” *Proceedings of the National Academy of Science*, vol. 96, no. 6, pp. 3257–3262, Mar. 1999.
- [25] J. Whitacre and A. Bender, “Degeneracy: A design principle for achieving robustness and evolvability,” *Journal of Theoretical Biology*, vol. 263, no. 1, pp. 143–153, Nov. 2010.
- [26] Z. Ma and A. Krings, “Bio-robustness and fault tolerance: A new perspective on reliable, survivable and evolvable network systems,” in *Proceedings of Aerospace Conference, 2008 IEEE*, pp. 1–20, Mar. 2008.
- [27] J. M. Whitacre, P. Rohlfshagen, A. Bender, and X. Yao, “The role of degenerate robustness in the evolvability of multi-agent systems in dynamic environments,” in *Proceedings of PPSN 2010*, pp. 284–293, Sept. 2010.
- [28] J. Whitacre, “Degeneracy: A link between evolvability, robustness and complexity in biological systems,” *Theoretical Biology and Medical Modelling*, vol. 7, no. 1, p. 6, Feb. 2010.
- [29] J. Whitacre and A. Bender, “Networked buffering: A basic mechanism for distributed robustness in complex adaptive systems,” *Theoretical Biology and Medical Modelling*, vol. 7, no. 20, pp. 1–20, Mar. 2010.
- [30] G. M. Edelman and J. A. Gally, “Degeneracy and complexity in biological systems,” *Proceedings of the National Academy of Sciences*, vol. 98, no. 24, pp. 13763–13768, Nov. 2001.

- [31] B. Guo, C. A. Styles, Q. Feng, and G. R. Fink, “A *saccharomyces* gene family involved in invasive growth, cell-cell adhesion and mating,” *Proceedings of National Academy of Sciences*, vol. 97, no. 22, pp. 12158–12163, Oct. 2000.
- [32] U. Sauer, F. Canonaco, S. Heri, A. Perrenoud, and E. Fischer, “The soluble and membrane-bound transhydrogenases *udha* and *pntab* have divergent functions in nadph metabolism of *escherichia coli*,” *Journal of Biological Chemistry*, vol. 279, no. 8, pp. 6613–6619, Dec. 2003.
- [33] A. Wagner, “The role of population size, pleiotropy and fitness effects of mutations in the evolution of overlapping gene functions,” *Genetics*, vol. 154, no. 3, pp. 1389–1401, Mar. 2000.
- [34] Y. Masatoshi, T. Kana, K. Rikako, I. Takuro, and I. Tadashi, “Striate cortical lesions affect deliberate decision and control of saccade: Implication for blindsight,” *The Journal of Neuroscience*, vol. 28, no. 42, pp. 10517–10530, Oct. 2008.
- [35] 長谷川 剛, 村田 正幸, “生物ネットワークの縮重性・冗長性に基づく情報ネットワーク設計に関する一検討,” *電子情報通信学会技術研究報告*, vol. 112, no. 119, pp. 29–34, July 2012.
- [36] 岡崎 拓郎, 長谷川 剛, 村田 正幸, “生物ネットワークの縮退特性に基づく頑強性を有する情報ネットワークシステムの冗長化手法,” *電子情報通信学会技術研究報告*, vol. 112, no. 463, pp. 595–600, Mar. 2013.
- [37] IBM Corp., “Live guest relocation.” available at <http://www.vm.ibm.com/perf/reports/zvm/html/6201gr.html>.
- [38] 長谷川 剛, 村田 正幸, “生物ネットワークの縮退特性に基づくシステム冗長化方式における自律的障害回復手法,” *電子情報通信学会技術研究報告*, vol. 113, no. 4, pp. 31–36, Apr. 2013.