

令和 5 年度
修士学位論文

クラウドコンピューティングにおける
可用性と消費電力を考慮した
仮想マシン配置手法

Virtual Machine Allocation Method Considering
Availability and Power Consumption in Cloud
Computing

1265096 菊森 剛

指導教員 横山和俊

令和 6 年 2 月 28 日

高知工科大学大学院 工学研究科 基盤工学専攻
情報学コース

要旨

クラウドコンピューティングにおける 可用性と消費電力を考慮した 仮想マシン配置手法

菊森 剛

近年、クラウドコンピューティングが急激に普及している。クラウド事業者は、膨大な計算資源を用意することでサービスレベルを保っているが、膨大な計算資源を用意するとデータセンタでの消費電力が増大する問題がある。このため、消費電力が少なく、かつサービスの品質を保つための仮想マシン配置手法が提案されている。

本研究では、サービスの品質を保つために可用性を考慮した仮想マシン配置手法を提案する。提案手法では、デュプレックスシステムを前提とし、可用性を要件を見たす現用系と予備系の配置の組合せを求め、消費電力の推定を行う。消費電力は、CPU 使用率に依存するため、それらの相関関係を定義し消費電力の利用した。

シミュレーションによる評価では、FirstFit と比較し最大 1.4 倍の消費電力が増加するものの、障害復旧時間を 45%削減することが確認できた。

キーワード クラウドコンピューティング, 消費電力, 可用性

Abstract

Virtual Machine Allocation Method Considering Availability and Power Consumption in Cloud Computing

Go Kikumori

In recent years, cloud computing has been spreading rapidly. Cloud service providers maintain their service level by providing a huge amount of computing resources, but the problem is that providing a huge amount of computing resources increases power consumption at the data center. For this reason, virtual machine allocation methods that consume less power and maintain quality of service are being studied.

In this study, we propose a virtual machine placement method that considers availability in order to maintain quality of service. The proposed method assumes a duplex system and estimates power consumption by finding a combination of primary and standby system allocation that satisfy the availability requirements. Since power consumption depends on CPU utilization, those dependencies are used to estimate power consumption.

Simulation evaluations confirmed that although power consumption increased by up to 1.4 times compared to FirstFit, the failure recovery time could be reduced by 45%

key words Cloud Computing, Power Consumption, Availability

目次

第 1 章	はじめに	1
第 2 章	関連研究	3
2.1	仮想マシンの配置手法に関する研究	3
2.2	消費電力モデルに関する研究	3
第 3 章	関連技術	5
3.1	クラウドコンピューティング	5
3.2	デュプレックスシステム	5
第 4 章	システムモデル	7
4.1	可用性モデル	7
4.2	消費電力モデル	7
4.3	システム構成	7
第 5 章	提案手法	10
5.1	目的関数と制約条件	10
5.1.1	制約条件 1	10
5.1.2	制約条件 2	11
5.1.3	制約条件 3	11
5.2	配置決定アルゴリズム	12
5.3	N 対 1 構成への拡張	15
5.3.1	複数の現用系の取り扱い	15
5.3.2	制約条件の拡張	15
第 6 章	評価	17

目次

6.1	評価方法	17
6.1.1	評価指標	17
6.1.2	比較手法	18
6.1.3	パラメータ	18
6.2	結果と考察	22
6.2.1	1対1構成における結果	22
6.2.2	障害復旧時間における追加調査	24
6.2.3	3対1構成における結果	26
6.2.4	提案手法における計算時間の結果	28
6.2.5	消費電力モデルの違いにおける結果	28
第7章	おわりに	30
7.1	まとめ	30
7.2	今後の課題	30
	謝辞	32
	参考文献	33
	付録 A	34

目次

4.1	クラウドコンピューティング環境の構成図	8
4.2	配置の具体的	9
5.1	制約条件1のイメージ図	11
5.2	制約条件2のイメージ図	12
5.3	制約条件3のイメージ図	12
5.4	列挙の例	13
5.5	抽出の例	14
5.6	N対1構成における制約条件2の例	16
6.1	線形な相関を持つモデル	20
6.2	60%辺りで収束する相関を持つモデル	20
6.3	80%辺りで収束する相関を持つモデル	21
6.4	1対1構成の消費電力	23
6.5	1対1構成の障害復旧時間(単位時間)	23
6.6	仮想マシンの割合	24
6.7	0%の割合	25
6.8	3対1構成の消費電力	27
6.9	3対1構成の障害復旧時間	27
6.10	N対1構成の計算時間	28
6.11	消費電力モデルの違いによる消費電力	29
A.1	1対1構成の消費電力量	34
A.2	1対1構成の障害復旧時間	35
A.3	3対1構成の消費電力量	35

図目次

A.4 3対1構成の障害復旧時間	36
----------------------------	----

表目次

4.1	物理マシンが持つ情報	8
4.2	仮想マシンが持つ情報	9
6.1	シミュレーションパラメータ	19

第 1 章

はじめに

近年、クラウドコンピューティングが急激に普及している。総務省が発行する令和 5 年版情報通信白書 [1] によると、世界のパブリッククラウドサービス市場は年々拡大しており、2021 年には、45 兆 621 億円と前年比 28.6% 増となっている。特に、Microsoft Azure や、Amazon Web Services がシェアの 3 割を超えており、今後も拡大していくと予測されている。

需要が拡大していく中、クラウド事業者はユーザとの間で Service Level Agreement (以降、SLA と略す) と呼ばれる規約を作成することで、サービスの安定的な供給をしている。SLA とは、事業者がユーザに提供するサービスのレベルを指し示す契約である。特に、稼働時間や応答時間、解決時間などの指標を規定している。例として、Amazon Web Services の SLA を抜粋する [2]。AWS の例では、99.5% の稼働率でユーザが利用できるよう目指すとしている。

Instance-Level SLA For each individual Amazon EC2 instance (“ Single EC2 Instance ”), AWS will use commercially reasonable efforts to make the Single EC2 Instance available with an Instance-Level Uptime Percentage of at least 99.5%, in each case during any monthly billing cycle (the “ Instance-Level SLA ”).

クラウド事業者は、この SLA を遵守するため、膨大な計算資源を用意することでサービスの品質を保っているが、膨大な計算資源を用意するとデータセンタでの消費電力が増大する問題がある。

主なクラウドサービスでは、複数の計算機を用いた大規模な仮想化環境を使用すること

で、エンドユーザのリソース要求に応じた計算資源を仮想マシンとして提供している。仮想マシンをどの計算機に配置するかは、クラウド事業者によってポリシーが定められており、計算機のリソース使用量や仮想マシンの要求リソース量によって日々、配置場所の変更（マイグレーション）が行われている。

そのため、仮想マシンの配置ポリシーが検討されている。Mollamotalebi ら [3] は、クラウドコンピューティングでのサービスレベルと消費電力量を考慮した仮想マシンの動的配置手法を提案している。この研究では、障害が発生した際のサービスレベルの保証を考慮していない。そこで、本稿では、可用性と消費電力の両方を考慮した仮想マシン配置手法を提案する。可用性とは、システムが継続して稼働できる性能のことであり、障害が発生した際にシステムが停止している時間を短くすることが求められる。このため、デュプレックスシステムを用いたクラウドコンピューティングの仮想マシン配置手法を検討する。提案手法では、サービスを提供している現用系の仮想マシンのほかに予備系の仮想マシンを用意し、障害が発生した際に迅速に復旧することを可能にすると共に、消費電力量が少なくなるように仮想マシンを配置することを可能にする。

第 2 章

関連研究

2.1 仮想マシンの配置手法に関する研究

クラウドコンピューティングでの、仮想マシンの配置手法は様々なものが検討されている。文献 [3] では、消費電力と SLA 違反率（以下、SLAV）を考慮し、動的な負荷変動に対応した仮想マシンの配置アルゴリズムを提案している。消費電力の削減に対応した配置アルゴリズムと SLAV の削減に対応した配置アルゴリズムの 2 種類を用意し、その 2 つを選択的に適応させる提案手法を提案している。具体的には、一定時間ごとに SLAV の数値を確認し閾値を超えていたら、SLAV の削減に対応したアルゴリズムに変更し、それ以外の場合は消費電力の削減に対応したアルゴリズムに変更する手法である。結果として、消費電力の削減に対応した配置アルゴリズムと SLAV の削減に対応したアルゴリズムに比べて、互いに良い結果となった。しかしながら、SLA 違反が発生した際の可用性について考慮されていない。そのため、本研究では、消費電力と可用性を考慮した手法の提案を行う。

2.2 消費電力モデルに関する研究

物理マシンの消費電力については、一般に CPU 使用率に依存することが知られている。実際に、文献 [4] では、サーバにおけるリソース使用量と消費電力の相関関係を調査している。結果として、CPU 使用率と消費電力は強い相関があると結論づけられている。

一方、文献 [5] では、CPU 使用率が上昇するにつれて、消費電力が収束という結果もある。これらの文献より、消費電力は CPU 使用率に依存するが、使用する CPU によって相関

2.2 消費電力モデルに関する研究

関係が異なる。

そのため本研究では、事前に CPU 使用率と消費電力の相関関係を定義し、物理マシンの消費電力とする。

第 3 章

関連技術

3.1 クラウドコンピューティング

クラウドコンピューティングとは、インターネット上で、web サーバやデータベース、ストレージなどの様々な IT の計算資源を提供するサービスの総称である。ユーザが自ら物理マシンを用意する必要がなく、必要な時に必要な計算資源をインターネット上から使用することができる。

このようなクラウドコンピューティングサービスは、仮想化技術を用いて実現されている。仮想化とは、サーバやストレージ、ネットワークなどの物理的な構成を、仮想的に表現して使用することのできる技術です。仮想化ソフトウェアを使用して、1 台の物理マシンに複数の仮想マシンを実行することで、物理的なリソースを効率的に使用できる。

本研究では、このような仮想化技術を用いたクラウドコンピューティング環境を想定している。

3.2 デュプレックスシステム

デュプレックスシステムとは、システムの可用性を考慮した障害に強いシステムモデルである。可用性とは、システムやサービスが利用できる状態を表し、障害が発生することが少なく安定して利用することができるサービスを可用性が高いという。

具体的には、1 つのサービスや処理に対して、複数の仮想マシンを用意し、計算資源を冗長化しておくことで、可用性を高める。このモデルでは、仮想マシンの 1 つを障害が発生し

3.2 デュプレックスシステム

た際の予備機として、待機させておく。以下では、デュプレックスシステムにおいて、主に処理を実行している仮想マシンを現用系、予備機として待機させている仮想マシンを予備系と呼ぶ。一般に、デュプレックスシステムでは、現用系を複数個の仮想マシンに対し、予備系を1個の仮想マシンで運用する。これを本稿では、N対1構成と称し説明する。

また、デュプレックスシステムにおいて、予備系の待機の方法が主に3つあり、特徴を以下に挙げる。

ホットスタンバイ

予備系を常時起動した状態で現用系とデータを同期しておき、障害が発生した際に早急に復旧できるスタンバイ方式である。予備系のリソース要求量は、3つの中で一番高い。

ウォームスタンバイ

予備系を低負荷な状態で起動しておき、障害が発生した際に迅速に復旧できるスタンバイ方式である。予備系のリソース要求量は、少ない。

コールドスタンバイ

予備系が実行できるリソースを確保し、サスペンドまたはシャットダウン状態で配置しておき、障害が発生した際にサスペンドやシャットダウン状態から起動して復旧するスタンバイ方式である。

デュプレックスシステムのメリット・デメリットを挙げる。メリットとしては、まず第一に可用性を高めることができることである。デメリットは、1つの処理に複数の仮想マシンを使用するため、計算機の消費電力が増大することである。

第 4 章

システムモデル

本研究で，想定するクラウド環境を説明する．

4.1 可用性モデル

本研究では，可用性モデルとして，3.2 節で述べたデュプレックスシステムを想定する．

4.2 消費電力モデル

一般に，物理マシンの消費電力は，CPU 使用率に依存することが知られている．そのため，CPU 使用率と消費電力の相関関係について，事前に計測し，わかっていることを想定する．

本稿では，CPU 使用率と消費電力の相関関係について，数種類のモデルを検討する．

4.3 システム構成

図 4.1 では，本稿で想定するクラウドコンピューティング環境の様子を示した図である．まず，物理マシンを `host1`，`host2` … と表し，物理マシンの数を `HOSTS` とする．

4.3 システム構成

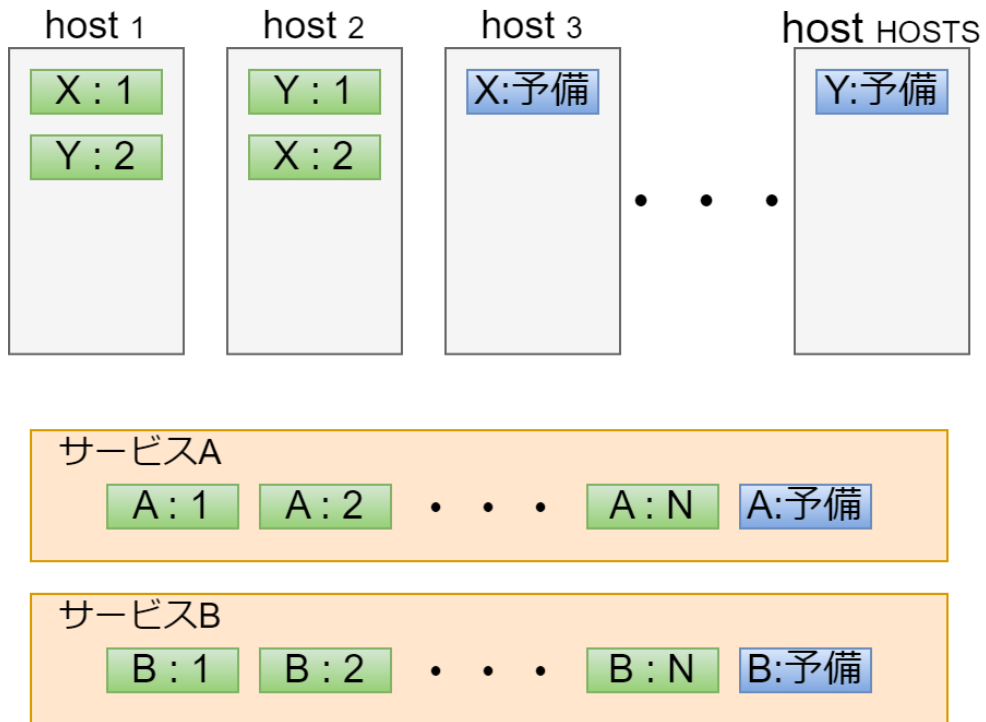


図 4.1 クラウドコンピューティング環境の構成図

次に、物理マシンに配置する仮想マシンについて説明する。N 対 1 構成では、1 つの処理を複数の仮想マシンに分散させるため、本稿ではその処理の 1 つをサービスと名付け、サービス A、サービス B … と表し、サービスの数を SERVICES とする。また、サービス 1 つに対して、現用系の仮想マシンが N 台、予備系の仮想マシンが 1 台あり、A:1, A:2, …, A: 予備と表す。仮想マシンの総数は、以下の式 4.1 で表される。

$$\text{VMS} = \text{SERVICES} \times (\text{N} + 1) \quad (4.1)$$

物理マシンと仮想マシンは、それぞれ表 4.1, 4.2 の情報を持つ。

表 4.1 物理マシンが持つ情報

情報	説明
CPU 使用率	物理マシンの CPU 使用率
消費電力モデル	4.2 節で述べた、消費電力モデル
配置マシン	現在、配置されている仮想マシンの情報

4.3 システム構成

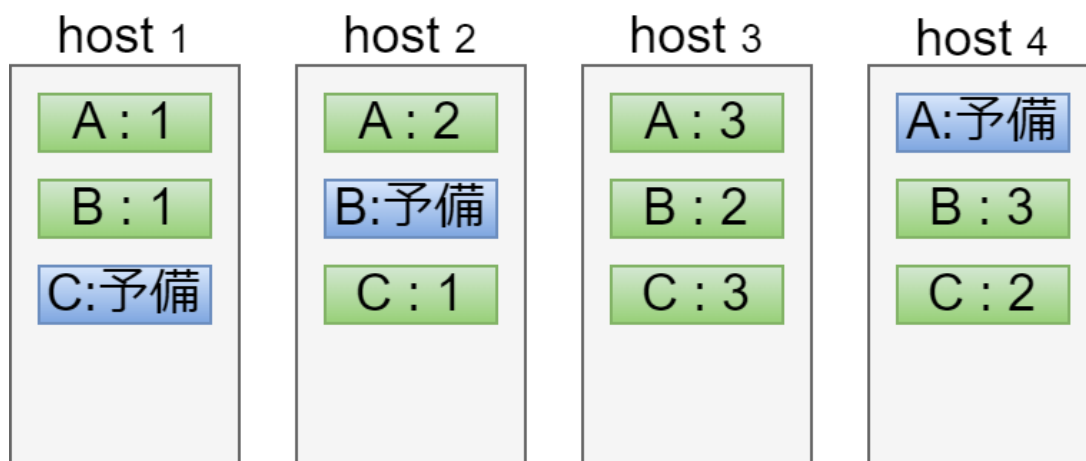


図 4.2 配置の具体的

表 4.2 仮想マシンが持つ情報

情報	説明
種別	現用系か予備系
CPU 使用率	仮想マシンの CPU 使用率, 物理マシンとは別

図 4.2 は、以上の構成を踏まえた、配置例である。物理マシンの数を HOSTS = 4, サービスの数を SERVICES = 3, N 対 1 構成の N を N = 3 とした。

第 5 章

提案手法

本稿では、現用系と予備系を同時に配置する組合せを全通り列挙し、最も良い組を決定する配置手法を提案する。

5.1 目的関数と制約条件

本手法では、目的関数を以下の式 5.1 とする。

$$\text{Min}(\sum_{k=1}^{\text{HOSTS}} P(\text{host}_k)) \quad (5.1)$$

次に、制約条件を 3 つ設定し、それらについて説明する。

制約条件 1 サービスの現用系と予備系を同じ配置にしない

制約条件 2 違うサービスの現用系と予備系の配置と同じ配置にしない

制約条件 3 配置後のホストの CPU 使用率は X%以下とする

5.1.1 制約条件 1

制約条件 1 は、サービスの現用系と予備系を同じ配置にしないことである。この制約条件は、デュプレックスシステムにおいて、一般的な条件である。イメージを図 5.1 に示す。図は、サービス A、B を配置した様子を表している。この配置では、host1 で障害が発生した場合、サービス A は host2 にある予備系に処理が移るが、サービス B は予備系も同時に障害に合うため、意味をなさない。そのため、サービス A のように、現用系と予備系を同じホ

5.1 目的関数と制約条件

ストに配置しない。

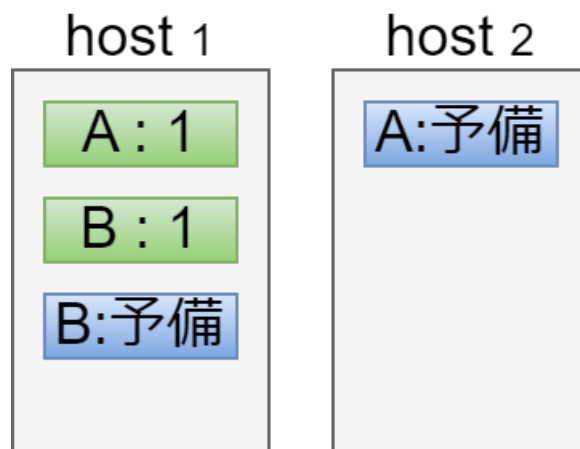


図 5.1 制約条件 1 のイメージ図

5.1.2 制約条件 2

制約条件 2 は、違うサービスの現用系と予備系の配置と同じにしないことである。この制約条件は、可用性を考慮した条件である。イメージを図 5.2 に示す。デュプレックスシステムでは、障害が発生した際に、予備系に処理が移動することで可用性を高くしている。図のサービス A、B のような状態の際に、host1 で障害が発生した場合、サービス A と B に処理が移動するため、host2 のリソース要求が高くなることが予測される。さらには、host2 まで処理落ちしてしまう危険性もある。そのため、サービス A、B のように、現用系と予備系が重なる配置はしない。

5.1.3 制約条件 3

制約条件 3 は、仮想マシンを配置した host の CPU 使用率は X%以下とすることである。この制約条件は、物理マシンのリソース逼迫を低減するための条件である。host の CPU 使用率が高くなると物理マシンが処理落ちする危険性があるため、閾値 X を設定し、それ以上の CPU 使用率にならないようにする。イメージを図 5.3 に示す。図では、host1 に 4 台の仮想マシンが配置されており、90%を使用している。閾値 X を $X = 85$ とすると、閾値を

5.2 配置決定アルゴリズム

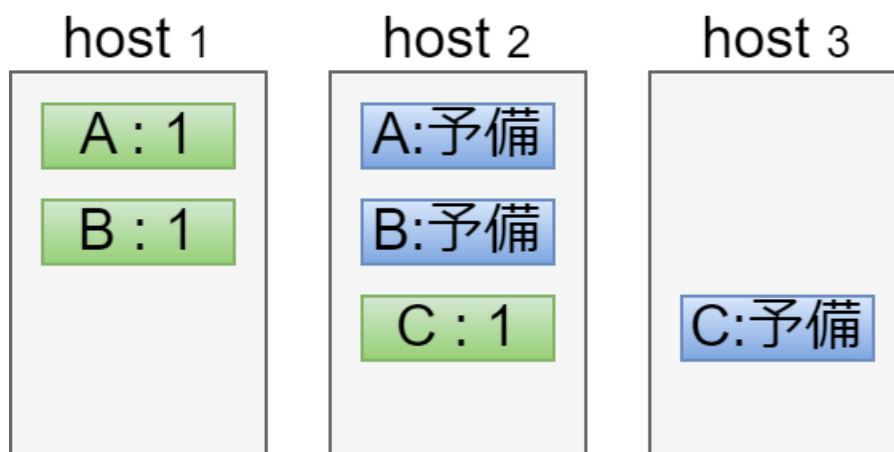


図 5.2 制約条件 2 のイメージ図

を超えていることがわかる。

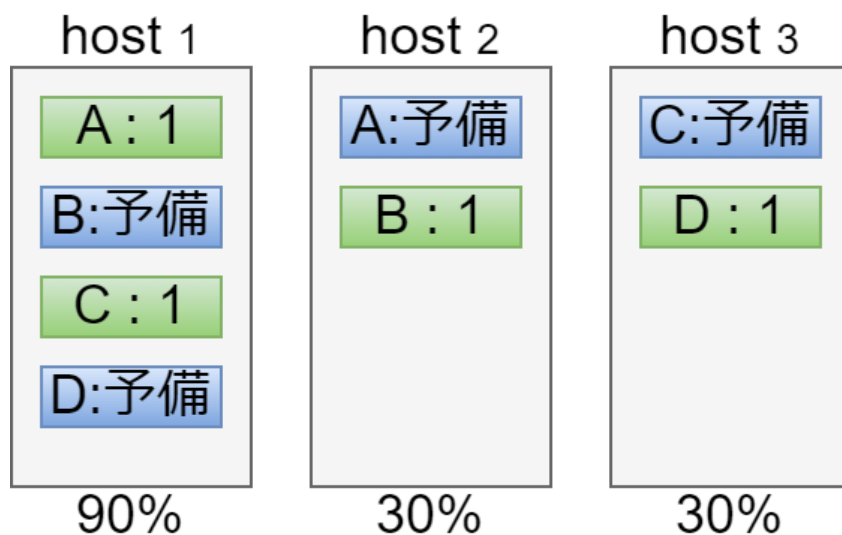


図 5.3 制約条件 3 のイメージ図

5.2 配置決定アルゴリズム

配置を決定するアルゴリズムの手順を以下に示す。

手順 1 その時点でのクラウド環境に配置できる組合せを全て列挙

はじめに、現在のクラウド環境で配置できるすべての組合せを列挙する。ただし、

5.2 配置決定アルゴリズム

制約条件 1 は他のサービスに関係なく決定できるため、この際に考慮して列挙する。

図 5.4 では、1 対 1 構成において、HOSTS=4 の例を列挙した例である。

	host1	host2	host3	host4
組1	A:1	A:予備		
組2	A:1		A:予備	
組3	A:1			A:予備
組4	A:予備	A:1		
組5		A:1	A:予備	
組6		A:1		A:予備
組7	A:予備		A:1	
組8		A:予備	A:1	
組9			A:1	A:予備
組10	A:予備			A:1
組11		A:予備		A:1
組12			A:予備	A:1

図 5.4 列挙の例

この時、全ての組の総数を一般化すると以下の式 5.2 である。物理マシンの数を HOSTS, C は組合せ (Combination) を表す。

$$\text{HOSTS} C_2 \times 2 \quad (5.2)$$

手順 2 列挙した組から、制約条件を満たす組を抽出

手順 1 で列挙した組について、制約条件を満たす組を抽出する。図 5.5 は、その抽出の様子を示す。図 5.4 で、列挙した組のうち組 1~6 までを例として説明する。まず、組 1~3 については、E:1 を host1 に配置する組であるが、host1 は既に 85%の

5.2 配置決定アルゴリズム

CPU 使用率となっており，閾値 X を超えるため制約条件 3 を満たさない．次に，組 5 については，サービス C の配置と同じ配置になるため，制約条件 2 を満たさない．これらより，組 4，6 が制約条件を全て満たし，配置の可能性がある．同様に，組 7～12 については，組 7，8，9 が host3 で CPU 使用率が閾値を超えるため制約条件 3 を満たさない．従って，組 10～12 が制約条件を全て満たし，配置の可能性がある．

	host 1	host 2	host 3	host 4
	<div style="border: 1px solid black; padding: 5px;"> <div style="background-color: #90EE90; padding: 2px;">A : 1</div> <div style="background-color: #90EE90; padding: 2px;">B : 1</div> <div style="background-color: #ADD8E6; padding: 2px;">D:予備</div> <div style="text-align: center;">85%</div> </div>	<div style="border: 1px solid black; padding: 5px;"> <div style="background-color: #ADD8E6; padding: 2px;">A:予備</div> <div style="background-color: #90EE90; padding: 2px;">C : 1</div> <div style="text-align: center;">50%</div> </div>	<div style="border: 1px solid black; padding: 5px;"> <div style="background-color: #ADD8E6; padding: 2px;">B:予備</div> <div style="background-color: #ADD8E6; padding: 2px;">C:予備</div> <div style="background-color: #90EE90; padding: 2px;">D : 1</div> <div style="text-align: center;">65%</div> </div>	<div style="border: 1px solid black; padding: 5px;"> <div style="text-align: center;">0%</div> </div>
組1	<div style="background-color: #90EE90; padding: 2px;">E : 1</div>	<div style="background-color: #ADD8E6; padding: 2px;">E:予備</div>		
組2	<div style="background-color: #90EE90; padding: 2px;">E : 1</div>		<div style="background-color: #ADD8E6; padding: 2px;">E:予備</div>	
組3	<div style="background-color: #90EE90; padding: 2px;">E : 1</div>			<div style="background-color: #ADD8E6; padding: 2px;">E:予備</div>
組4	<div style="background-color: #ADD8E6; padding: 2px;">E:予備</div>	<div style="background-color: #90EE90; padding: 2px;">E : 1</div>		
組5		<div style="background-color: #90EE90; padding: 2px;">E : 1</div>	<div style="background-color: #ADD8E6; padding: 2px;">E:予備</div>	
組6		<div style="background-color: #90EE90; padding: 2px;">E : 1</div>		<div style="background-color: #ADD8E6; padding: 2px;">E:予備</div>

図 5.5 抽出の例

手順 3 制約条件を満たした組全てにおいて推定消費電力を計算

手順 2 で抽出された組の全ての推定消費電力を計算する．推定消費電力は，以下のよう
に計算される．

- (a) その組が配置された後の CPU 使用率を取得する
- (b) 各物理マシンの CPU 使用率に対応した消費電力を取得する
- (c) すべての物理マシンの消費電力を合計した値を推定消費電力とする

手順 4 推定消費電力が最も小さい組の配置に決定

5.3 N 対 1 構成への拡張

手順 3 で計算された推定消費電力が最も小さい組の配置に決定する。

5.3 N 対 1 構成への拡張

前節では、1 対 1 構成のアルゴリズムについて説明してきたが、デュプレックスシステムにおいては N 対 1 構成への拡張が求められる。以降では、1 対 1 構成と N 対 1 構成の違いを重点的に説明する。

5.3.1 複数の現用系の取り扱い

まず、N 対 1 構成において最も違う点として挙げられるのは、現用系同士の配置である。複数の現用系について、以下の通りの仕様とする。

- 1 つのサービスの現用系は、どれも同様の CPU 使用率を持ち、区別はしない。
- 1 つのサービスの現用系は、耐障害性を高めるため、同じ host に配置しない。

まず、一つ目は、現用系は区別しない。N 対 1 構成では、サービスの処理を N 台の仮想マシンでなるべく均等に分散して、処理をしていることが一般的である。そのため、本研究では N 台の現用系仮想マシンの CPU 使用率は、同じと仮定し区別しない。

二つ目は、現用系は同じ配置にしない。N 対 1 構成であれば、現用系の仮想マシンは N 台あるが、耐障害性や可用性を考えると、1 台の物理マシンに同じサービスの仮想マシンが複数台配置されていることは、望ましくない。従って、本研究ではこれらはすべて別の物理マシン上で稼働するようにする。また、N 対 1 構成の場合、物理マシンは最低でも $N + 1$ 台必要になる。

5.3.2 制約条件の拡張

制約条件についても、以下のように N 対 1 構成に拡張する。

制約条件 1

5.3 N 対 1 構成への拡張

制約条件 1 については、特に拡張する必要はないが、N 台の現用系すべてと同じ配置にならないようにする。

制約条件 2

制約条件 2 については、N 台の現用系すべての配置パターンを確認する。図 5.6 は、N 対 1 構成において、制約条件 2 を満たさない例である。サービス A の配置に対して、サービス B, C, D の配置は、制約条件を満たさないとする。

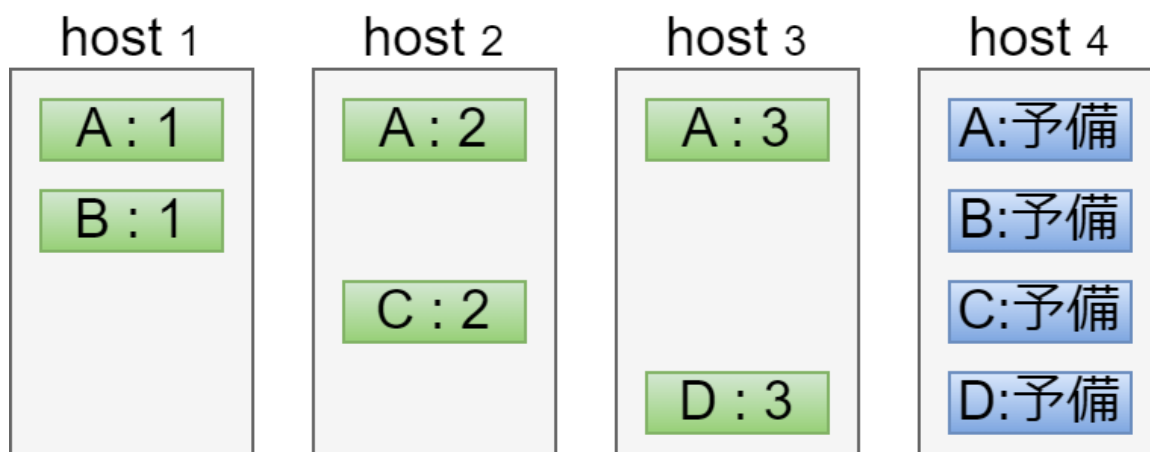


図 5.6 N 対 1 構成における制約条件 2 の例

制約条件 3

制約条件 3 については、特に拡張する必要はない。

第 6 章

評価

提案する配置手法の有効性を確認するため、シミュレーションにより評価を行う。以下、評価手法と結果を示す。

6.1 評価方法

提案する配置手法および、比較手法の配置アルゴリズムを実装し、シミュレーションを行う。シミュレーターでは、物理マシンと仮想マシンを用意し、物理マシンのリソース容量を超えないように仮想マシンを配置し、疑似的に一定時間運用する。このシミュレーションより、クラウドコンピューティング環境全体の消費電力や可用性を検証する。さらに、シミュレーション中に 1 台の物理マシンに障害を発生させ、その際の可用性も検討する。

6.1.1 評価指標

評価指標には、以下の指標を用いる。

1. 消費電力
2. 障害復旧時間
3. 提案手法における配置決定までの計算時間

本研究では、消費電力と可用性に着目しているため、以上の指標を用いる。消費電力は、すべての物理マシンの消費電力の合計で表され、以下の式 6.1 となる。

6.1 評価方法

$$\text{Power} = \sum_{\text{time}} \sum_{\text{HOSTS}} P(\text{host}_k) \quad (6.1)$$

障害復旧時間は、以下のように定義する。

- 復旧時間として、現用系 1 台につき一定時間の起動時間の合計として定義
- 予備系は、障害が発生しても後回しとして、復旧時間は 0 秒

また、提案手法では、N 対 1 構成において、配置決定にかかる計算時間が増大することが予測されるため、それぞれの計算時間を比較、検討する。

6.1.2 比較手法

比較手法には、提案手法のほかに以下の 2 つを用いる。

1. FirstFit
2. FirstFit (可用性考慮有)

FirstFit は、ビンパッキング問題におけるアルゴリズムの一つで、物理マシンのリソース容量を超えないように、仮想マシンを配置する点でビンパッキング問題と似ているため、ビンパッキング問題で基本的なアルゴリズムである FirstFit を用いる。具体的には、配置したい仮想マシンについて、物理マシンに配置できるかどうかを確認していき、一番最初に配置できる物理マシンに配置する。可用性を考慮した FirstFit も同様に一番最初に配置できる物理マシンに配置する。例として、2 対 1 構成で同時に 2 つのサービス A, B が配置される場合は、以下の順に配置する。

A : 1 → A : 2 → A : 予備 → B : 1 → B : 2 → B : 予備

6.1.3 パラメータ

シミュレーションに用いたパラメータを表 6.1 に示す。本シミュレーションでは、ホストの台数とサービスの数を固定し実験を行った。

6.1 評価方法

表 6.1 シミュレーションパラメータ

パラメータ	値
N 対 1 構成	1, 3
スタンバイ方式	HOT, WARM, COLD
ホストの消費電力モデル	線形, 60%収束, 80%収束
ホスト数	100 台
サービス数	60 台
仮想マシン数	120, 240 台
シミュレーション時間	3 日
閾値 X	85%

本シミュレーションで用いた、消費電力モデルを図 6.1, 6.2, 6.3 に示す。この消費電力モデルは、評価の際にクラウド環境全体の消費電力の最小値と最大値が変わらないように、物理マシンにおいても、最大値と最小値がほぼ同等になるようなモデルとした。以下に、使用したモデルを示す。

図 6.1 CPU 使用率と消費電力が線形な関係となる消費電力モデル

図 6.2 CPU 使用率が 60% 辺りで消費電力が収束する消費電力モデル

図 6.3 CPU 使用率が 80% 辺りで消費電力が収束する消費電力モデル

6.1 評価方法

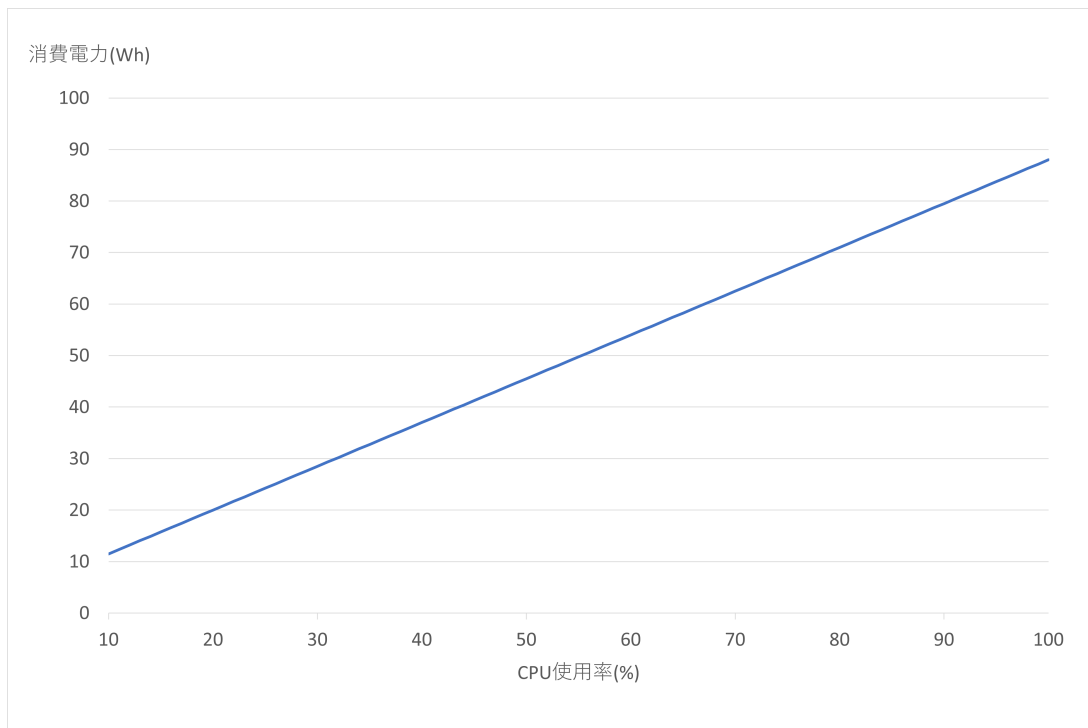


図 6.1 線形な相関を持つモデル

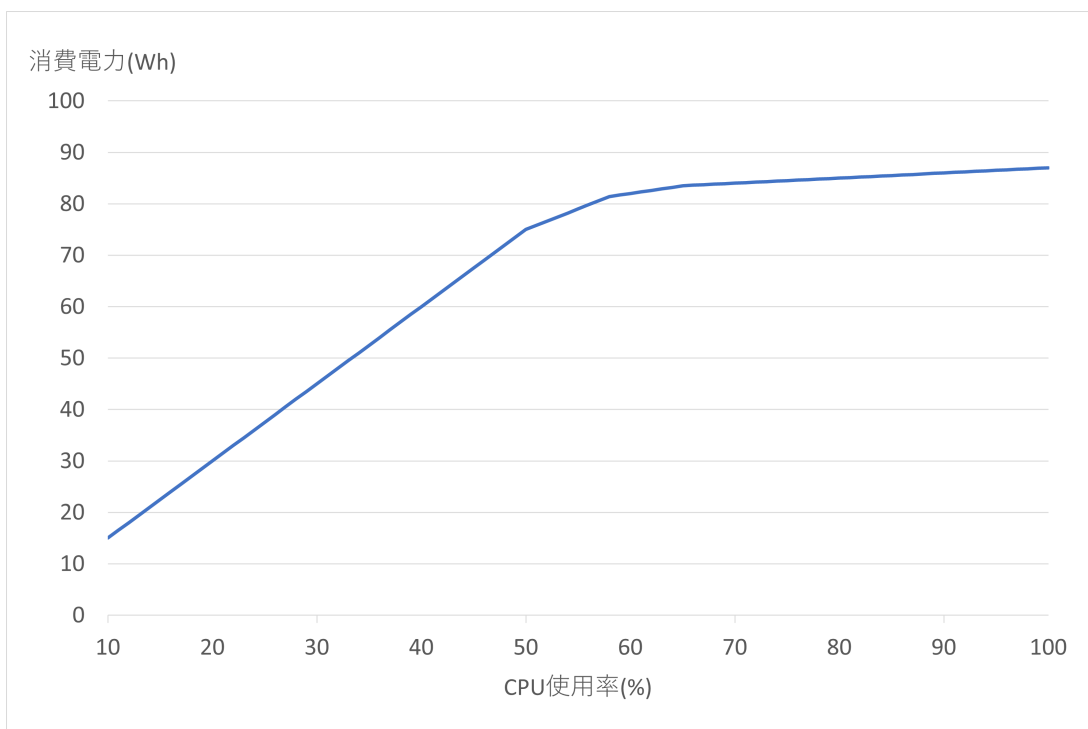


図 6.2 60%辺りで収束する相関を持つモデル

6.1 評価方法

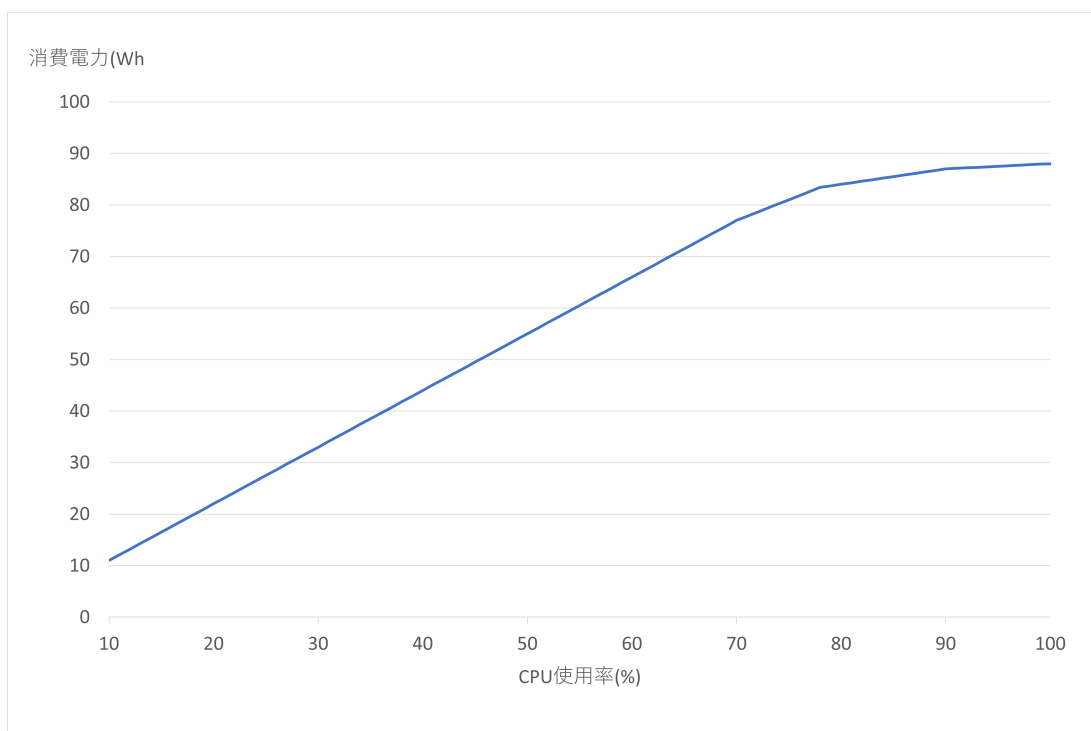


図 6.3 80%辺りで収束する相関を持つモデル

6.2 結果と考察

本節では、結果の一部を抽出して考察する。全ての結果は付録に掲載する。

6.2.1 1対1構成における結果

1対1構成における消費電力を図 6.4 に、障害復旧時間を図 6.5 に示す。

まず、FirstFit と提案手法を比較する。ホットスタンバイでは、消費電力が約 1.4 倍ほどに増加し、障害復旧時間が 45%程削減された。消費電力について、FirstFit と FirstFit(可用性考慮有) では、消費電力が約 1.5 倍ほど増加しているのに対して、提案手法では約 1.4 倍と 1 割程削減することができている。障害復旧時間については、一般にホットスタンバイでは障害復旧にかかる時間が数十秒程度であるため、45%の削減されたとすると 33 秒ほど削減することができる。コールドスタンバイでは、消費電力が 1%ほど増加し、障害復旧時間が 45%程削減された。一般的に、コールドスタンバイの障害復旧にかかる時間が数時間～数日程度であり、24 時間の 45%と考えても、13 時間の短縮となり絶大な効果を得られる。このようにホットスタンバイとコールドスタンバイでは、消費電力の増量幅に大きな差が開いた。これは、予備系の使用リソースが消費電力に大きく影響すると考えられる。

次に、FirstFit(可用性考慮有) と提案手法の比較をする。この結果では、どの形式においても約 10%の消費電力を削減できた。これは、提案手法が消費電力と可用性を考慮した配置手法として、有用であることを示している。しかし、障害復旧時間については、ホットスタンバイで 7.5%削減、コールドスタンバイで 45%削減となった。実際の削減時間としては、ホットスタンバイでは数秒ほど、コールドスタンバイでは十数時間ほどとなると考えられる。

つまり、1対1構成においては、提案手法の有用性が確認できた。

6.2 結果と考察

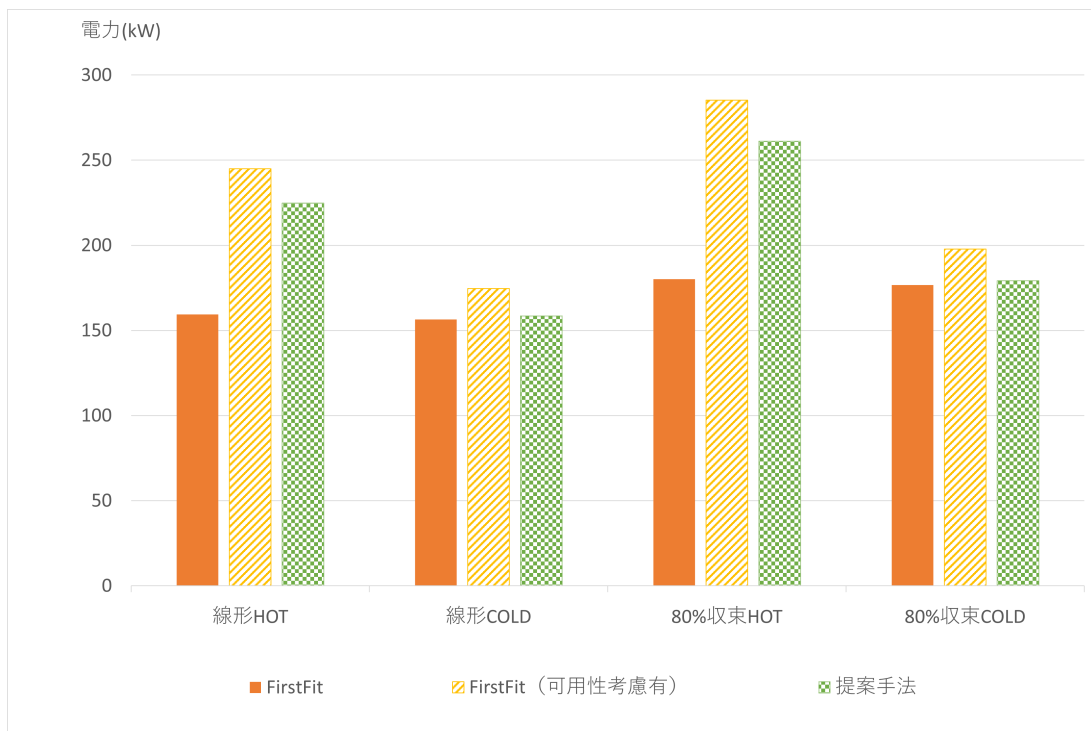


図 6.4 1 対 1 構成の消費電力

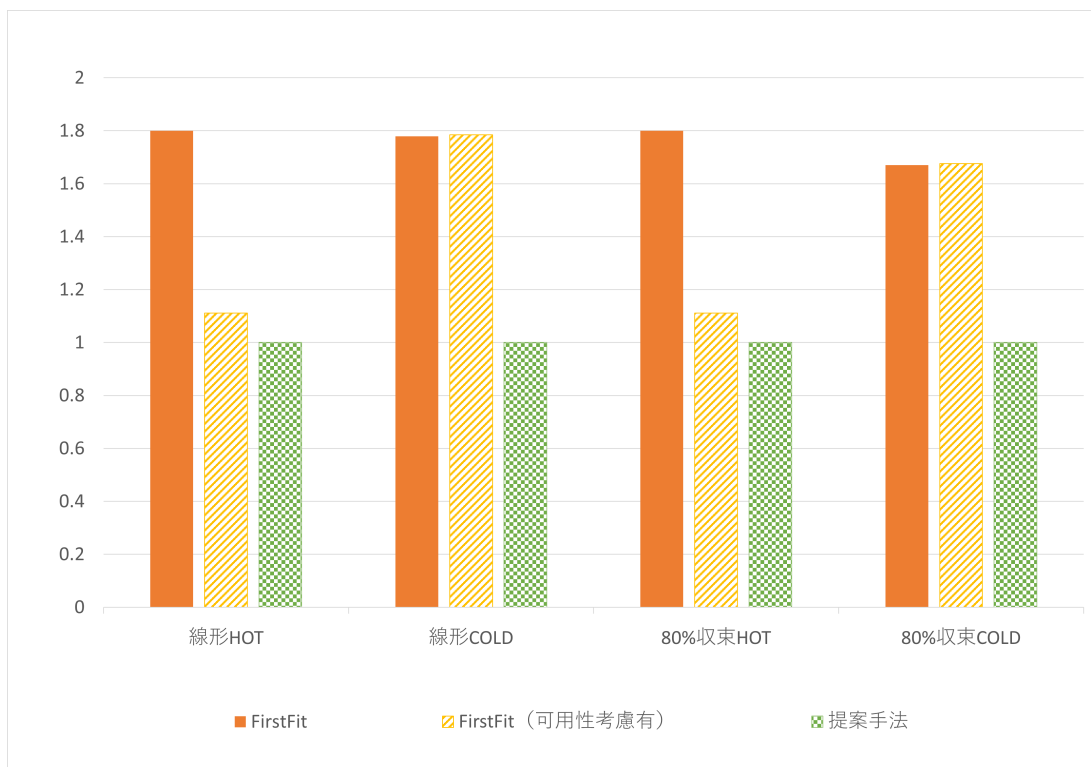


図 6.5 1 対 1 構成の障害復旧時間 (単位時間)

6.2 結果と考察

6.2.2 障害復旧時間における追加調査

ここで、障害復旧時間における追加調査を行った結果を示す。

前節では、提案手法の障害時間が大幅に削減されていることが明らかになり、FirstFit(可用性考慮有)と提案手法において、1台のホストに配置されている現用系と予備系の割合を調べることで、どのような分布で配置されるかを調査した。図6.6は、割合を示したイメージ図で、この割合ごとのホスト数の分布を調べた。

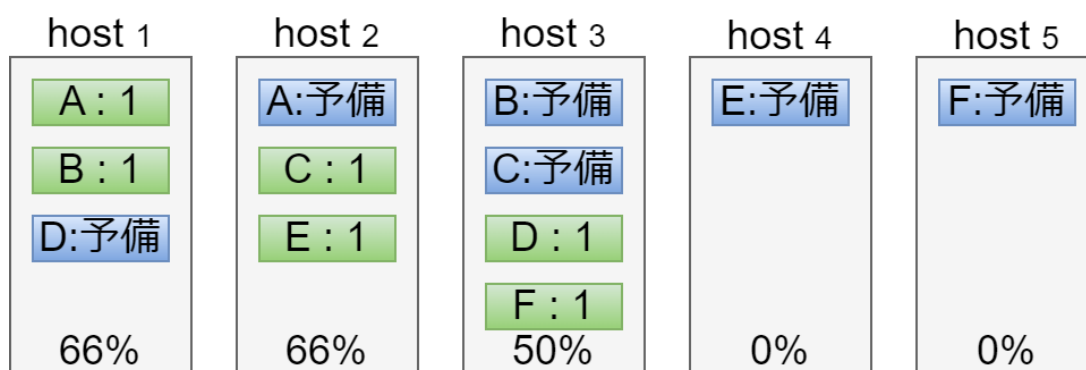


図 6.6 仮想マシンの割合

図 6.7 は、そのホスト数の分布において、特徴的であった 0%の割合を示している。この図より、提案手法では、予備系しか配置されていない 0%のホストが 40%~50%程あるという結果となった。これは、制約条件 2 が予備系を重ねないように配置しているため、予備系を配置するホストがなくなり予備系だけが集まると考えられる。

6.2 結果と考察

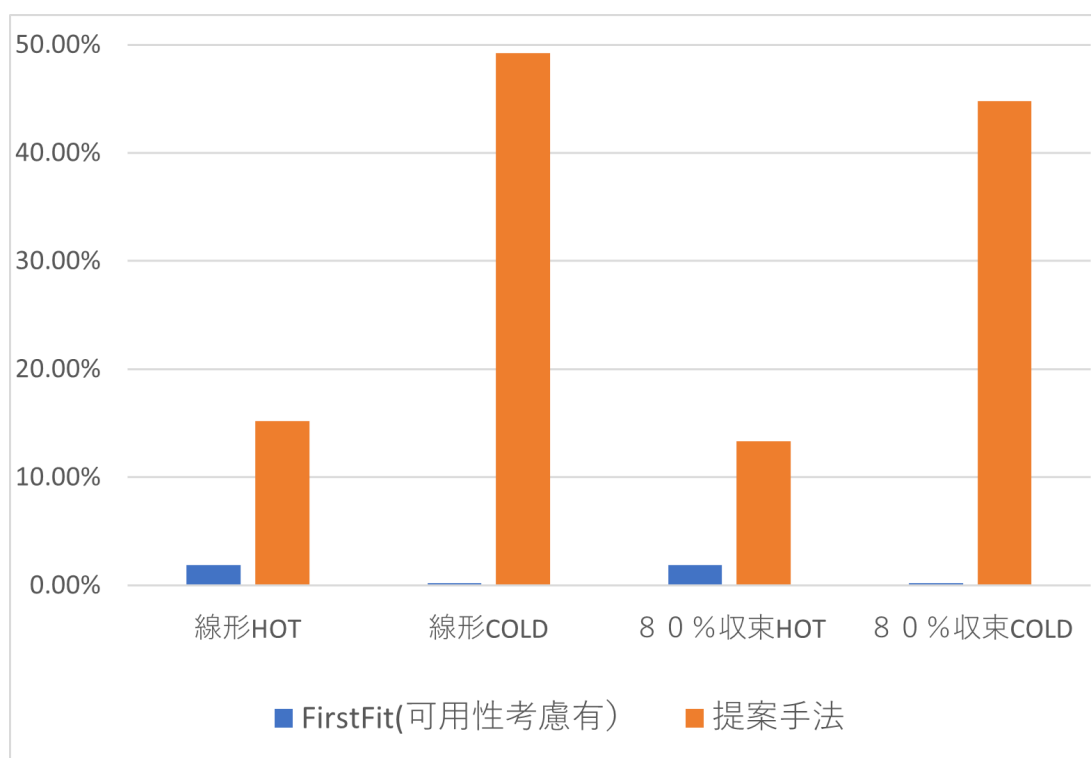


図 6.7 0%の割合

6.2 結果と考察

6.2.3 3対1構成における結果

3対1構成における消費電力を図6.8に、障害復旧時間を図6.9に示す。

まず、FirstFitと提案手法を比較する。ホットスタンバイでは、1.25倍程の消費電力で、障害復旧時間が45%程短縮されている。前項でも述べた通り、ホットスタンバイは復旧時間が数十秒程度であると考えられる。コールドスタンバイでも同様の傾向がみられるが、消費電力が2~4%増加し、障害復旧時間は最大25%程短縮された。前項と同様に、コールドスタンバイであれば、25%の削減で絶大な効果があると考えられる。

次に、FirstFit(可用性考慮有)と提案手法を比較する。消費電力は、どの形式においても2~4%程しか削減されない結果となった。障害復旧時間については、スタンバイ形式や消費電力モデルによって変わるが、最小でも15%程の削減となった。これも、前項で述べた通り、FirstFit(可用性考慮有)のデータのバラつきが大きいため、このような結果になったと考えられる。

3対1構成の2つの比較では、1対1構成と比べると提案手法の効果が小さいとわかる。このような結果となった要因として、予備系が現用系に対して数が少なく影響をしないからではないかと考えられる。N対1構成では、予備系を1つ増やすことによる消費電力は単純計算で、 $\frac{N+1}{N}$ 倍となり、1に収束するためNが大きくなるほど、提案手法との差が出にくいと考えられる。

6.2 結果と考察

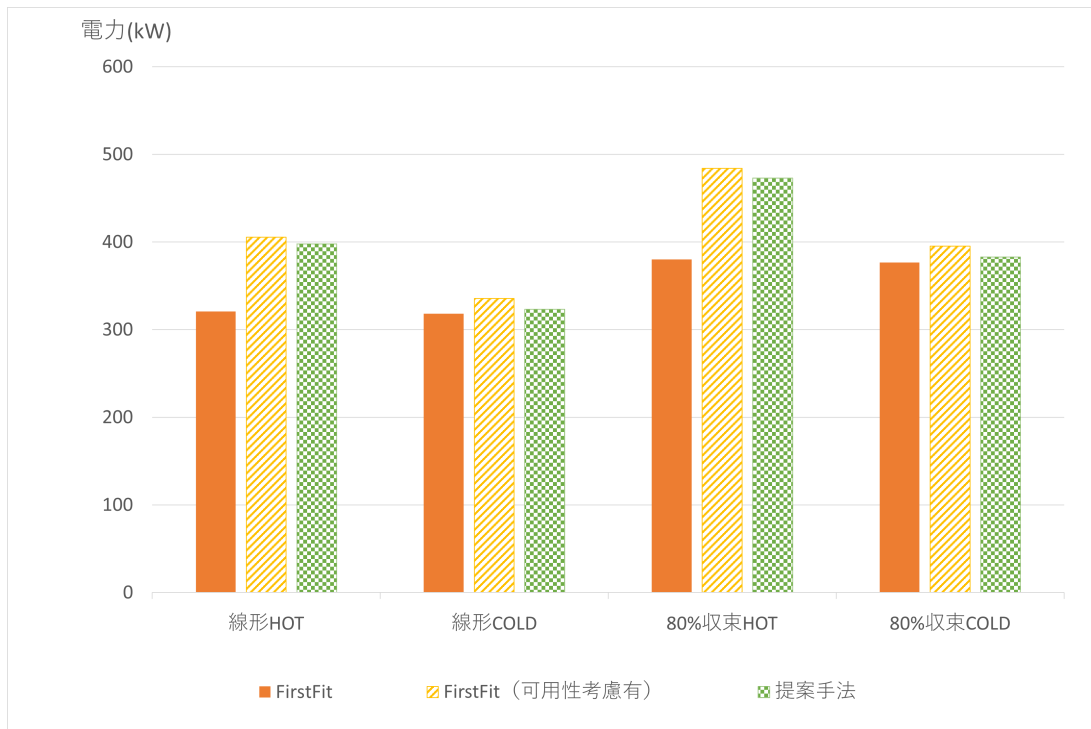


図 6.8 3 対 1 構成の消費電力

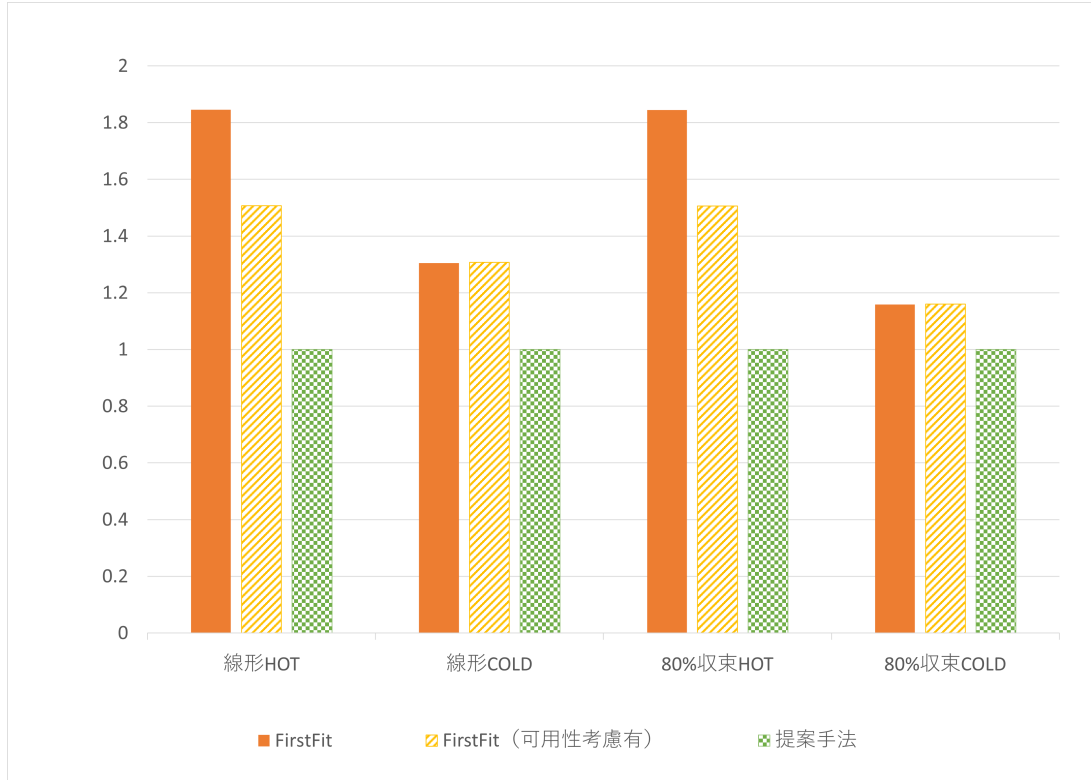


図 6.9 3 対 1 構成の障害復旧時間

6.2 結果と考察

6.2.4 提案手法における計算時間の結果

図 6.10 は、提案手法の N 対 1 構成における、1 つのサービスを配置する計算時間を示す。図では、N が大きくなるにつれて、計算時間が急激に増加することが確認できた。従って、N がさらに大きくなると、指数関数的に計算時間が増大すると考えられる。

また、初期配置とそれ以外の配置については、N が大きくなるにつれて、計算時間に差がでることが分かった。これは、クラウド環境にサービスが配置されていないと、アルゴリズムの手順 2 で抽出する組が多くなることが要因であると考えられる。

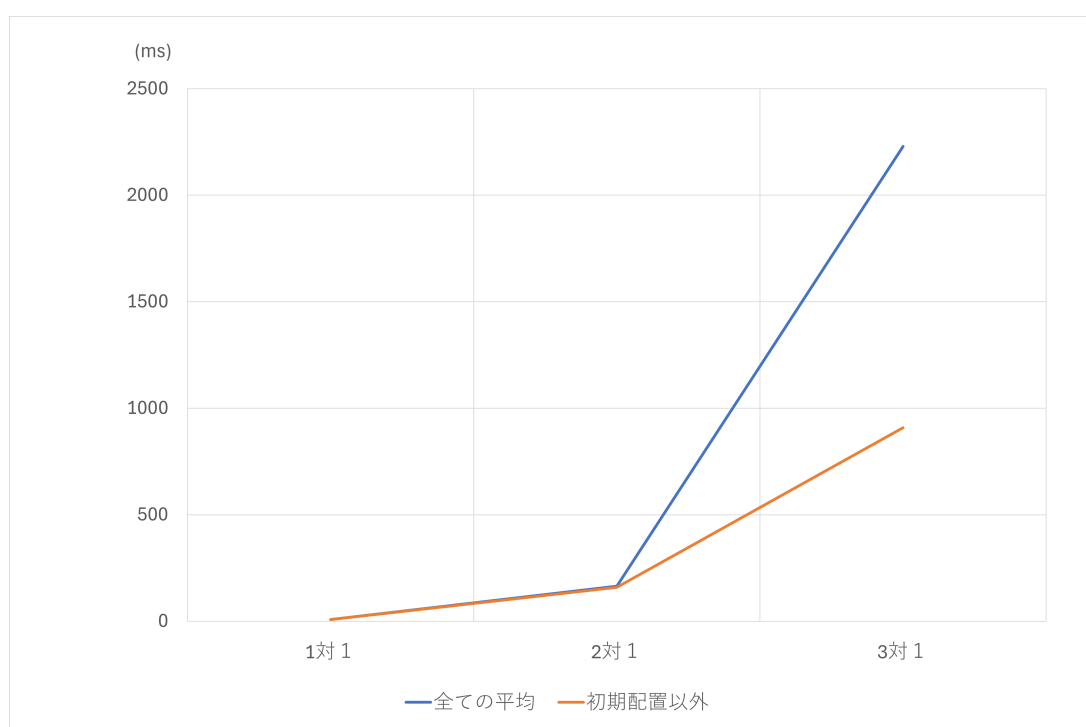


図 6.10 N 対 1 構成の計算時間

6.2.5 消費電力モデルの違いにおける結果

図 6.11 は、物理マシンの消費電力モデルを比較した消費電力である。グラフより、FirstFit(可用性考慮有) と提案手法を比較すると、線形なモデルと 80%収束のモデルが約 10%の消費電力を削減しており、60%収束のモデルでは、5%程の削減になった。これは、今回の閾値が 85%であることが影響していると考えられる。今回は、物理マシンに数台しか配

6.2 結果と考察

置されない程度のスケールになっている。そのため、85%の閾値を超えないように仮想マシンを配置するとなると、多くの物理マシンが75~80%程のCPU使用率に値する。その時、図6.2, 6.3より、75~80%では、60%収束モデルと80%収束モデルで、10Wh程違うことがわかり、これが起因していると考えられる。

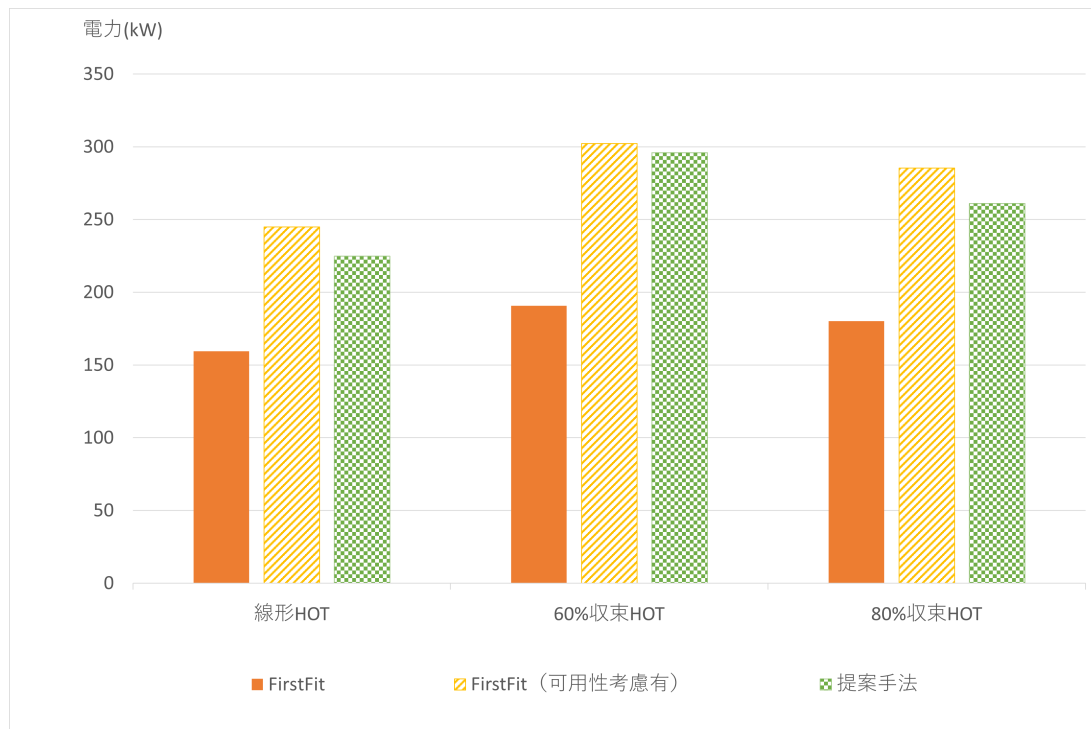


図 6.11 消費電力モデルの違いによる消費電力

第7章

おわりに

7.1 まとめ

本研究では、クラウドコンピューティングにおける可用性と消費電力を考慮した仮想マシン配置手法を提案した。

近年、クラウドコンピューティングの需要が高まっており、クラウド事業者は安定的なサービスの提供に尽力している。安定的なサービスの提供には膨大な計算資源が必要であるが、膨大な計算資源を用意するとデータセンタでの消費電力が増大する問題がある。そのため、安定的なサービスの提供のために可用性を考慮し、さらに消費電力も考慮した仮想マシン配置手法を提案した。

提案手法では、現用系と予備系の仮想マシンを同時に配置できる組合せを考え、推定消費電力を計算し、最小な配置を選択する。評価により、提案手法の消費電力が削減され、障害復旧時間が短縮されたことが確認できた。さらに、N対1構成における評価も行い、Nが大きくなると提案手法の有効性が薄れていくことが確認できた。

7.2 今後の課題

本研究では、物理マシンの消費電力モデルに複数のモデルを用意して、シミュレーションを行った。しかし、N対1構成で、Nが大きくなるにつれて提案手法の有効性が薄れていくことが分かった。そこで、物理マシンの消費電力モデルを組合せたクラウドコンピューティング環境でシミュレーションする必要がある。

7.2 今後の課題

さらに、本研究は動的な負荷変動を考慮しておらず、配置手法のみを提案している。配置した後の仮想マシンの動的な負荷変動によっては、障害が発生する可能性がある。従って、仮想マシンの動的な負荷変動を考慮した仮想マシンの配置手法の検討が必要である。

謝辞

本研究を進めるにあたり，ご指導いただいた指導教員及び主査である本学情報学群横山和俊教授には，心より感謝いたします。またお忙しい中，本研究の副査をお引き受け頂きました同学群福本昌弘教授，敷田幹文教授のお二方に感謝申し上げます。さらに，研究室の方々には，多くご教授頂きました。心より感謝いたします。

参考文献

- [1] 総務省. 令和 5 年版情報通信白書. <https://www.soumu.go.jp/johotsusintokei/whitepaper/r05.html>, 2023.
- [2] Amazon Web Services. Amazon compute service level agreement. <https://aws.amazon.com/compute/sla/>, 2022.
- [3] Mahdi Mollamotalebi and Shahnaz Hajireza. Multi-objective dynamic management of virtual machines in cloud environments. *Journal of Cloud Computing*, Vol. 6, No. 1, p. 16, Jul 2017.
- [4] Meryeme El Yadari, Ali Yahyaouy, Stéphane Le Masson, Khalid El Fazazy, and Hamid Gualous. Study of the correlation between server resources utilization and energy consumption. In *2022 10th International Conference on Systems and Control (ICSC)*, pp. 125–130, 2022.
- [5] ysklog. “パソコンの cpu 使用率と消費電力は比例関係なのか？実験してみました”. <https://ysklog.net/pc/5933.html>.

付録 A

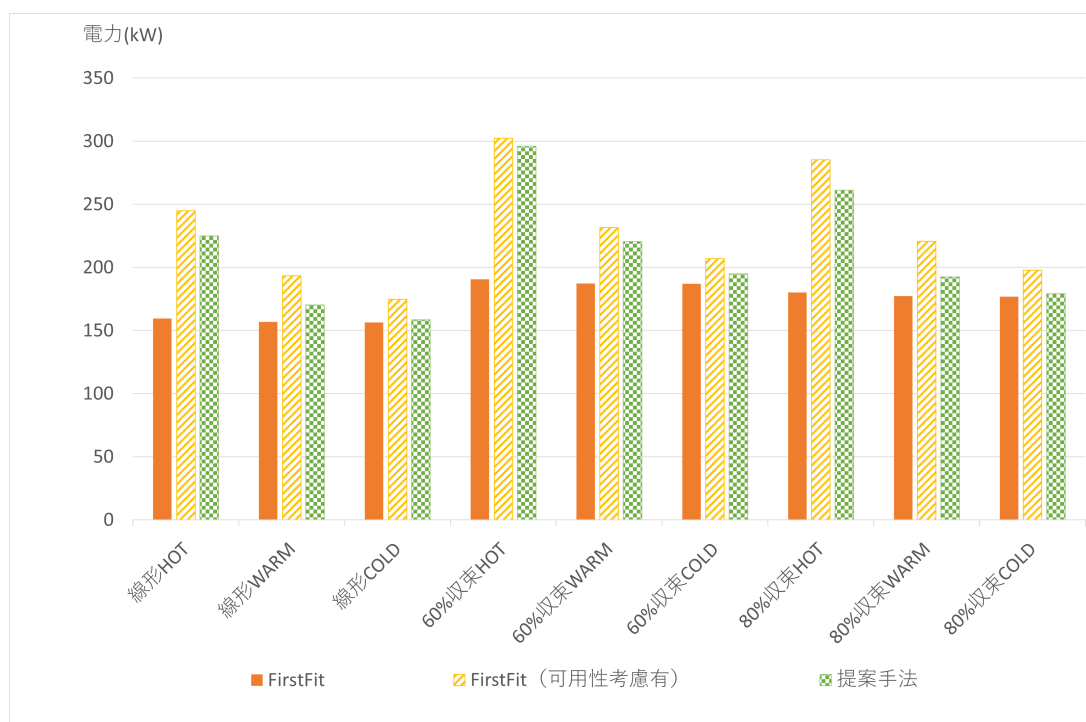


図 A.1 1対1構成の消費電力量

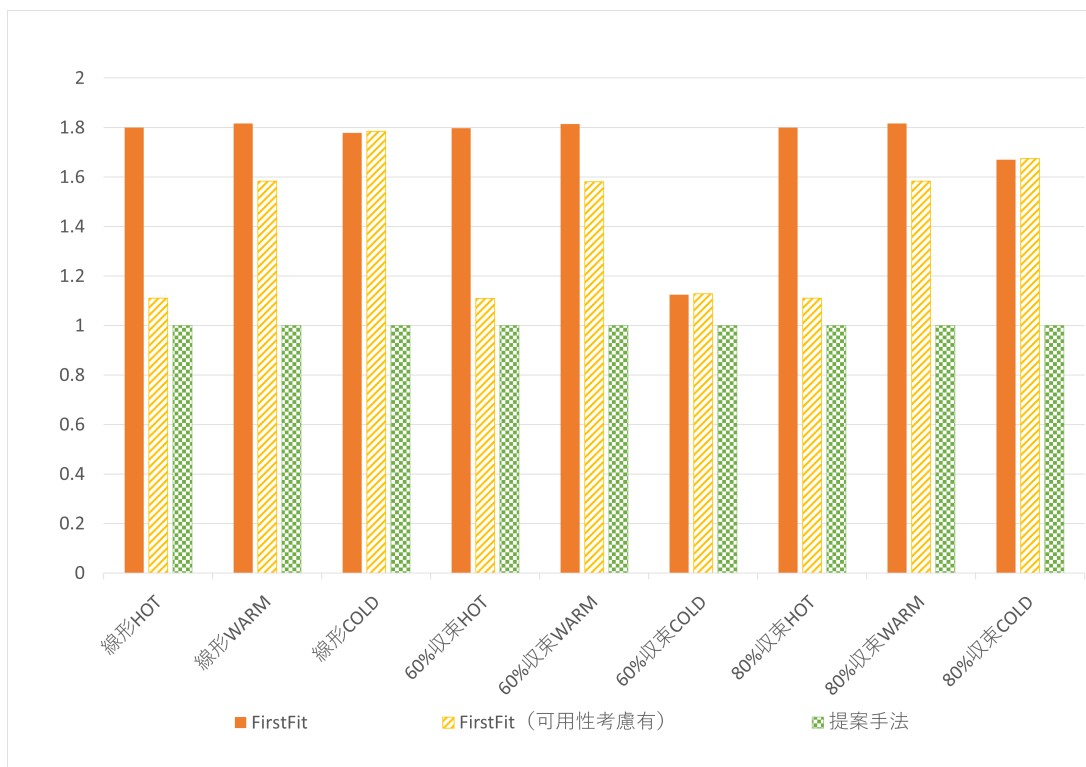


図 A.2 1対1構成の障害復旧時間

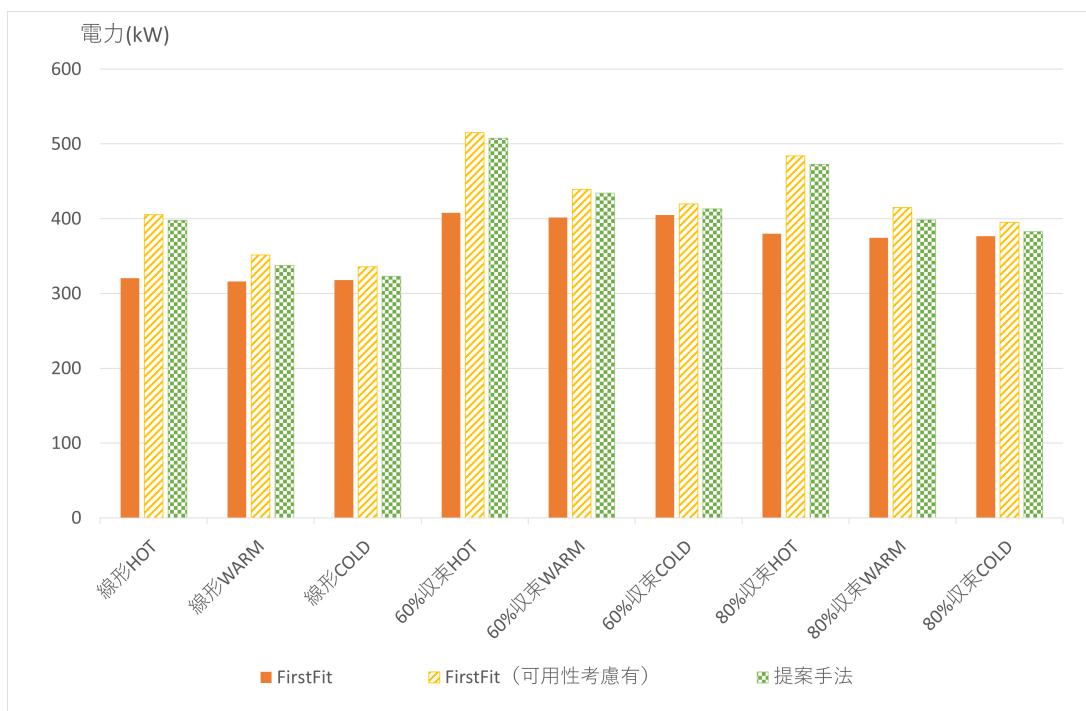


図 A.3 3対1構成の消費電力量

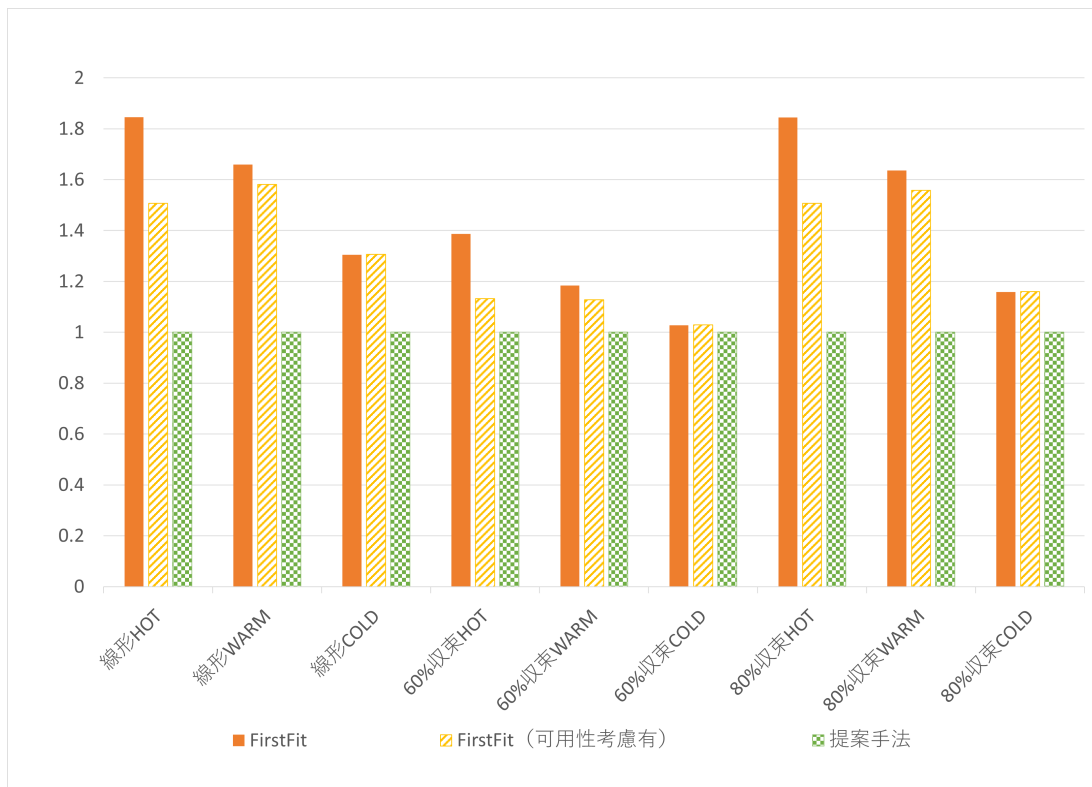


図 A.4 3 対 1 構成の障害復旧時間