

# 大規模データセンターにおける運用ノウハウ共有による 障害再発防止方式の提案

西野 博之<sup>1</sup> 坂下 幸徳<sup>1</sup> 敷田 幹文<sup>2</sup>

**概要:** 仮想化技術の普及に伴い、今日のデータセンターは大規模化複雑化が進んでいる。サーバやストレージ、ネットワークといった複数分野の機器を同時に運用することが求められる一方で、それらの運用管理を行う人材の不足は深刻化し、担当者の不在等により普段の担当とは違う分野の運用に携わる機会が増えている。担当外の管理者が設定変更操作を行った際、把握しきれていない他の設定や仕様との連携により予期せぬ障害が生じる事がある。管理者は実際の運用業務を通して複雑なシステムの構成を理解しなくてはならないが、実際の障害時になぜその障害が発生したかを判断するためには経験や勘を要する。そのため、担当外の管理者だけで復旧作業を行い、経験や勘をノウハウとして習得することは困難である。復旧作業を支援する障害原因解析を行う様々な手法が提案されているが、なぜ障害が生じたのかを類推する部分に関しては熟練管理者依存となっている。そこで、本研究では管理者の操作履歴を用い、操作によって障害が発生した理由を明確化する。また、それらの理由を該当操作時にノウハウ情報として提示することで、担当外の管理者のノウハウ形成を支援し、操作による障害再発を抑制する手法を提案する。

**キーワード:** ノウハウ共有, 障害発生理由, 大規模サーバ, 障害原因解析, 運用管理

## The Proposal of Recurrence Prevention Method with Sharing Know-How in Large-scale Data Center

**Abstract:** Due to the development of virtualization technologies data centers are more and more becoming huge. Therefore Keeping the plural apparatuses like servers, storages and routers in good working order is needed. On the other hand, shortage of server managers is serious. This is the reason why the opportunity for server managers to operate outside machines of their expertise are increasing. Those managers are likely to cause unexpected obstacles when they operate outside machines of their expertise. Although the managers must understand system configuration while maintaining servers, empirical intuition are needed to specify the cause of obstacles when obstacles happen. On this account, It is hard to do the repair work and learn know-how only in managers who do not have expertise about where applicable. As a technique to support repair work, there are some RCA methods. However it depends on the skilled managers to estimate why the obstacle happened. This proposal method supports managers who is outside of his/her area of expertise with using operation logs as a know-how-information. it can finally inhibit operation mistakes.

**Keywords:** Sharing Know-How, Failure Reason, Large-scale Servers, Root Cause Analysis(RCA), System Management

### 1. はじめに

近年、データセンターで提供されるサービスの需要や重

要性が高まっている。他方で、文献 [1] で述べられているとおり、効率化のために小規模データセンターの統合や再編によって、500m<sup>2</sup> 以上の大規模なデータセンターが増加してきている。また、仮想化技術や分散化技術の普及により、データセンターで運用されているサーバ群の運用形態は複雑なものになってきている。特に大規模なサービスになると、一つのサービスが複数の機器にまたがって存在す

<sup>1</sup> 北陸先端科学技術大学院大学情報科学研究科  
School of Information Science, Japan Advanced Institute of Science and Technology

<sup>2</sup> 北陸先端科学技術大学院大学情報社会基盤研究センター  
Research Center for Advanced Computing Infrastructure,  
Japan Advanced Institute of Science and Technology

る、複数の機器が連携して動作している等、複雑なシステム構成をもって運用されている。その上、こうした大規模環境では、一人で運用することは困難であるため、担当を分けて複数の管理者によって運用されることが一般的である。しかし、データセンターにおける運用管理者の数は慢性的に不足しているため、一人当たりの担当箇所が増加傾向にあり、担当部門外の機器を操作する機会も増えてきている。その一方で、データセンターの効率化や運用コストの削減のため、今後も再編や統合によるデータセンターの大規模化は加速するとみられている。そのため、大規模システムを運用する管理者の支援が求められている。

管理者が設定変更の操作を行った際、仕様やバグ等により、予期せぬ障害が発生することがある。障害の影響は広範囲に伝播するため、担当外の管理者にはその原因が分かりづらいといった問題点が挙げられる。障害の症状だけを見て管理者が原因を特定することは困難であるため、システム上で障害原因解析 (RCA:Root Cause Analysis) を行う手法が提案されている。しかし、今日のサーバ群の運用形態は複雑であるため、RCAによって提示された箇所以外にも障害に関与している要因を含んでいる事も多く、これらの要因も加味して復旧作業を行うことは担当している熟練管理者に依存しているのが現状である。

データセンターでは管理者の人材が不足しているため、本来の担当者が別の作業に手一杯となり対応できず、担当外の管理者が運用に携わる機会が増加している。あらゆる障害に対応するためには、担当外の管理者も担当管理者と同様のノウハウを身につける必要があるが、障害事象が複雑なものになると担当管理者に尋ねることで解決してしまいがちで、運用に携わっているにも関わらず、ノウハウを習得することができないといった問題がある。メンバーの入れ替わりや、担当管理者の不在など、状況によっては、現在の運用形態では障害復旧への対応が遅れ兼ねない。そのため、担当外の管理者へのノウハウ継承は急務であるといえる。

そこで本研究では、管理者が運用管理ツールを用いて行った操作と、他の設定や機器の仕様との連携障害を対象とする。復旧作業における RCA による解析結果と、熟練者の過去の運用操作に対する知見を関連づけてノウハウ情報として記録し、担当外の管理者が該当操作を行った際に注意喚起として運用操作時に提示する事で、担当外の管理者のノウハウ形成を支援する手法を提案する。本支援手法によって、担当外の管理者がシステムに変更を加えるとき、あるいは新規のシステムを構築するときに自発的に障害事例に対する理解、検討を促し、過去にあった障害の再発を予防しつつ担当外の管理者の管理能力を向上させる事を目的としている。

以下、2章で関連研究について考察し、3章では障害時における解析作業の問題について述べる。解析における熟

練の担当管理者による経験や勘をノウハウとして提示する提案方法を4章で述べ、5章で実際のサーバ群における本支援手法の実行例を示し、6章で議論を行う。

## 2. 関連研究

### 2.1 障害原因解析手法

文献 [2] では、ネットワーク上のリンク障害に対し、リンクダウンによる障害イベントだけでなく、障害箇所を利用してサービスの品質劣化やネットワークの品質劣化によって別のイベント情報が出力されることに着目している。こうした同一の障害原因によって生じる複数のイベント情報を関連づけて記録する事で、実際に障害が発生した際に、障害原因箇所を絞り込む手法を提案している。しかし、サーバ群においては、サーバを構成するそれぞれの設定ファイルが障害原因となり得る。その上、各サーバは依存関係を持って連携してサービスを提供しているため、障害発生時のイベント情報は大量になり、より複雑になる。

文献 [3] では障害事象と原因をパターン化し、障害原因機器を特定する手法が提案されている。機器内部の構成情報単位での解析手法としては、文献 [4] のように構成が一意でない分散システムで起こる障害やアプリケーション障害の障害派生関係を推測し解析する手法や、文献 [5] のようなシステムを構成する情報を収集し、解析を行うことで依存関係を明確化することでシステム構成管理を支援する手法が提案されている。また、文献 [6] では依存関係を用いた障害原因解析に要する計算コストを削減する研究も行われている。

これらの研究で用いられている解析手法は障害原因の特定を支援するためのものだが、障害がなぜ生じたのかを言及するまでは行っていない。前述の通り、一つのサービスが複数のサーバの連携によって提供されているため、これらの手法で特定された障害原因箇所以外に障害の要因を含んでいる箇所が存在する可能性も考えられる。また、それらの要因は依存関係を保持していないため、その解析及び復旧作業は熟練者の経験や勘に依存している。担当外の管理者の技術向上のためにも、そうした熟練者の経験や勘を、ノウハウとして共有する必要がある。

### 2.2 ノウハウ共有手法

ノウハウ共有に関する研究分野では、企業などの組織における情報共有手法がいくつか挙げられている。文献 [7] では、オフィスワークで必要になる情報を、重要度や必要になる場面などによって体系的に分類する事で、必要な情報がどこにあるのかをわかりやすく管理する手法を提案している。しかし、この手法では継続的に情報を整理しなければならないことや、情報収集を自発的に行わなければならない事からサーバ運用におけるノウハウ共有には適さない。文献 [8] では、ワークフローの各アクティビティにノ

ノウハウ情報を関連づけて入力する事で、その後、他の作業者が同様の作業を行った際に自動的にノウハウ情報を提示する手法を提案している。

### 3. 障害再発の防止における課題

担当外の管理者が操作を行う場合において、過去発生した障害を再発させないようにするために、障害の要因把握と、担当の管理者のノウハウの伝達が重要となる。本章では、これらについて説明する。

#### 3.1 障害の要因把握

障害発生時、1章で述べたRCAを行う事で障害原因箇所をある程度限定する事ができる。しかし、今日のデータセンターでは、複数のサーバが連携して各サービスを提供することが一般的であるため、本来であれば障害が生じない操作であっても、他の設定や仕様との連携によってはじめて障害として認識される場合がある。この場合、RCAによって特定された障害原因箇所を確認するだけでは、なぜ障害が起きたかまで把握することは難しく、障害を再発させないようにするためには、担当外の管理者が自力で障害原因箇所以外の障害に関与している要因を探せるように支援を行わなければならない。以後、このような障害に関与している要因を本論文では「障害発生理由」と呼ぶ。

#### 3.2 管理者のノウハウの伝達

障害発生時、その影響は様々なサービスに伝播し、被害は甚大なものになりかねない。そのため、一刻も早い復旧が求められる。そうした状況において、担当外の管理者が障害の事象に関して時間をかけて試行錯誤することは現実的ではない。障害の復旧作業にあたった管理者が担当外であったとしても、すぐに復旧できそうにない複雑な障害であった場合、ツールで障害情報を吸い上げ、担当管理者に尋ねることで解決しがちである。それぞれの機器に対して詳しい知識を持った管理者が解析にあたる事が、復旧までの時間を短縮する上で最適な方法である。しかし、障害に直面した担当外の管理者は、自力で障害発生理由を見つけた訳ではないため、後日担当管理者に説明をうけてもノウハウとして身に付きにくい。その上、その場に居合わせた管理者しか障害の内容を知る事ができないため、機器の仕様や他の操作との連携による障害であった場合、熟練の管理者であっても、予期せぬところで同様の障害を再発させてしまう可能性もある。

熟練している担当管理者が不在であっても、ツールによって障害情報を抽出し、遠隔にいる担当管理者に尋ねる事で解決できるが、そのような熟練の管理者がいつでもサーバ運用に携わることができるとは限らない。担当の管理者の多忙時や異動などによって、担当外の管理者だけで障害の復旧作業を行わなければならない状況は十分に考え

られる。そのため、担当外の障害解析を行うことがあっても対応できるよう普段の運用操作時から、起こり得る障害事例と、その障害発生理由を知り、ノウハウ情報として蓄積しておかなければならない。

### 4. ノウハウ共有による障害再発防止方式

本章では、運用時に必要なノウハウを管理者の間で共有する手法を説明する。まず始めに提案する手法の概要を述べた後、本方式で扱う管理者のノウハウの表現形式について説明する。最後に管理者へのノウハウ情報の提示方法について述べる。

#### 4.1 概要

今日のデータセンターは大規模化が進み、複数の機器が連携して一つのサービスが提供されている。サービスの一部を担うそれぞれの構成機器を一つずつ設定変更していくことは現実的ではない。実際の運用場面でも、運用管理ツールを用いられる事が一般的になっているため、本研究においてもそれらの運用管理ツールを用いて運用を行う事を前提に障害発生理由をノウハウ情報として提示する。

障害発生時、経験を積んでいない担当外の管理者は障害発生理由を見つけることができず、障害の復旧作業は担当管理者に依存している。そのため、操作を行った際に仕様や他の設定との連携障害を起こしてしまう可能性がある。本支援手法では、復旧にあたった担当管理者の復旧作業の内容をノウハウ情報として定義し、後日、他の管理者が該当する操作を行った際、操作の実行前に障害発生理由を注意喚起として提示することで障害の復旧作業に携わっていない管理者とノウハウを共有する手法を提案する。

RCAの解析結果と、熟練している担当管理者が復旧作業で得た知見を関連づけ、以下の5つの項目をノウハウとして蓄積する。

- 障害原因箇所  
障害の原因としてRCAによって提示された箇所。
- 障害原因操作  
復旧作業を担当した管理者が障害復旧時に運用管理ツールの操作履歴を確認し、障害を発生させるきっかけであると判断された操作。
- 類似度推定範囲  
障害原因箇所の依存関係の中で、障害に関与していると考えられる範囲。
- 障害発生理由  
復旧作業を担当した管理者が障害解析時に、障害原因箇所以外に障害に関与していると判断した要因。
- 障害理由箇所  
障害発生理由が含まれる箇所。

これらの項目を用いた提示手法の概要を図1に示す。障害原因操作に該当する操作が入力されたタイミングで、障害

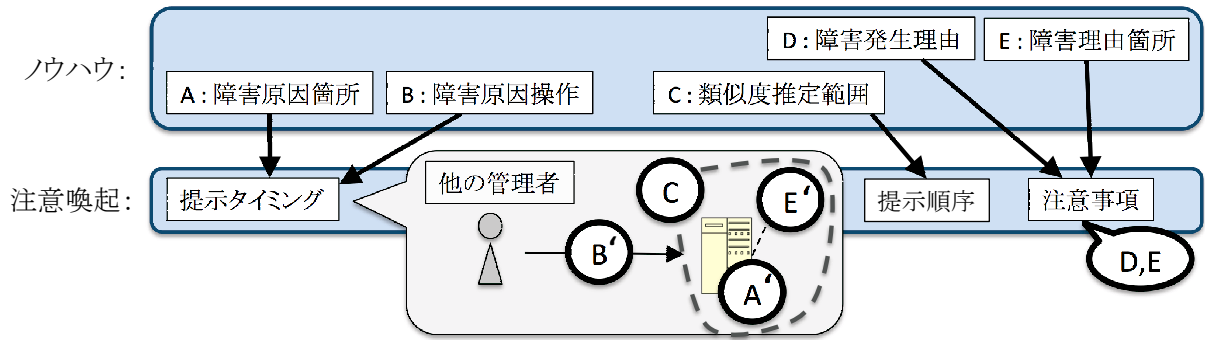


図 1 ノウハウ提示概要

Fig. 1 The outline of know-how presentation method.

発生理由と障害理由箇所を注意喚起として提示するフェーズを設ける事により担当外の運用操作を行う管理者のノウハウ構築を支援する。

#### 4.2 ノウハウの表現形式

文献 [3] の森らの研究では、システム構成情報を解析するため、システムを構成する計算機や周辺機器の情報をクラス分けしている。クラス分けされた構成情報をオブジェクトと呼び、管理者によってシステムを構築する際に設定される。各オブジェクトは属性を持ち、共通の属性を持つものを1つのクラスとして分類する。さらにオブジェクトを一意に定義できる属性を主属性とする。これらのオブジェクト単位で依存関係を抽出する手法を提案している。本支援手法においてもこの手法のクラス分けを用いる事で、障害が発生した箇所だけでなく、障害が発生した箇所と同一の構成情報を保持している他のシステムにおいても同じノウハウ情報を利用することを可能としている。これにより、違う箇所でも同じ障害が生じる危険がある場合に、同じ内容のノウハウ情報を各箇所毎に定義する手間を回避している。

管理者のノウハウの表現形式として5つの項目について

入力方法と表現形式を説明する。

- 障害原因箇所

同じシステム構成上で起きた同一の障害事例に対して、同じノウハウ情報を提示できる必要があるため、RCAでの解析結果である障害原因箇所を入力とする。  
ID: < 障害原因オブジェクト >

- 障害原因操作

本論文では運用管理ツールを用いた環境を想定している。それらの操作の履歴から障害の引き金となった操作を、障害の復旧作業にあたった管理者によって選択することで入力とする。また、管理に用いられるツールは複数あり、作業内容によって使い分けることが多い。その全ての操作に対し、一元的にIDを割り振る事は現実的ではない。そこで、その環境で用いられている各運用管理ツールにツールIDを割り振り、そのそれぞれの操作にもツール別操作IDを割り振る。この二つのIDを組み合わせる事によって各運用管理ツールの操作を一意に識別する。

ID: < ツール ID > . < ツール別操作 ID >

- 類似度推定範囲

起こった障害に関与していると思われる依存関係の範囲を管理者が選択することによって入力する。障害に関与している依存関係の範囲は障害事例によって異なるため、その範囲に含まれるオブジェクトの個数もノウハウ情報によって異なる。また、別のオブジェクトであっても同じ種類の製品である場合は同種の物として識別できる必要があるため、オブジェクトそのものを識別するオブジェクトIDと、オブジェクトの種類を識別するオブジェクトタイプIDの二つで各オブジェクトを識別する。

ID: < オブジェクト ID1 > . < オブジェクトタイプ ID1 > . < オブジェクト ID2 > . < オブジェクトタイプ

表 1 クラス分類の例 [4]

Table 1 The example of classification. [4]

クラス	該当情報	主属性	属性
Host	ホスト情報	ホスト名	OSの種類 バージョン
Dir	記憶装置上のデータ	ホスト名 パス	
Disk	物理的記憶装置(仮想ディスク)	ホスト名 ディスク名	FS ファイルサイズ
Serv	提供するサービス	ホスト名 サービス名 ポート	ソフト名 バージョン
Tape	磁気テープ	ホスト名 メディア	メディアの種類 サイズ

ID>...

- 障害発生理由  
障害が発生した理由は、障害の復旧作業にあたった管理者によって推測する部分であるため、管理者によって自由記述される。
- 障害理由箇所  
障害発生理由が含まれる箇所に関しても、障害原因箇所と同様の理由から一意に識別できる必要がある。そのため、管理者によって障害理由が含まれるオブジェクトを ID として選択することで入力する。障害発生理由がソフトウェアの仕様であるか不明の場合は '0' を入力する。

ID : [オブジェクト | '0']

### 4.3 ノウハウ情報の提示方法

管理者によって障害原因操作に該当する操作を実行しようとする前に、予め蓄積されたノウハウ情報の中から類似するノウハウ情報を選出し、障害発生理由を注意喚起として提示する。以後、障害原因操作に該当する操作のことを「提示対象操作」と呼び、操作が行われた箇所を「提示対象操作箇所」と呼ぶ。提示された注意喚起を確認することで、当該管理者は提示された障害発生理由を考慮し、操作による予期せぬ障害に対して対策を講じやすくなる。しかし、一つの提示対象操作に対し、複数の障害発生理由により多様な障害が発生することも考えられる。そのため、一つの操作に対して複数の注意喚起が提示される事も考慮しなければならない。

提示対象操作箇所が過去に障害が生じた状況と同じとは限らないため、その時々状況と照らし合わせ、同様の障害が発生するかどうかを判断する必要があるが、注意喚起の数が多ければ多いほど、全ての注意喚起に対して気を配ることは困難となる。そこで本支援手法では、運用管理ツールに各オブジェクトのメーカーやバージョン、ハイパーバイザの違い等のステータスが保持されていることに注目する。ノウハウ情報の類似度推定範囲に含まれるオブジェクトと提示対象操作箇所を含む類似度推定範囲に対応する範囲のオブジェクトの比較を行い、類似度を算出する。本論文では、ノウハウ情報に含まれる類似度推定範囲と提示対象操作箇所を含んだ対応する範囲の各オブジェクトを「操作箇所対応オブジェクト」と呼び、操作箇所対応オブジェクトの種類を「操作箇所対応オブジェクトタイプ」と呼ぶ。図2に類似度の算出アルゴリズムを示す。類似度推定範囲の類似度が高いノウハウ情報から順に出力することにより、複数の障害発生理由を提示された管理者は障害が起こる可能性が高いノウハウ情報を優先的に確認する。

```
foreach(障害原因箇所が提示対象操作と一致する全ノウハウ情報)
{
  if(障害理由箇所が類似度推定範囲に含まれている &&
    障害理由箇所の操作箇所対応オブジェクトタイプと一致しない)
    (類似度) = 0;
  else
  {
    foreach(類似度推定範囲の全オブジェクト)
    {
      if(オブジェクトID == 操作箇所対応オブジェクト)
        (一致数)++;
      else if(オブジェクトタイプID == 操作箇所対応オブジェクトタイプ)
        (一致数) += 0.5;
    }
    (類似度) = (一致数) / (類似度推定範囲のオブジェクト数);
  }
}
```

図 2 類似度算出アルゴリズム

Fig. 2 Degree of similarity calculation algorithm.

注意喚起の中には障害発生理由が不明であるが、障害が生じたため、ノウハウ情報として入力されたものも含まれる。また、理由は明記されているが、その内容に過不足があり、他の管理者が見たときに改善の余地が残されている場合も考えられる。そのため本支援手法では、各ノウハウ情報に対し改善機能として、注意喚起を閲覧したすべての管理者にノウハウ情報を編集することができる機会を設けている。これにより、ノウハウ情報そのものが成長し、より精度の高いノウハウ共有支援を行う事が可能である。

## 5. 支援システムの動作例

本章では実際の障害事例に基づき作った事例を説明し、そこでRCAの解析結果とその障害発生理由の違いを述べ、提案方式の動作例を示す。

### 5.1 障害事例

著者らの大学で実際に運用されているメールサーバの例について説明する。このメールサーバは学生、教職員、さらにそれぞれの事務局員に割り当てられたメールアドレスを用いてメールをやりとりするために運用されている。さらにそれらのアドレスを用いたメーリングリストが管理されている。このメーリングリストに対し、送信元を学内のメールアドレスのみに限定する設定を行ったところ、その

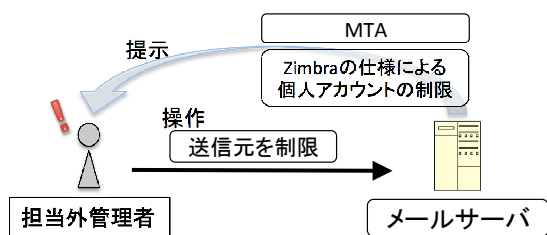


図 3 動作例における注意喚起

Fig. 3 Alert in contrast with the usage example.



メーリングリストに対し、学内のメールアドレスであるにも関わらず、事務局からの送信ができなくなってしまう障害が発生した。

## 5.2 障害原因とノウハウ

この障害に対して RCA を行うと、障害原因として、複数の派生障害に参与しているメール転送エージェント (MTA:Message Transfer Agent) が提示されることが考えられる。普段メールサーバの運用に携わっていない管理者は、障害原因として挙げられる MTA の設定に気を取られ、設定以外に障害発生理由がある可能性を見落としがちである。実際に障害の引き金になった操作は MTA に対して実行されているため、設定そのものが障害原因として障害に参与していることは確かである。しかし、その設定だけでは今回の様な障害が起こり得ないため、それ以外の障害発生理由を探さなければならない。

この事例では、MTA に用いられていた Zimbra が“送信元を限定すると個人アカウント以外のメールアドレスからの送信も制限される”という仕様であったため、個人アカウントではない事務局員のメールアドレスからのメールが送信できなくなってしまう。この障害はメールサーバ上で動作する MTA の詳細な仕様を把握していなければ気づく事ができず、担当の管理者に依存してしまいがちな例である。メーリングリストの送信元を制限する操作とメールサーバの仕様を関連づけ、表 2 の様な ID をノウハウ情報として入力しておく事で、それ以降、他の管理者が操作を行う際、実際に知っていなければ気づくことができない仕様が提示されるため、マニュアルに記述することが難しい細かい仕様や各設定による障害に対して有用である。

表 2 ノウハウ情報データセット

Table 2 Data-set of know-how information.

指標	ID
障害原因箇所	MTA
障害原因操作	< 操作ツール ID>.< 送信者制限操作 ID>
類似度推定範囲	Storage2.Xen...Zimbra...
障害発生理由	Zimbra の仕様と送信者制限操作の連携障害によって事務局から ML への送信ができなくなる
障害理由箇所	MTA

## 5.3 類似度

この事例では、メールサーバで用いられている MTA の仕様によって障害が生じたが、同様のシステムであっても MTA が違えばこの障害は発生し得ない。そのため、本支援手法においても同様のシステムで過去の障害が起こる可能性について言及する必要がある。

MTA の他にも、メーカーやハイパーバイザ種別など管

理する機器の違いによって障害が引き起こされる可能性もあるため、提示対象操作箇所を含めた類似度推定範囲内の各オブジェクトを比較することによって類似度を求める。

同一システムに蓄積されたノウハウ情報を表 3 に示す。

表 3 ノウハウ情報のオブジェクト ID・オブジェクトタイプ ID と障害理由箇所

Table 3 Know-how information object ID / Object type ID & The obstacle reason part.

ノウハウ	Storage	Hyper visor	...	MTA	障害 理由箇所
該当ノウハウ	A Storage2	B Xen	...	C Zimbra	MTA
ノウハウ A	D Storage2	E ESXi	...	F Qmail	MTA
ノウハウ B	G Storage1	E ESXi	...	H Postfix	Storage

本章で取り上げている事例に該当するノウハウ情報が“該当ノウハウ”である。これらのノウハウ情報が蓄積された後、同一システムに対して表 4 の提示対象操作が行われた場合のノウハウ情報提示について説明する。

表 4 提示対象操作箇所を含んだ操作箇所対応オブジェクト

Table 4 Operation part correspondent object including The part of target operation.

提示対象操作	Storage	Hypervisor	...	MTA	...
送信元を制限	G Storage1	E ESXi	...	I Zimbra	...

該当ノウハウは MTA の操作箇所対応オブジェクトが異なる (C, I) が、操作箇所対応オブジェクトタイプ (Zimbra) が同じであるため、一致数が 0.5、ハイパーバイザは操作箇所対応オブジェクト、操作箇所対応オブジェクトタイプ共に異なるため、一致数は 0 となる。また、ストレージも MTA 同様に操作箇所対応オブジェクトタイプのみが一致するため一致数 0.5 となる。これらの一致数を足し合わせ、類似度推定範囲の全オブジェクト数で割った値が類似度となる。他のノウハウも同様であるが、ノウハウ A は障害理由箇所の一致数が 0 であるため、表 4 の提示対象操作では起こり得ないノウハウ情報と判断し、類似度 0 となる。類似度算出結果が表 5 である。この提示対象操作に対してはノウハウ B と該当ノウハウの障害発生理由が注意喚起として順に提示される。

表 5 類似度算出結果

Table 5 The result of similarity calculation.

	Storage	Hypervisor	...	MTA	類似度
該当ノウハウ	0.5	0	...	0.5	1/n
ノウハウ A	0	1	...	0	0
ノウハウ B	1	1	...	0	2/n

## 6. 考察

本章では提案手法である担当外の管理者へのノウハウ共有支援手法について議論を行う。

### 6.1 ノウハウ提示手法の妥当性

今日のデータセンターは大規模化が進んでおり、サーバ群に障害が発生した際、広範囲に影響し、なぜ障害が生じたのかが分かりづらいという問題点があった。障害原因箇所を発見する手法として様々な RCA 手法が提案されているが、障害発生理由を見つけることは担当の管理者に依存していたため、担当外の管理者が各事例の障害発生理由を把握することは難しく、担当者の不在時の対応に支障を来していた。担当管理者の不在時の対応策として、現状の仕様をまとめたマニュアルの作成が考えられるが、それぞれの仕様や設定に対して行ってはいけない操作を網羅的にマニュアルに記述することは困難である。障害の解析時にノウハウ情報を蓄積する事で、実際に知らなければ障害が発生する恐れがある事象について効率的に周知することが可能となる。

ノウハウ情報の入力、障害の解析作業にあたった管理者であれば誰でも入力することができ、障害が発生したシステムに詳しい人が対応する事が多いが、他の管理者がみた際、情報の過不足や誤りなどに気づくことも考えられる。そのため、最初に入力した管理者以外がその情報を閲覧した際、その中身を書き換えることが可能な仕様となっている。担当管理者が気づかなかった知見が加わることで、一つ一つのノウハウ情報の正確性が増し、より質の高いノウハウの共有が期待できる。

本支援手法では操作箇所対応オブジェクトが一致するかどうかだけでなく、一致しない場合でも、その種類である操作箇所対応オブジェクトタイプが一致していれば適用可能なノウハウ情報であると判断する。これにより、同一のオブジェクトではない別のシステムに対しても同じノウハウ情報を提示する。そのため、広範囲で同じノウハウ情報を共有できる。その一方で、一つの提示対象操作に対し、障害発生理由が異なる複数の障害事例がノウハウ情報として提示される事が考えられる。提示されたノウハウ情報は過去の障害事例であるため、閲覧するだけでもノウハウの構築支援は行える。しかし、提示対象操作が行われたかどうかのみを提示の条件にしているため、提示対象操作が行われたそれぞれの状況でも注意すべき内容かどうかは自明ではない。また、これらの情報が多すぎる場合、その全てを確認する事は困難であるため、復旧作業にあたった管理者がノウハウ情報に入力した類似度推定範囲にあるオブジェクトの類似度を算出することでノウハウ情報に格納されている障害理由が起きる確率が高そうなものから順に提示する手法を提案している。

5章の例では一つの提示対象操作に対して MTA が障害理由箇所であるノウハウ情報と、ストレージが障害理由箇所のノウハウ情報が提示されていた。類似度推定範囲内のオブジェクトを比較することで重みをつけたが、実際にはどちらの障害も起こり得るため、一番上位に提示されたノウハウ情報だけでなく、ある程度下位のノウハウ情報も検討する必要がある。しかし、ノウハウ情報の順位付けは担当の熟練管理者が障害発生に関与していると考えた範囲を用いて行われているため、各ノウハウ情報の障害発生理由を検討する一助となるだろう。

### 6.2 管理者教育

1章で述べた通り、障害が発生した際、その復旧作業は一刻を争うため、その過程でノウハウを得ようと試行錯誤することは現実的ではない。しかし、システムの設定や仕様を熟知した担当の管理者によって障害の解析が行われがちであるため、復旧後にその内容をきかされるだけではなくノウハウとして身に付かず、障害の復旧作業における知識を得る機会が少ないといった問題があった。

本支援手法では、運用操作にあわせて該当する過去の障害事例を提示する。また、提示するだけに留まるため、その内容が現在の状況で起こりえるのかといった判断は各管理者で検討する事になる。これにより、各管理者は操作によって発生する予期せぬ連携障害を知り、その対応策に関して検討することが可能である。5章で取り上げたメーリングリストの例では、メールサーバの仕様について知らなければ、障害を予見することは難しい。本支援手法で提示対象操作時に提示することで、経験しなければなかなか知ることができない仕様に関して詳細に知る事ができる。これらのノウハウ情報は操作に合わせて提示されるため、ノウハウを忘れてしまった場合でも、ノウハウ情報提示を閲覧する事で思い出す事が可能である。また、その知識は今後の障害の解析作業を行う上でのノウハウとして役立てられることが期待される。

ノウハウ情報の提示は、担当外の管理者のノウハウ構築支援だけでなく、担当管理者の教育コストを削減することにも寄与している。また、担当管理者であっても、仕様との連携障害は実際に経験しないと知ることが難しく、同じ構成を持った別のシステムに適用可能であるということは、担当の管理者間で仕様との連携障害に関するノウハウをシェアするという点でも有用である。

## 7. おわりに

本論文では過去の障害の復旧作業にあたった担当管理者が復旧に用いたノウハウを、注意喚起として他の管理者の運用操作時に提示して障害再発を防止する手法の提案を行った。末端の各機器からサービスに至るまで、その構成は年々複雑化し、障害の復旧作業は担当外の管理者には困

難なものになっている。そのため、なかなかそのノウハウを習得できないといった状況が続いていたが、本論文の方式により、復旧作業を行えなかった担当外の管理者と過去の障害に関するノウハウを共有することを可能にした。このような担当外の管理者の運用管理支援は、管理者の世代交代だけでなく、人員交代、組織の改編があっても、継続的にサービスを提供するために重要であり、大きな意味を持つといえる。

今後の予定として、運用管理ツールに格納されているメーカーやバージョンなどのステータスが取得できない場合に、確率モデルを用いてより正確なノウハウ情報の提示手法を検討していきたいと考えている。

## 参考文献

- [1] 「国内のデータセンター数は減少、再編や統合へ。IDC Japan」  
<[http://www.publickey1.jp/blog/10/idc\\_japan.html](http://www.publickey1.jp/blog/10/idc_japan.html)>  
(参照 2013-11-14)
- [2] 宮澤雅典, 西村公佐: サービス品質管理を考慮した障害原因解析手法の提案, 電子情報通信学会技術研究報告, ICM, 情報通信マネジメント 110(466), 7-10, 2011-03-03
- [3] 永井崇之, 名倉正剛: 迅速な危機回復を目的とする大規模向け障害原因解析システム, 情報処理学会論文誌, 54(3), 1109-1119, 2013-3-15.
- [4] 登内敏夫, 村田正幸: 潜在的な派生関係を有する障害に対する故障分析手法, 電子情報通信学会論文誌.B, J92-B(8), 1236-1244, 2009-08-01
- [5] 森一, 敷田幹文: サーバの依存関係を考慮したシステム構成管理の支援法, 情報処理学会論文誌, 46(4), 940-948, 2005-4-15.
- [6] 幾世知範, 榎本真俊, 樫山寛章, 門林雄基, 山口英: 動的依存性グラフを用いた計算コスト削減に関する一考察, 情報処理学会研究報告, [システムソフトウェアとオペレーティング・システム]2011-OS-119(7), 1-8, 2011-11-22.
- [7] 齊藤典明: 組織における知識の共有と継承に関する一考察, 情報処理学会研究報告, GN,[グループウェアとネットワークサービス]2010-GN-77(13), 1-6, 2010-11-18.
- [8] 敷田幹文, 門脇千恵, 國藤進: フローに連携した組織内インフォーマル情報共有手法の提案, 情報処理学会論文誌, 41(10), 2731-2741, 2000-10-15.