

# 大規模システムにおける 利用実態を考慮したメンテナンス通知手法の提案

藤澤 恵一朗<sup>†1</sup> 敷田 幹文<sup>†2</sup>

近年、大学や企業といった組織において、さまざまなサービスが提供されている。サービスの維持にはメンテナンスが必要であり、影響のあるユーザには事前に通知を行うことが求められている。しかし、サービスを支えるシステムの大規模化・複雑化が進み、ユーザの利用実態の把握は困難となっている。その結果、影響の有無を問わず全ユーザに通知するという手段を避ける為、管理者が通知を行うユーザを決定することになっている。今後もユーザが利用可能なサービスは増え続けることが予想され、管理者の努力だけでは十分な通知が行えなくなる可能性がある。したがって、ユーザがどの程度サービスを利用しているかという利用実態を把握する必要がある。そこで、本論文では、我々が研究しているサーバの部品依存関係の抽出手法を用い、ユーザの利用履歴から、適切な情報を適切なユーザに通知することが可能な手法の提案を行う。これまで、管理者が行っていた通知先を特定する作業の自動化がはかれ、管理者への負担の低減が可能となる。

## Proposal of notice with maintenance notified method considering actual the grasp user's use in large scale servers

KEIICHIRO FUJISAWA<sup>†1</sup> and MIKIFUMI SHIKIDA<sup>†2</sup>

In recent years, in the organization named the university and corporation are providing for students and employee a lots of services. It is necessary to maintain by administrator for the maintenance of service. However, making to a large scale and the complication of the system that supports service are advanced. Therefore, the grasp of user's use realities is actually difficult. As a result, it will notify all users regardless of the presence of the influence to prevent the notification leakage. The user whom the administrator notifies is prevented deciding it. But, the service that the user can use will keep increasing in the future. Therefore, there is a possibility that it becomes impossible to do an enough notification only by the administrator's effort. So, in this paper, The proposal that actual the grasp of the user's use can be understood from user's use history by using the solution technique of the part dependence resolving that we are researching is done. Up to now, the autimation of work to specify the notification that the administrator was doing by hand power destination can be attempted. As a result, the decrease of the load to administrator becomes possible.

### 1. はじめに

近年、インターネットが広く普及し、我々が生活する上で書かせない重要なインフラとなっている。それらを支えるサーバやネットワークは大規模化・複雑化しており、全体の依存関係の把握は管理者でさえ困難となっている。

その為、文献1)のように障害発生時の影響範囲を特定する研究や、複雑な知識を記述することで、自動で障害を通知する研究<sup>2)</sup>が行われている。前者は大規

模なシステムには不向きであり、後者は複雑な知識を記述する必要があり、管理者には高い負荷がかかる。

文献3)では、サーバの部品依存関係をクラス化し、管理者がシステムへの理解度を深める支援手法の提案を行っている。この依存関係の抽出を動的にしたものが、文献4)である。文献5)では、障害発生時に影響ある管理者を特定し、全体の通知量を削減することで、重要な情報の埋没化を防ぐ提案が行われている。

しかし、いずれも管理者に対して適切な通知を行うことが目的であり<sup>3),5)</sup>、ユーザに対しての通知は考慮されていない。つまり、ユーザに対しては有用な通知が行われているとは言えない。

たとえば、アップデートにより不具合が生ずることが予想されたとする。この場合、不具合が生ずることをユーザに対して通知を行う必要がある。しかし、障

<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 Infomation 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科学センター

Center for Infocation Science, Japan Advanced Institute of Science and Technology

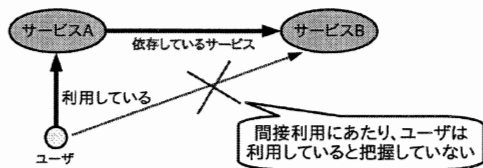


図 1 間接利用の例

害通知と異なり、通知時点では不具合はない。したがって、通知を受け取っても自分に影響があるかはユーザの利用状況によって差が出る。また、そのサービスが他のサービスと連携しているもの場合、ユーザは連携しているサービスまで利用しているという把握をすることは困難である。

そもそも、既存の研究では、障害発生時に影響が出るユーザ<sup>4)</sup>や管理者<sup>5)</sup>が対象である。つまり、現在サービスを使っているユーザが対象であり、そのサービスをよく使っているといった利用実態は全く考慮する必要が無い。そのため、アップデートのようなメンテナンス関連の通知にはむいていない。なぜなら、通知をしても現在利用しているユーザにしか通知が届かない為である。

情報漏れを防ぐ為に、利用の可能性ある全員に通知を行うと、ユーザに大量の通知が届き、ユーザ自身が影響があるか通知のたびに各自で判断し、ユーザの負担が高くなる。自分に影響がないにもかかわらず、頻繁にメンテナンス通知が来ると、自分には影響がないものと内容を確認しないまま次のメールへと移ってしまい、情報の埋没化が発生してしまう。全く通知をしないという選択をすると、全ユーザに対して通知漏れが発生する。また、図1のような、ユーザが間接的に利用しているサービスに対するメンテナンスの場合、ユーザはストレージサーバのメンテナンス通知を受け取っても自分は使っていないので影響が無いものと思ってしまう問題がある。

そこで、本論文では、メンテナンス通知を行う場合に、どのユーザに通知すべきかわかる手法の提案を行う。その結果、管理者が行っていた通知すべきユーザの特定する作業の負荷を低減できる、ユーザはサービスを使っているだけで、メンテナンスの際にユーザにも影響がある場合は自動で通知を受け取ることができるようになる。

以下、2章でサーバ部員依存関係抽出手法について述べ、3章で提案手法について述べ、4章で考察を行い、5章でまとめを述べる。

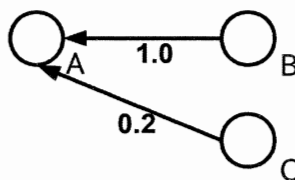


図 2 依存関係の表現方法

## 2. サーバ部品依存関係抽出手法

本論文ではサーバ部品依存関係抽出手法を用いる。近年、組織で使用されるシステムの大規模化・複雑化が進んでいる。重要なサービスに関しては安定稼働を目的として、複数のサーバにまたがって運用するなどの冗長化が図られている。その結果、複雑な構成となっており、構成の把握はシステムを熟知していない管理者にとって困難である。そこで、我々が研究しているサーバ部品依存関係抽出手法を用いる<sup>3)-5)</sup>。

サーバを構成する部品やサービスはオブジェクトという単位で扱う。このオブジェクト間のつながりを有向グラフで示すことで依存関係を表現する。有向グラフのリンクは依存度という値を持つ。依存度は0から1までの値であり、1のとき完全に依存している状態であり、逆に0のときは依存していない状態を表す。

図2に依存関係の例を示す。この場合、オブジェクトB、CはオブジェクトAに依存していることを表現している。しかし、依存度は異なる。オブジェクトBはオブジェクトAに依存度1.0で依存している。これは、オブジェクトAに障害が起こった場合、オブジェクトBにも影響が大きくなることを示す。しかし、オブジェクトCは依存度0.2でオブジェクトAに依存している。これは、オブジェクトAに障害が起こった場合、オブジェクトCには強く影響はしないが、いくらか影響することを示す。この依存度は管理者のポリシーにしたがって決める。

## 3. ユーザ向けのメンテナンス通知手法

本章では、提案するユーザ向けのメンテナンス通知手法について説明する。

### 3.1 概要

本手法は、企業や大学といった組織を対象とする。さらに、組織に属する人はそれぞれ固有のユーザIDを持ち、ユーザ同士がそれぞれ区別できるとする。

提案手法は3部にわかれて構成されている。

- (1) サーバ部品依存関係部
- (2) 利用履歴部
- (3) 情報通知部

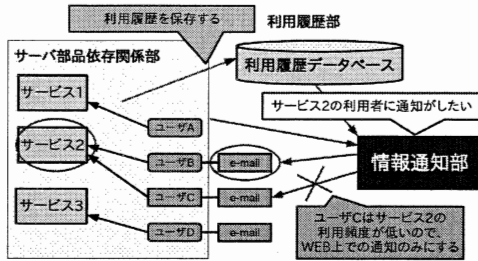


図 3 提案手法の動作例

各部の詳細は後述するとし、ここでは全体の動作例について説明する。

図 3 は、サービスが 3 個あり、ユーザがそれぞれ自由に利用できる環境である。ここで管理者は、サービス 2 に対してメンテナンスをする為、ユーザに通知を出したいとする。そのとき、管理者が情報通知部に対してサービス 2 に対して通知したい旨を入力する。情報管理部は依存関係と利用履歴から、ユーザ B とユーザ C がサービス 2 にどれくらい依存しているかという利用頻度を算出する。この利用頻度から、ユーザ B はサービス 2 にとても依存しているが、ユーザ C はごく稀にしか使っていないということを利用履歴部が判断する。その結果、ユーザ B に対しては e-mail で通知を行うが、利用頻度の低いユーザ C に対しては WEB 上だけの通知に変更し、ユーザの負担の低減をはかっている。

### 3.2 サーバ部品依存関係部

サーバ部品依存関係部は、2 章で述べたサーバの部品依存関係を把握する役割を担う。また、ユーザの利用があった場合、利用したオブジェクトの情報を利用履歴部に送る役割を持つ。

依存度は 2 章の通り、管理者のポリシーに従って定める。

### 3.3 利用履歴部

利用履歴部では、ユーザがサービスを利用したときに発生する依存関係の変化を、前項の部品依存関係部から受けとり、保存をする。

保存する履歴の例として、ユーザがアプリケーションを利用し、ファイルを編集しているとす。この場合、ユーザが利用したアプリケーションと編集したファイルがそれぞれ履歴として保存される。

### 3.4 情報通知部

情報通知部には以下の機能がある。

- (1) 利用履歴からユーザの利用頻度を算出する
- (2) 利用頻度と依存関係から、間接利用頻度を算出する

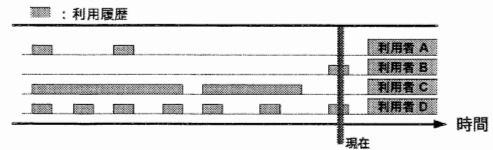


図 4 利用頻度が異なるユーザの例

- (3) 間接利用頻度から通知すべきユーザと通知手段を決定する
- (4) 利用履歴のうち間接利用頻度の高い上位の利用履歴を添付し通知を行う

機能 1 については、3.5 章、機能 2 については、3.6 章、機能 3 については、3.7 章、機能 4 については、3.8 章にてそれぞれ説明をする。

### 3.5 利用頻度の算出方法

利用頻度の算出方法について説明する。

利用履歴をオブジェクトごとにそのユーザが利用したかどうか 1 日 1 回取得する。ただし、1 日に同じオブジェクトに対して、複数回利用履歴があっても 1 回とする。つまり、1 日より細かい単位では見ない。なぜ 1 日 1 回にするかはプリンタを例にするとわかりやすい。つまり、100 ページの論文を 1 回だけ印刷したユーザと、数ページの印刷を複数回行ったユーザも、そのプリンタを利用したという点について同じだからである。プリンタ以外でも、毎日触れるものというのはそう多くは無い。触れないものに関しては利用履歴が残らないことから、1 日 1 回で問題ないと判断した。

次に、この利用履歴から 3 種類の値を計算する。3 種類の値は以下である。

- $n_w$  : 直近一週間の利用回数
- $n_m$  : 直近一ヶ月の利用回数
- $n_y$  : 直近一年の利用回数

この 3 種類の値から利用頻度  $p$  (1) を算出する。

$$p = n_w / 7 * r_1 + n_m / 30 * r_2 + n_y / 365 * r_3(1)$$

$r_1, r_2, r_3$  は割合である。重視する割合によって変更可能とする。直近をさらに重視したい場合は  $r_1$  の割合を付したり、長期的な利用を重視したい場合は  $r_3$  の割合を増やせば良い。

この数式を例に当てはめてみる。利用頻度が異なるユーザの例を図 4 に示す。既存の手法は障害通知を対象としており、通知対象は現在利用している人である。図のユーザでいうと B と D である。しかし、メンテナンス通知であれば、ユーザ C に通知すべきである。

ユーザ A は過去一ヶ月に 2 回、ユーザ B は今日のみ、ユーザ C は今週使っていないが、過去一ヶ月に 20 日、過去一年に 229 日使用しているとす。ユー



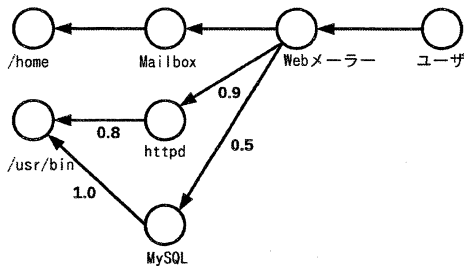


図5 間接利用頻度の算出が必要となる依存関係の例

ザ D は毎週 3 日ほど長期的に使っているユーザである。

この場合、利用頻度  $p$  の式 1 に当てはめた場合、ユーザ A は 0.02、ユーザ B は 0.05、ユーザ C は 0.42、ユーザ D は 0.36 となる。なお、 $r_1, r_2, r_3$  はすべて 0.33 とした。利用頻度  $p$  が高いほど、よくサービスを利用している、通知すべきユーザであることがわかる。

### 3.6 間接利用頻度の算出方法

間接利用頻度の算出方法について説明する。

#### 3.6.1 概要

前章 3.5 は、ユーザがサービスを直接利用している場合の利用頻度である。その直接利用しているサービスに対しては適切なユーザに通知ができるが、実際のシステムでは通知をうまく行えない可能性がある。前述のとおり、システムの大規模化・複雑化が進んでおり、ひとつのサービスは複数のサービスが組み合わさって動作していることも多くなっている。そのため、あるサービスが別のサービスに依存している場合、ユーザはその別のサービスを使っていない場合であっても、間接的に利用しているといえる。したがって、間接的な利用頻度の算出が必要となる。

間接利用頻度はサーバ部品依存関係部の有向グラフから探索を行い算出をする。図 5 の例で説明する。ユーザは Web メールer を利用しており、Web メールer は httpd と MySQL に依存している。この場合、Web メールer に対して通知を行う場合は、前章 3.5 の利用頻度をそのまま使えば良い。しかし、httpd や MySQL のようにユーザが直接利用していない間接利用にあたるサービスにはユーザの利用履歴は存在しない。したがって前章 3.5 の手法ではユーザに通知を行えない。そこで、サーバの部品依存関係の有向グラフから探索を行い、間接利用にあたるサービスに対してもメンテナンス通知を行うことが可能なアルゴリズムについて説明する。

おおまかな動作の流れは以下ようになる。

経路	依存度
/usr/bin	1.00
/usr/bin,httpd	0.80
/usr/bin,MySQL	1.00
/usr/bin,httpd,Web メールer	0.72
/usr/bin,MySQL,Web メールer	0.50

表 1 依存度テーブルの例

- (1) 依存関係テーブルの作成
- (2) 利用頻度テーブルの作成
- (3) 上記 2 テーブルから間接利用頻度テーブルの作成

#### 3.6.2 依存関係テーブルの作成例

依存関係テーブルとは、オブジェクト間の依存度がどのようになっているかを把握するために必要なテーブルである。システムの規模にもよるが、すべてのオブジェクト間の依存度を常に計算しておくことは困難である。したがって、依存関係テーブルはメンテナンス通知を行う際に毎回作成する。具体的なテーブルの例として、図 5 のオブジェクト「/usr/bin」の依存関係テーブルを表 1 に示す。

図 5 の通り、オブジェクト「/usr/bin」に依存しているオブジェクトは「httpd」と「MySQL」の 2 つである。それぞれ 0.80、1.00 の依存度で「/usr/bin」とリンクで結ばれている。さらに「httpd」と「MySQL」に依存しているオブジェクトがあり、この場合「Web メールer」がこれらに依存している。依存度はそれぞれ 0.90 と 0.50 で結ばれている。この場合、通知対象オブジェクト「/usr/bin」と「Web メールer」の依存度はオブジェクト「httpd」と「MySQL」のどちらを通るかによって異なる。しかし、テーブルを作り終えた結果、オブジェクト「httpd」を経由した場合のほうが高く、その依存度は 0.72 であることがわかる。結果、通知対象オブジェクトである「/usr/bin」の利用履歴はないが、オブジェクト「Web メールer」に利用履歴があるユーザの依存度は 0.72 であることがわかる。

#### 3.6.3 利用頻度テーブルと間接利用頻度テーブルの作成例

利用頻度テーブルとは、ユーザの利用頻度を前章 3.5 で算出したものである。ただし、依存関係テーブルにあるオブジェクトの利用頻度に限る。図 5 のオブジェクト「/usr/bin」の利用頻度テーブルの例を表 2 に示す。

この場合、ユーザ A、B ともに通知対象オブジェクトである「/usr/bin」の利用履歴は存在しない。しかし、利用頻度テーブルより、「/usr/bin」と「Web メールer」の依存度は 0.72 であることが分かっている。ここで「Web メールer」の利用頻度に依存度を掛け合わせたものをユーザの「/usr/bin」の間接利用頻度とする。図 5 のオブジェクト「/usr/bin」の間接利用頻度テーブルの例を表 3 に示す。

ユーザ	オブジェクト	利用頻度
A	Web メール	0.50
B	Web メール	0.20
A	Mailbox	0.50
B	Mailbox	0.20
A	/home	0.60
B	/home	0.40

表 2 利用頻度テーブルの例

ユーザ	間接利用頻度
A	0.36
B	0.14

表 3 間接利用頻度テーブルの例

### 3.6.4 間接利用頻度算出アルゴリズム

間接利用頻度テーブル作成の動作の流れをまとめると、以下のようになる。

- (1) 利用頻度テーブルを初期化する。
- (2) 依存度テーブルを初期化する。
- (3) 通知を行いたいオブジェクトを入力する。
- (4) 依存度テーブルに オブジェクトを依存度を 1 として追加する。利用頻度算出オブジェクトと経路オブジェクトとして オブジェクトをそれぞれ与え、経路依存度を 1 として次に進む。
- (5) 引数から利用頻度算出オブジェクトと経路オブジェクトと経路依存度を受け取る。
- (6) 経路依存度が一定以下になった場合は終了する。この値は管理者があらかじめ決めておく。
- (7) 5 の利用頻度算出オブジェクトの利用履歴を参照する。
- (8) 7 で利用履歴がある場合、3.5 章に従い、ユーザの利用頻度の算出を行う。
- (9) 8 で算出した利用頻度は 1 に利用頻度算出オブジェクトをキーとして追加する。
- (10) 5 の利用頻度算出オブジェクトの被リンクと被リンクオブジェクト一覧を取得する
- (11) 10 で取得した被リンクオブジェクトがある場合は、被リンクオブジェクトを現在の経路オブジェクトの後ろに追加したものをキーとして、依存度テーブルに現在の依存度に被リンクの依存度を掛け合わせたものを追加する。
- (12) 10 の被リンクオブジェクト毎に 5 からの作業を再帰的に実施する。引数である利用頻度算出オブジェクトは被リンクオブジェクトを代入し、経路オブジェクトは現在の経路オブジェクトに被リンクオブジェクトを後ろに加えたものを使用する。引数に与える経路依存度は依存度テーブルで追加したものと同じ値を与える。ただし、1 に利用頻度オブジェクトが追加されている場合は既に探索済みのオブジェクトなので再帰的に計算させない。
- (13) 間接利用頻度の算出を行う。利用頻度テーブル

の行ごとに処理を行う。利用頻度テーブルの行にあるオブジェクトと通知対象オブジェクトの依存度を依存度テーブルから取得し、利用頻度テーブルにあるユーザの利用頻度とこの取得した依存度をかけあわせて間接利用頻度を計算し、ユーザとオブジェクトと合わせて間接利用頻度テーブルに格納する。

### 3.7 通知ユーザとその通知手段の決定

章 3.5 および章 3.6 で算出した利用頻度と間接利用頻度を用い、通知を行うユーザと通知手段の決定を行う。

通知手段は利用頻度もしくは間接利用頻度の値によって決定する。値が高い場合は、そのユーザに影響が大きいことを示す。したがって、e-mail がふさわしい通知手段である。値が平均以下であるなど、それほどユーザに対して影響が大きくないと判断されれば、e-mail のようにユーザ負荷の高い方法ではなく、WEB 上のページのみといった負荷の低い通知手段にする。ユーザにはそれほど影響がないのに e-mail のような方法を使ってしまうと、e-mail ボックスはメンテナンス通知ばかりになってしまい、利用頻度が高いと判断された通知すら見失ってしまう問題が起こるためである。

### 3.8 わかりやすい通知内容

提案手法では利用履歴がデータベースにあるので、この情報とサーバの部品依存関係から、よりよい通知内容にできる。

たとえば、すべてのユーザが利用するファイルサーバがあったとする。しかし、ユーザ数が多いなど、ファイルサーバが複数台あったりすることはよくある。このうち、ファイルサーバの一部のパーティションに対してのみメンテナンスを行うとする場合、すべてのユーザには影響が出ないことが予想される。しかし、メンテナンスを行おうとしている部分はそのユーザが利用しているかは、実際にあるファイルを見てみるのが正確であるが、手動で行うのには無理がある。多くの場合、学年や学科、研究室といった単位ごとにパーティションを与えており、管理者はこの情報から利用可能な全ユーザに通知をするのではなく、影響が出る可能性が高いユーザ全員に対して通知お行っている。

この場合、「file-sv09 に対してメンテナンスを行います。」といったホスト名やパーティション名の内容では、file-sv09 に自分は何のファイルを置いてあるのか、いつ使ったかという情報が無い。なので、通知を受け取っても、自分には関係ないと思うなどして対策を取らない可能性がある。これは何度も出てきているが、サービスは複数のサービスが連携して動作していることが多く、ユーザには具体的な影響を示して気づかせなければならない。そこで、提案手法を用いて、file-sv09 の利用履歴のうち、利用頻度の高いオブジェクトの内容をユーザ毎に記述する。その結果、ユーザ

は自分が操作したファイルやサービス名が示されていることから、自分のファイルに影響があるということがわかる。

#### 4. 考 察

これまででは、ユーザが臨機応変にサービスを利用可能であることに加え、サーバの部品依存関係の問題から、どのユーザに通知をすべきかはわからなかった。しかし、重要な情報の埋没化が発生することから利用可能である全員に通知を行うことはできなかった。そこで、よく用いられているのが以下である。

- メールングリスト
- RSS
- WEB ページ

特にメールングリストがよく用いられる。メールングリストは、サービス毎に用意したり、大学の学科や学年、企業の一部門といったグループに作成するものである。メールサービスを利用する場合はユーザが特定のメールングリストに登録する手間が発生する。複数のサービスを利用している場合はその分メールングリストに登録しなければならない。サービスを使わなくなったらメールングリストから退会する手間もかかる。こういったわずらわしさにより、サービスの利用者全員がメールングリストに登録しているケースは少ないと思われる。図6に例を示す。この例ではあるサービスに対してメールングリスト等のユーザ向けの通知手段を設けており、利用者はサービスを利用する場合はあらかじめ登録することを管理者は想定している。しかし、前述のとおり、実際は図6のようにすべての利用者が登録しているわけではない。サービスを利用していても通知手段に登録してないユーザには通知漏れが発生してしまう。また、サービスを利用しなくなってもこの通知手段から退会をしないユーザも居ることが想定される。

さらに、提案方式では、サーバ部品依存関係のオブジェクト単位で、そのオブジェクトを間接的にも利用しているユーザに対して通知を行うことができる。すなわち、ファイルサーバの内部にあるハードディスクドライブ一台から利用しているユーザ、そうでないユーザの判別が自動で行える。もし、この場合メールングリストでメンテナンス通知を出す場合、膨大な数のメールングリストが必要となる。ユーザは新しくサービスを利用するたびにたくさんのメールングリストに登録が必要になり、これは現実的には困難である。

特に、ストレージサーバのように被依存となるサービスの場合、ユーザはストレージサーバに依存するサービスのメールングリストには参加するが、ストレージサーバまでは参加しないことが多いからである。ストレージサーバのようにユーザが直接触れないサービスに関して、これまでは管理者の努力により影響範囲を

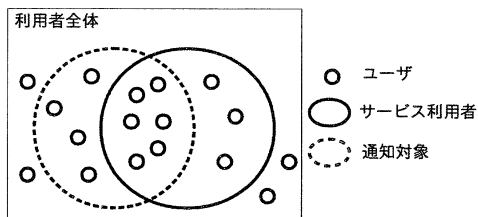


図6 通知漏れのユーザが居る例

推定し、通知を行っていた。具体的には、ユーザが直接触れないサービスに関しては影響がありそうなサービスを特定し、そのメールングリストに流すといったことが行われている。ここで、本論文の提案手法を用いることで、影響範囲を特定でき適切なユーザに通知ができる。特に影響範囲やその通知を行うユーザの判断は管理者のスキルにも左右される。この点においても、自動で行うことのできる提案手法には利点があると言える。

さらに、本論文の提案手法では、ユーザ毎に内容の異なる通知を行う。これまで、メンテナンス通知はそのメンテナンスの内容のみがかかれ、ユーザへの影響はかかれていない。これも、メールングリストのように、複数のユーザに同じ内容を配信する都合上、複数のユーザ全員に関係ある内容しか書けないからである。しかし、提案手法では、利用履歴を保存してあり、この情報を活かし、ユーザ毎に影響が大きいと思われる利用頻度が高いものを記載することができる。その結果、ユーザはメンテナンス前に対応を取ることができる。メンテナンス後に問題が起ってしまった場合も、管理者に対して、問題の内容を伝えやすくなり、管理者はどこに問題が起きているのか把握しやすくなる。

#### 5. おわりに

本論文では、サーバ部品依存関係とサービスの利用履歴を用い、メンテナンス通知手法の提案を行った。提案方式を用いることで、通知を行うべきユーザの把握を自動化でき、その結果、管理者が手動で行っていた作業負担の低減が可能となる。ユーザ側にとっても、これまでメンテナンス予告を受けるためにはサービスごとに独自のメールングリストやRSSを購読するといった作業が必要であったが、これらが不要となり、自分への影響が高いものはサービスを使っているだけで通知が来る利点がある。特に、メンテナンス事後に問題が起ってからユーザは管理者に問い合わせていたが、事前に影響がわかっているなら事前に管理者に

問い合わせることできるようになる。ユーザにとっては問題が起こる前に対応がとれ、管理者にとっては、メンテナンス事後に問い合わせが集中する問題が緩和されることにつながる。

現在、提案手法の実装を行っており、ユーザの利用頻度を算出可能な部分まで完成している。今後は、実際に近い環境で比較実験を行いたい。既存の通知手法と、提案手法の比較をすることで、提案手法の算出する利用頻度の値がどの程度の利用実態を表すのかといったことを示したい。

### 参 考 文 献

- 1) 酒井将人, 石川裕: CIM を用いた障害検知システム, 情報処理学会 OS 研究報告, No.86, pp. 125-132 (2006).
- 2) 今野将, 吉村智志, 羽鳥秀明, 岩谷幸雄, 阿部享, 木下哲夫: 能動化された状態情報に基づくネットワーク管理方式, Vol.46, No.2, pp. 493-505 (2005).
- 3) 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, Vol.46, No.4, pp. 940-948 (2005).
- 4) 奥井裕, 敷田幹文: 大規模サーバにおける部品依存関係の動的抽出方式の提案, 情報処理学会 インターネットと運用技術シンポジウム (2009).
- 5) 敷田幹文: 大規模サーバ間の部品依存関係に基づく障害通知方式の提案, 情報処理学会論文誌, Vol.49, No.3, pp. 1185-1193 (2008).