情報処理学会論文誌 Vol.53 No.3 978–986 (Mar. 2012)

大規模サーバ間の部品依存関係を利用したユーザ指向通知方式

敷田 幹文[1,a)] 藤澤 恵一朗[2,†1]

受付日 2011年6月30日, 採録日 2011年12月16日

概要：近年，大規模な組織では様々なサービスが提供されており，組織に所属するユーザは研究や業務を遂行するためにこれらのサービスを自由に利用する．一方，システムを構成するサーバ群の各部品間の依存関係が複雑になっており，システムの一部に対するメンテナンスの際に，どのユーザにどの程度影響が出るかを特定することが困難となっている．その結果，影響のない多数のユーザに対しても通知を行う可能性があり，また通知を受け取ったユーザが自身の利用への影響を推測することができず，影響のあるユーザへの通知に効果がないことも多くなる．本論文では，サーバの部品依存関係と各サービスの利用履歴を用い，適切なユーザに適切な内容を適切な手段で通知する方式を提案する．また，提案方式に基づくメンテナンス通知のシミュレーションを行い，有用性を議論する．

キーワード：運用管理，大規模サーバ，依存関係，メンテナンス通知

A Method for User-oriented Notifications Using Part Dependence in Large-scale Servers

Mikifumi Shikida[1,a)] Keiichiro Fujisawa[2,†1]

Received: June 30, 2011, Accepted: December 16, 2011

Abstract: In recent years, various network services are provided in large organizations, and users who belong to the organization use these services freely. On the other hand, part dependence in servers become complex. It is difficult to specify how to be influenced when a part of the system is maintained. As a result, a lot of users who do not have influence are notified. The notifications to users who have influence are not effective, because the users who receive the notification do not understand influence on oneself. In this paper, we propose a method to notify appropriate contents to appropriate users by appropriate media. This method uses the part dependence of the server and access logs of each service. We simulated the notification of maintenance based on this method. And, we discuss usefulness of the method.

Keywords: operation and management, large-scale server, dependence, maintenance notification

1. はじめに

近年，インターネットが広く普及し，我々が生活するうえで欠かせない重要なインフラとなっている．こ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，全体における各部の依存関係の把握は管理者でさえ困難になっている．そのため，障害発生時の影響範囲を特定する研究 [1] や，知識を記述することで自動的に障害を通知する研究 [2] が行われている．

[1] 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, Nomi, Ishikawa 923–1292, Japan
[2] 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, Nomi, Ishikawa 923–1292, Japan
[†1] 現在，日本電気株式会社
Presently with NEC Corporation,
[a)] shikida@jaist.ac.jp

一方，サーバ内部を構成する部品どうしの依存関係情報を自動収集・解析し，管理者のシステムへの理解度を深める支援手法の提案 [3] が行われている．さらに，動的な依存関係の抽出を可能にする提案も行われている [4]．また，抽出した依存関係を用い，障害発生時に影響する管理者を特定し，管理者全体へ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 [5]．しかし，これらの研究では管理者に対して適切な通知を行うことが目的であり，ユーザに対しての適切な通知は考慮されていない．

システムは大規模化・複雑化している．このような環境で，一部のメンテナンスを実施しようとした際に，どのユーザにどの程度の影響があるか特定することは困難である．たとえば，あるディスクアレイのリビルドが発生するメンテナンスを仮定する．ディスクアレイは通常 RAID などで冗長構成となっており，内部のハードディスク数台を交換する程度では，サービスを止める必要がない．ただし，リビルド中にパフォーマンスが劣化する場合も多く，このディスクアレイを利用しているユーザには注意喚起の通知を行うことが望ましい．このディスクアレイを利用しているサービスの一覧を特定し，このサービスの全利用者に対して通知を行うことは可能である．しかし，各ユーザが実際にサービスを利用しているかどうかは分からないため，適切なユーザのみに通知ができない．さらに，利用しているサービスが他のサービスに依存している場合も想定され，ディスクアレイを直接利用していないがサービスを介して間接的に利用しているユーザには通知が届いていない場合がある．

ユーザに通知する手段としてはメーリングリストを用いて行われる場合が多い [6], [7], [8]．組織で用いられるメーリングリストは組織に所属する全員を対象にするものから，部署や学科，研究室単位といったある程度細かい単位のものまで数多くある．利用の有無にかかわらず全員を対象としたメーリングリストを用いて頻繁に通知すると，受信側であるユーザは大量のメールを受け取ることになる．これは，ユーザにとって負荷であり，重要な情報が埋没化して見逃される可能性が高くなる．したがって，管理者は影響のあるユーザのみに通知しなければならない．

このように，ユーザに対する通知は，これまであまり考慮されておらず，実際の利用にそった適切な通知が行われていない．今後も，サービスの高品質化を目的として，システムがさらに複雑化することが予想され，ユーザへの通知がますます困難になる．

本論文では，サーバの部品依存関係と各サービスの利用履歴を用い，各ユーザに適切な通知を行う方式を提案する．この方式では，通知が必要なユーザを特定し，さらに，どのような影響があるかを一般ユーザに理解できる表現で説明し，その影響度合いに応じた手段での通知を可能にする．これによって，大規模な組織において様々なサービスが提供され，各ユーザの利用が管理者に把握できていない状況であっても，各ユーザに適したメンテナンス通知を行うことができる．さらに，本論文では，提案方式に基づくメンテナンス通知のシミュレーションを行い，本方式の有用性を議論する．

以下，2 章で関連研究について述べ，3 章では，本論文の先行研究であるサーバの依存関係抽出法について述べる．4 章では，サーバの部品依存関係を用いたユーザ指向通知方式を説明し，5 章で提案方式を用いた評価実験について述べ，6 章で本方式の有用性に関する議論を行う．

2. 関連研究

本章では，大規模システムの運用管理および通知に関連する研究について述べる．

文献 [1] のような，システム全体をモデル化することで，障害発生の影響範囲を特定する提案がされている．これは，小規模であればシステムのモデル化をすることは可能であるが，大規模なシステムをモデル化することは大変困難である．さらに，大規模なシステムは改良されたり，一部に変更を加えられたりした際も再度モデル化が必要であり，最新の情報をつねに把握し続けることは困難である．

文献 [2] では，サーバやクライアントごとに監視を行うサービスを稼働させる．この監視するサービスが障害を検知した場合，自動で他のサーバやクライアントを監視しているサービスに対して通知を行うシステムである．この研究も，小規模なシステムであれば自動で他の監視するサービスのルールを記述することが可能であるが，大規模になるとこのサービス自体の数も増えてしまい，膨大な量のルールを記述する必要がある．そのようなルールを記述することは管理者にとって大きな負担であり，またシステムに変更があった際のルール改変の手間を考慮すると大規模なシステムでの利用は現実的でない．したがって，大規模システムではユーザに適切な通知ができない可能性がある．

3. サーバの依存関係抽出法

本章では，本論文の先行研究として我々が提案したサーバの依存関係抽出法 [3], [4] について説明する．

近年，大規模なシステムを運用する組織においては，複雑なサーバ群を複数人の部署で集中管理する形態が増加している．このような組織のサーバ群は，ストレージサーバ，データベースサーバ，Web サーバ，各種アプリケーションサーバなどが，互いに依存関係を持つことが多い．その場合，一部のみを担当する管理者にとっては，全体システムの把握は困難であり，担当サーバ/サービスと依存関係があっても，さらにその先の依存関係先まで把握できない．

文献 [3] で提案した方式は，各種サーバや周辺機器の設定情報を収集し，またシステム設計者が情報を補足する．それらの情報をもとに各部の依存関係をすべて抽出し，各

```
ASM2:search>>> depend -v %DIR%is14e1%/home/i2008 %DISK%*%*
       CLASS:DIR HOST:is14e1 PATH:/home/i2008
100:3  CLASS:DISK HOST:fs90 DISKname:/dev/dsk/c3t0d1s2
66:7   CLASS:DISK HOST:fs8-dctl0 DISKname:RG-01
76:8   CLASS:DISK HOST:fs8-dctl0 DISKname:/dev/dsk/c1t0d1s2
70:14  CLASS:DISK HOST:fs8-dctl0 DISKname:/dev/dsk/c1t0d7s2

ASM2:search>>> depend -vod 76:8
    CLASS:DIR HOST:is14e1 PATH:/home/i2008
<-- CLASS:DIR HOST:fs90 PATH:/home/fs5001 (nfs_mount)
<-- CLASS:DISK HOST:fs90 DISKname:/dev/dsk/c3t0d1s2 (ufs_mount)
<-- CLASS:DISK HOST:fs8-dctl0 DISKname:RG-01 (SAN FibreChannel)
<-- CLASS:DISK HOST:fs8-dctl0 DISKname:/dev/dsk/c1t0d1s2 (RAID5)
```

図 2 依存関係の出力例

Fig. 2 The example of output of dependencies.

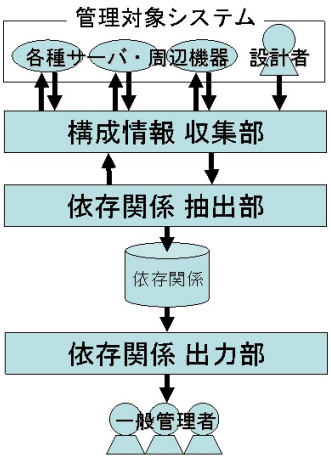

図 1 依存関係に基づく構成管理支援システムの概要 [3]

Fig. 1 The outline of configuration management system based on part dependence.

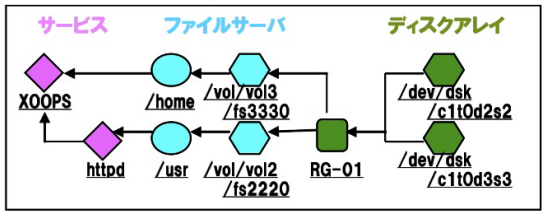

図 3 オブジェクトの依存関係例

Fig. 3 An example of part dependence on servers.

管理者がシステム全体の構成を把握することの支援を行うものである．この方式を用いた構成管理支援システムの概要を図 **1** に示す．

たとえば，ディスクドライブなどの物理デバイス，RAID ディスクなどの仮想デバイスや Web サーバなどの上位レイヤにおけるサービスなど，サーバを構成するハードウェア・ソフトウェアの部品を**オブジェクト**と呼び，様々なオブジェクトに関して依存関係の調査を行う．調査には，それぞれのオブジェクトに関係する設定ファイル，管理コマンドの出力結果の解析や，場合によっては設計を行った管理者が与える情報をもとに行う．

なお，各依存関係は**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を表す．逆に，依存度が 0 の場合は依存していない状態を表す．依存度は管理者のポリシによって定める．たとえば，RAID でディスクアレイを組んでいる場合は，冗長性があるため，1 より小さい値に設定する．

本方式を用いた構成管理支援の試作システムの出力結果を図 **2** に示す．これは，我々の組織で実際に稼動している高可用性クラスタストレージシステムの構成情報の一部を入力として与え，NFS クライアント上のディレクトリを指定して依存している全ディスクの一覧と，その一覧の 1 つの詳細を出力させた例である．

ストレージに関する依存関係のみでなく，物理デバイスからアプリケーションまで広範囲にわたる間接的な依存関係の検索も可能である．たとえば，図 **3** に示すようなオブジェクトの依存関係が把握できていると，ストレージサーバの管理者は，自分が管理するどのディスクドライブ内のデータがどのような重要サービスに利用されているのか知ることができる．また，Web 上のアプリケーション管理者も，自分が管理するアプリケーションのファイルがどのサーバのどの物理資源を利用しているか容易に把握できる．すなわち，この方式によってレイヤを越えた依存関係の把握をも支援可能である．近年はストレージの仮想化も進んでおり，このような依存関係は通常把握が困難であった．

また，これらの抽出された依存関係情報を用いて，障害通知に利用する提案 [5] も行われている．この提案では，障害を検出した際に他のどの部門のどのオブジェクトまで影響するか計算し，影響のない管理者への通知量を削減することで，情報の埋没化を防いでいる．しかし，ユーザに対する通知には適していない．これは，管理者とユーザの違いから発生する問題である．管理者は管理する部門が固定されており，基本的に自分の部門に関する情報を必要とす

る．しかし，ユーザはシステムの全体構成を把握しておらず，また臨機応変にサービスを利用することから，どの情報を受け取ればよいか，また管理者から見れば，どのユーザにどの情報を通知すればよいか分からないという問題がある．

4. サーバの部品依存関係を用いたユーザ指向通知方式

本章では，提案するユーザ指向の通知方式について述べる．

4.1 概要

提案手法は，サーバ部品依存関係部，利用履歴部，情報通知部の 3 つの部分で構成されている．これらの構成図を図 4 に示す．このように，ユーザはサービスを自由に利用でき，その履歴は利用履歴部に保存される．ここで，管理者は図中に「通知対象」と示した箇所に対してメンテナンスを行いたいとする．このメンテナンス箇所は，ユーザ側から見れば，間接的に利用しているサービスである．そのため，管理者はこういった依存関係を考慮してユーザに通知しなければならない．

4.2 直接利用頻度の算出法

利用頻度はオブジェクトごとに 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．サービスによって，データ量の差やある単一の目的のために複数回アクセスすることがありうるが，そのような差異を吸収して一定の利用と処理するためである．しかし，毎日同じサービスを利用していれば利用頻度が高いことになる．

次に，この利用履歴から，直近 1 週間の利用回数（n_w），直近 1 カ月（30 日）の利用回数（n_m），直近 1 年（365 日）の利用回数（n_y）を計算し，この 3 種類の値から式 (1) によって**直接利用頻度** p を算出する．ここで $r_1+r_2+r_3=1$ であるため，直接利用頻度 p は $0 \leq p \leq 1$ となる．なお，各項に付けられた重み r_i は管理者のポリシによって設定する．これによって，過去 1 年間の利用履歴を用いるが，各項で分母の値が異なるため 1 日の利用があった場合による影響が異なり，r_i が等しくても直近ほど重視することになる．なお，ユーザである人間の活動は 1 週間単位の規則性や 1 カ月単位の変動などもありうるため，各組織の活動の特徴から管理者が推測・設定しやすいという考えから，この 3 つの利用回数を用いる．ただし，週ごとの規則性が予想されるため，1 週間より近い期間は重視しない．

$$p = n_w/7 * r_1 + n_m/30 * r_2 + n_y/365 * r_3 \quad (\text{ただし，} r_1+r_2+r_3=1) \tag{1}$$

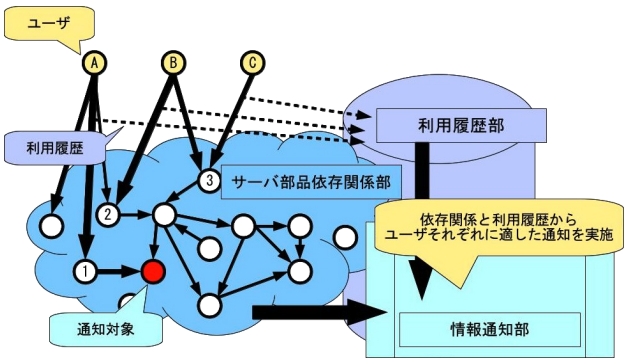

図 4 提案手法の構成図
Fig. 4 The overview of proposed method.

4.3 間接利用頻度の算出法

前節ではユーザが直接利用しているサービスの利用頻度の算出法を述べたが，依存関係と直接利用頻度を組み合わせることで，**間接利用頻度**を算出する方法を述べる．間接利用頻度は，サーバの部品依存関係の有向グラフを探索することで算出する．そのために，まず**依存関係テーブル**と**利用頻度テーブル**を作成し，それらから間接利用頻度を求める．

4.3.1 依存関係テーブルの作成

依存関係テーブルとは，オブジェクト間の依存関係や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情報である．システムの規模によるが，すべてのオブジェクト間の依存度をつね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，ユーザへの通知を行う場合に毎回作成し直す．図 5 のオブジェクト「/usr/bin」に影響が出るメンテナンスを実施する場合の依存関係テーブル作成例を表 1 に示す．

4.3.2 利用頻度テーブルの作成

利用頻度テーブルとは，各オブジェクトに関してユーザごとの利用頻度を 4.2 節の方式で算出した値である．ただ

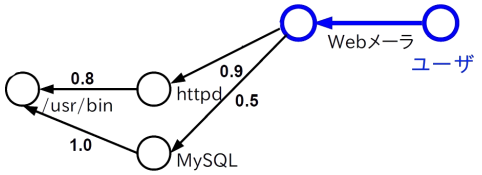

図 5 依存関係の例
Fig. 5 An example of part dependencies.

表 1 依存関係テーブルの例
Table 1 An example of part dependence table.

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

表 2 利用頻度テーブルの例
Table 2 An example table of direct use.

ユーザ	オブジェクト	利用頻度
A	Web メーラ	0.60
B	Web メーラ	0.10
C	Web メーラ	0.01

表 3 間接利用頻度テーブルの例
Table 3 An example table of indirect use.

ユーザ	間接利用頻度
A	0.43
B	0.07
C	0.01

し，依存関係テーブルの作成によってメンテナンスの影響範囲は特定できているので，その範囲内のオブジェクトについてのみ算出する．

利用頻度テーブルの例を**表 2** に示す．ここでは直接利用頻度 p（式 (1)）の各項の重み r_1，r_2，r_3 はすべて 0.33 とした．

4.3.3 間接利用頻度テーブルの作成

利用頻度テーブルに存在する各オブジェクトに関して，依存関係テーブルの中から最も依存度の高い経路を選択し，利用頻度と依存度を掛け合わせたものを間接利用頻度とする．

図 5 の例では，オブジェクト「/usr/bin」から「Web メーラ」までの依存度は 0.72 であるため，利用頻度テーブルにある値にこの依存度を掛け合わせて間接利用頻度テーブルを作成すると**表 3** のようになる．

4.4 通知内容の決定

依存関係と利用履歴を用いることで，通知内容をユーザごとに適切にする．複数人向けに作成された情報よりも，各個人向けに作成された情報の方が認知度が高まることが分かっている [9]．ユーザに分かりやすい通知内容は内容の理解を深め，情報の埋没化を防ぎ，メンテナンス通知の効果を高める．

たとえば，図 5 の例で，メンテナンス箇所を通知するだけではなく，「今回のメンテナンスにより，あなたの利用している『Web メーラ』に影響があります.」という説明を付加する．このような説明を，ユーザごとに，またそのユーザの間接利用頻度が高い各オブジェクトに対して記述する．

5. 評価実験

4 章で述べた提案手法に基づくメンテナンス通知機構の実験システムを実装した．このシステムを用いた実験とその結果について述べる．

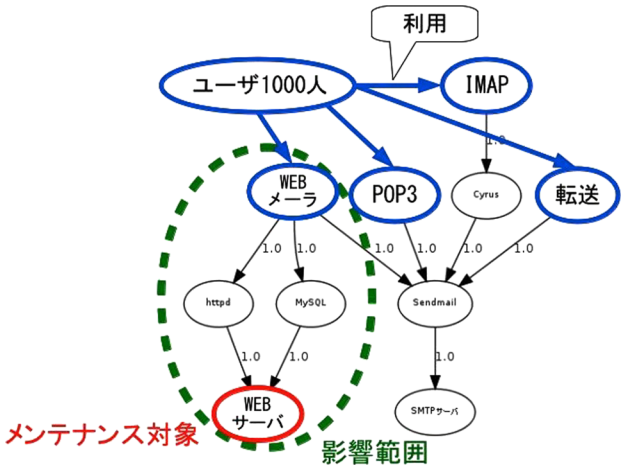

図 6 小規模なシステム例
Fig. 6 The overview of 1st example.

5.1 実験システムの実装

提案手法のうち，実験に必要な部分のみを実装した．本論文で提案する間接利用頻度算出機構は実装したが，先行研究である依存関係収集機構は作成しておらず，シミュレーションを行う対象システムに合わせてあらかじめデータベースに情報を格納しておく．また利用履歴についても同様に，各例に合わせて多数のユーザの履歴をデータベース上に生成しておいた．なお，ここでも直接利用頻度 p（式 (1)）の各項の重み r_1，r_2，r_3 はすべて 0.33 とした．

実装に用いた言語は Ruby 1.8.6，データベースは MySQL を使用した．実験は Debian GNU/Linux amd64 Squeeze を使用し，実験用インタフェースは CGI で作成したため，Ruby および MySQL を使用する CGI が動作する Web サーバも必要である．

5.2 小規模システムの例

最初に，通知の例を示すために小規模できわめてシンプルなシステム例を用いる．1,000 人のユーザが電子メールを受信するサービスを利用する場合を想定して実験を行った．これは**図 6** に示すように，POP，IMAP，Web メーラなど，類似サービスがあり，依存度はすべて 1.0 となっている．利用履歴は，ユーザごとに主として 1 つの方法でメールを受信していると想定して，乱数を用いて利用頻度情報を生成した．

この状況で Web サーバにメンテナンスを行う際の通知に関してシミュレーションを行った．4 章の方式に従って間接利用頻度を算出するが，依存があるオブジェクト間の依存度がすべて 1.0 であるため，グラフ上で経路のあるノード間の間接利用頻度もすべて 1.0 となる．したがって，計算過程の各テーブルは自明であるため省略するが，最終的に出力された通知例を**図 7** に示す．背景が白の行は通知する必要がないユーザで，背景が赤の行はサービスをよく利用しており，今回のメンテナンスで強く影響を受けると

0	s653	低 0.06336	WEBサーバに対してメンテナンスを行います．あなたは以下に影響が出ますが，最近の利用はありません． 影響が出る項目: Webメーラー (ただし，最近の利用頻度は低いです) 理由: s88はWebメーラーを使用されていますが，Webメーラーは今回メンテナンスのWEBサーバに依存している為影響が出ます．
0	s384	低 0.06336	WEBサーバに対してメンテナンスを行います．あなたは以下に影響が出ますが，最近の利用はありません． 影響が出る項目: Webメーラー (ただし，最近の利用頻度は低いです) 理由: s88はWebメーラーを使用されていますが，Webメーラーは今回メンテナンスのWEBサーバに依存している為影響が出ます．
0	s891	低 0.06336	WEBサーバに対してメンテナンスを行います．あなたは以下に影響が出ますが，最近の利用はありません． 影響が出る項目: Webメーラー (ただし，最近の利用頻度は低いです) 理由: s88はWebメーラーを使用されていますが，Webメーラーは今回メンテナンスのWEBサーバに依存している為影響が出ます．
0	s606	低 0.06897	WEBサーバに対してメンテナンスを行います．あなたは以下に影響が出ますが，最近の利用はありません． 影響が出る項目: Webメーラー (ただし，最近の利用頻度は低いです) 理由: s88はWebメーラーを使用されていますが，Webメーラーは今回メンテナンスのWEBサーバに依存している為影響が出ます．
0	s336	高 0.5181	WEBサーバに対してメンテナンスを行います．あなたは以下に影響が出ます． 影響が出る項目: Webメーラー 理由: s88はWebメーラーを使用されていますが，Webメーラーは今回メンテナンスのWEBサーバに依存している為影響が出ます．
1	s146	低 0.06897	WEBサーバに対してメンテナンスを行います．あなたは以下に影響が出ますが，最近の利用はありません． 影響が出る項目: Webメーラー (ただし，最近の利用頻度は低いです) 理由: s88はWebメーラーを使用されていますが，Webメーラーは今回メンテナンスのWEBサーバに依存している為影響が出ます．
1	s843	低 0.06897	WEBサーバに対してメンテナンスを行います．あなたは以下に影響が出ますが，最近の利用はありません． 影響が出る項目: Webメーラー (ただし，最近の利用頻度は低いです) 理由: s88はWebメーラーを使用されていますが，Webメーラーは今回メンテナンスのWEBサーバに依存している為影響が出ます．
1	s464	高 0.5181	WEBサーバに対してメンテナンスを行います．あなたは以下に影響が出ます． 影響が出る項目: Webメーラー 理由: s88はWebメーラーを使用されていますが，Webメーラーは今回メンテナンスのWEBサーバに依存している為影響が出ます．

図 7 提案手法による通知例
Fig. 7 A list of notifications for 1st example.

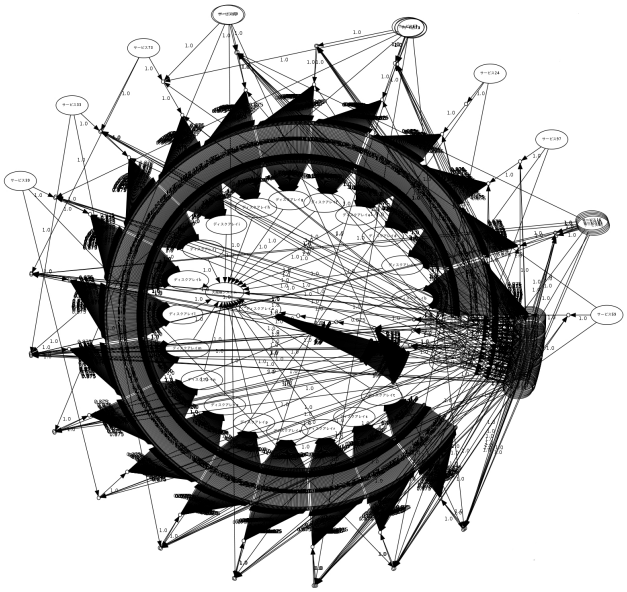

図 8 依存関係のグラフ描画例
Fig. 8 An example of dependence graph for large-scale system.

判断されたユーザである．

5.3 大規模システムの例

本節では，大規模なシステム上の一部に対してメンテナンスを行う際の通知を例に実験を行う．著者らの大学では，ハードディスクを数百個以上積載したストレージサーバを複数台運用しており，このサーバ内の領域を切り出した仮想ディスクを様々なサーバからアクセスしている．このような設計のシステムを模倣して依存関係を記述し，実験を行った．

図 8 にディスクが 1,280 台の場合の各オブジェクトの依存関係グラフの主要部分を示す．図中で円形になっている帯状の部分が 1,280 台のディスクである．実際のストレージ装置では，製品によって論理ユニットやプールなどの様々な概念があるため，さらにレイヤが増えるが，この例ではこれらのレイヤを省略して単純化してある．それでもこの規模で全体像を表示するとこの図のように細部が潰れてしまって不明瞭となる．このように大規模な例で管理者が設定情報をたどって各ユーザへの影響を推測することはきわめて煩雑な作業といえる．

このストレージを利用する様々なサービスを 1,000 人のユーザが利用する履歴をランダムに作成し，ある 1 カ所のディスクアレイをメンテナンスする場合でシミュレーションを行った結果を**表 4** に示す．全ユーザがそのディスクアレイに間接的に依存している度合いを高/中/低の 3 段階に分けて，その人数を示している．なお，この実験では「高」は利用頻度 0.20 以上，「中」は 0.08 以上 0.20 未満，「低」は 0 より大きく 0.08 未満の場合とした．このような区分は通知メディアを考えるうえで重要であるが，値の決め方に関しては 6.2 節で議論を行う．各サービスの利用者数やサービスが依存しているディスクアレイが異なるため，人数にもばらつきがある．この実験に用いた全オブジェクト数は 3,761 個であるが，算出に関係するのはメンテナンス対象のオブジェクトと直接または間接で依存関係があるオブジェクトのみであり，その数（表 4 の依存オブジェクト数）によって計算時間が異なっている．実験システムは Ruby での実装であるが，計算時間は 2.5〜4.5 秒程度であり，十分実用的な速度である．

6. 議論

本章では，実験結果に基づいて提案方式の有用性に関する議論を行う．

6.1 ユーザの視点

5.2 節の小規模システムの例では，ユーザによって利用している電子メール受信サービスが異なっている．したがって，一部のサービスにしか影響のないメンテナンスの場合には，他のサービスを利用するユーザには意味のない通知となる．ユーザの中には，学内では POP を使うが出張先や自宅では Web メーラを使うなど，1 つのサービスとは限らない様々なユーザがおり，通知すべきユーザを決定することは困難である．

著者らの大学では，このような場合でも多数のユーザに影響が予想されるならば全ユーザに通知を行っている．その結果，大規模集中化しているシステムの場合，ユーザには不必要な通知が多量に送られることになり，必要な情報

表 4 ディスクアレイのための通知時の算出結果
Table 4 The result of notifications for maintenances of disk arrays.

ディスクアレイ	利用頻度高	利用頻度中	利用頻度低	依存オブジェクト数	計算時間 (s)
/dev/md0	114	110	317	398	2.83
/dev/md1	176	136	585	585	4.10
/dev/md2	139	135	390	472	3.62
/dev/md3	173	132	429	584	4.13
/dev/md4	151	130	382	466	3.52
/dev/md5	129	122	353	440	3.27
/dev/md6	122	101	332	381	2.98
/dev/md7	91	80	258	327	2.33
/dev/md8	99	105	291	329	2.56
/dev/md9	111	111	312	359	2.77
/dev/md10	170	130	445	599	4.29
/dev/md11	95	91	260	296	2.26
/dev/md12	107	99	291	314	2.53
/dev/md13	153	120	384	464	3.56
/dev/md14	163	125	406	497	3.82
/dev/md15	207	157	491	606	4.56
/dev/md16	140	125	359	446	3.36
/dev/md17	143	125	386	505	3.69
/dev/md18	175	138	458	592	4.39
/dev/md19	178	157	449	542	4.29

の埋没化問題が発生する．実際，著者の周囲の学生ユーザにヒアリングしてみたところ，半数程度のユーザがメンテナンス通知メールをほとんど読んでいないことが分かった．その理由としては，「自分に関係ない」「よく分からない」といった意見が多かった．情報系の学生ではあるが，サービスを実現しているシステムの内部構成は知らないため，影響が理解できないことがあると考えられる．

これに対して提案方式では，どのユーザがどの程度利用しているかを算出することができる．その結果頻度の特に高いユーザにのみ電子メールによる通知を送信するので，不必要な通知が多量に送られることはない．また，利用頻度に合わせて，たとえば中程度の利用頻度であれば RSS フィードを利用したり，ポータルサイトで参照させたりし，利用頻度が小さければ Web ページのアナウンスのみとする，といった方法がある．通知手段を利用頻度に適したメディアにすることで，情報の埋没化を防ぎつつ情報の欠落もなくすことが可能である．なお，ユーザによってはできるだけメールで受け取りたい人や，逆にメールの数を減らしたいなど，ポリシが異なる場合があるが，現在の方式ではそのような個別対応は考慮されておらず，今後の課題である．

また，前述のヒアリング結果でも「よく分からない」という理由があったように，ユーザ向けの通知では，その内容の分かりやすさも重要である．著者らの大学では大規模集中化を進めたシステムを構築しており，たとえば，Web サーバのコンテンツの一部がデータベースサーバに依存していたり，また別のストレージサーバにデータを置いていたりするなど，きわめて複雑な依存関係がある．特に最近は仮想化技術の多用などにより複雑さは年々増しているため，管理者であってもしばしば見落としが起きている．すなわち，一般ユーザがまったく理解できていないことが当然のシステムになってきたといえる．

そのような状況下でも，従来のようにメンテナンス箇所や全ユーザ向けの主な影響を提示するのではなく，本方式では，各ユーザにそのユーザが最近よく使ったサービス名やファイル名などを提示して通知することができるので，システムをまったく理解できていないユーザであっても自分への影響を容易に推測することが可能になる．

6.2 管理者の視点

ユーザへのメンテナンス通知を行う作業は，管理者にとって煩わしい業務の 1 つである．障害通知は緊急性があり，ユーザに即影響があるため必要性が高いが，事前に行うメンテナンス通知は不必要なものが増えるとユーザが不満に思うことが分かっているため，通知範囲や手段を最適にする努力を行っている．影響範囲が自明なメンテナンスの場合には支障がないが，複雑な依存関係から各所に影響がある場合など，著者らの大学でシステムを運用するセンタでは，事前に定例会議で議論して通知方法や内容の表現を決定している．また，影響する利用者が少ない場合には，そのつどログから最近の利用者を調べて，そのメンテナンス用に新たなメーリングリストを作成している．提案方式

を利用して通知を行うことで，これらの煩わしい作業を自動化することが可能である．

ただし，利用頻度算出（式 (1)）の係数や，間接利用頻度と通知手段の対応は管理者が決定する必要がある．特に，間接利用頻度に応じて通知手段を決定するための境界値は，5.3 節の実験では 0.08 と 0.20 という値を用いたが，対象オブジェクトによって，たとえば，ほとんど全ユーザが毎日利用するようなメールサービスもあれば，ごく一部のユーザが稀に使う程度の特殊なサーバもあるため，オブジェクトごとに利用頻度の分布は大きく異なり，それぞれに適切な値を設定しなければならない．しかし，先に述べた著者らの大学のように，ユーザの特性を把握している管理者は通知に関するノウハウを持っており，間接利用頻度の分布を見て値を調整することは容易に習熟できると考えられる．また，いずれの問題もいったん適した値が決定できれば，類似の場合にも同じ値を用いて同様な通知が行えるため，徐々に省力化していくことが可能である．実運用環境で実験を行い，これらの値の決め方の指針を示し，また自動化を行うことが今後の課題である．

また，メンテナンス通知が埋没化してユーザに伝わっていないことで，トラブル対応の労力も増加している．当センタへの問合せ窓口には，メンテナンスを理解していなかったために当日になって「○○が動きません」といった問合せがある．これらのユーザ対応業務が削減できることも提案方式の利点である．

6.3 スケーラビリティ

大規模システムの実験では全ユーザ向け通知の算出に 2.5〜4.5 秒程度要した．この例ではディスクドライブが 1,280 個であったが，ここでさらに大規模になった場合を考える．大規模になってもある 1 つのサービスが使用しているディスク量に変化がなければ，依存しているオブジェクト数も変わらないため，計算コストの増大要因とはならない．ある 1 つのディスク装置のメンテナンスを考えても，そのディスク内のデータを利用しているサービス数はシステム規模が大きくなっても増えず，むしろ減ることが予想される．実際に，ディスク装置数を 2 倍にした例を作成して計算を行ったところ，メンテナンス箇所に依存するオブジェクト総数が若干減ったため，平均計算時間は約 0.2 秒短縮された．

ただし，メンテナンス箇所が増えて，それによる影響範囲が広くなった場合には計算量が増える．たとえば，計画停電により組織内の全システムが停止するような場合には計算量が極端に増大する．しかし，このように全システムをメンテナンスする場合には，全ユーザに一様に停電であることのみを通知するのが適切である．自分への具体的影響を多数列挙されると逆に理解しにくいといえる．したがって，このような場合は提案方式を利用する状況として想定していない．

一方，1 カ所のメンテナンスの場合で，ユーザ数が 10 倍の 10,000 人となると計算時間も比例して 10 倍になることが予想される．しかし，算出した結果により電子メールで直接通知すべきユーザもある程度の割合が存在することを考えると，それらのユーザへ個別メールを送信するために要する時間に比べれば十分小さい時間で算出できるといえる．

なお，従来は全ユーザに同一内容の通知を行っていたのに比べると，個別の電子メールを送信するコストはかなり大きいといえるが，各ユーザに適合した分かりやすい通知を行うためにはやむをえないと考える．前もって行うメンテナンス通知での利用を想定しているので，深夜に送信するなどの工夫を行えばメールシステムの負荷を軽減できる．昨今は，メールサーバの処理能力も増大し，たとえば，企業の顧客向けメールサービスでも，メール本文に顧客の氏名を入れたり，顧客に合わせた内容の送信を行ったりすることが一般的になりつつあるので，これは大きな問題とは考えられない．

7. おわりに

本論文では，サーバの部品依存関係と各サービスの利用履歴を用い，各ユーザに適切な通知を行う方式を提案した．この方式では，ユーザが直接利用したサービスの履歴から，間接的に利用しているサーバの部品を求め，それらの間接利用頻度をもとに，ユーザに適切な内容を適切な手段で通知することを可能とした．また，本方式に基づくメンテナンス通知のシミュレーションを行い，有用性を示した．

情報システムは今後も大規模・集中化が進み，これまで以上に複雑になることが予想される．ユーザにとっては利用しているサービスはどのようなシステムで実現されているのかますます理解できなくなる．その結果，管理者が発信したメンテナンス情報は一般ユーザにとってさらに理解しにくいものとなる．本論文の方式はそのような大規模システムできわめて有用となるであろう．

本論文では，実運用システムを参考に，生成した利用履歴でシミュレーションを行ったが，今後は実運用でのメンテナンス通知に利用して評価を行い，利用頻度算出の係数や通知手段を決定する値の決定法を明らかにする予定である．

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，情報処理学会研究報告 OS，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，情報処理学会論文誌，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，情報処理学会第 2 回インターネットと運用技術シンポジウム，pp.37–42 (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).
[6] 澤村博道：学生情報管理システム，筑波大学技術報告，Vol.30, pp.32–35 (2010).
[7] 内藤久資，山口由紀子：統合サーバの構築と運用，名古屋大学情報連携基盤センターニュース，Vol.7, No.2, pp.168–184 (2008).
[8] 江藤博文，只木進一：新しいメールアドレスの柔軟な運用に向けて，学術情報処理研究，No.12, pp.98–102 (2008).
[9] 長谷川祐司，竹内勇剛：公共空間における適切な情報伝達を実現する手法の効果検証，情報処理学会論文誌，Vol.48, No.12 (2007).

敷田 幹文 （正会員）

1965 年生．1995 年東京工業大学大学院理工学研究科情報工学専攻博士後期課程修了．博士（工学）．同年北陸先端科学技術大学院大学情報科学センター助手．2001 年同助教授．2011 年情報社会基盤研究センター准教授．大規模分散システム，グループウェアに関する研究に従事．ACM，電子情報通信学会，日本ソフトウェア科学会各会員．

藤澤 恵一朗

1986 年生．2011 年北陸先端科学技術大学院大学情報科学研究科博士前期課程修了．同年日本電気株式会社金融ソリューション事業本部．金融業界向け勘定系システム基盤の開発に従事．大規模サービスシステムに関心を持つ．