社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2006－DPS－129（8）
2006／11／17

サーバの依存関係を考慮したログ情報による障害管理支援の提案

後藤 宏志† 敷田 幹文‡

† 北陸先端科学技術大学院大学情報科学研究科
‡ 北陸先端科学技術大学院大学情報科学センター
†‡ 〒923-1292 石川県能美市旭台 1-1
E-mail: †‡{h-gotou, shikida}@jaist.ac.jp

あらまし 近年，企業や大学のシステムは大規模化かつ複雑化が進み，安定した運用が望まれ，障害発生時のダウンタイムを最小時間に抑えなければならない．よって障害を検知してから早期に解決するためのログ情報による調査が重要になる．しかし，依然としてログ情報の調査には，システムに熟知した上級の管理者のスキルや経験に依存している現状がある．そこで本稿では，サービスが複雑に連携しているサーバの各ログ情報同士の依存関係を抽出する障害管理支援を提案する．大規模かつ複雑なシステムにおいて，システムに不慣れな初級の管理者を対象に，システム内の膨大な量の全ログ情報から，障害の原因箇所と関係のあるログ情報の候補を提供する．このことにより，障害の原因箇所が担当している管理区域内なら対策復旧を施し，管理区域外なら連絡通知するといった判断が可能となる．
キーワード ログ情報, 障害管理, 運用コスト

Proposal of Supporting Fault Management Using Log Information Considered the Dependency of Servers

Hiroshi GOTO† and Mikifumi SHIKIDA‡

† School of Information Science, Japan Advanced Institute of Science and Technology
‡ Center for Information Science, Japan Advanced Institute of Science and Technology
†‡ 1-1 Asahidai, Nomi-shi, Ishikawa, 923-1292 Japan
E-mail: †‡{h-gotou, shikida}@jaist.ac.jp

Abstract In recent years, it is expected that a system is managed stably and downtime at failure has to be kept the minimum, because the system is becoming a large scale and complication in universities or companies. Thus a survey which is based on log information is important to solve problems at early stage. However, pragmatic survey of log information depends on skills and experience of expert. This paper proposes fault management support which extracts dependence of log information of servers which coordinate services complicatedly. It provides low level administrators with cause of fault and a list of the log information related to it from all log information in large scale and complicated systems. By this support, administrators can judge whether the cause of the fault is in their administration district or not and the process of the recovery becomes more smooth.
Keywords Log Information, Fault Management, Management Cost

1　はじめに

近年，企業内や大学内のシステムは，ネットワークサービスを介し，数千台のクライアントと複数のサーバ群から構成される巨大なシステムとなってきている．また，24 時間 365 日の安定したサービスの運用が望まれており，障害が起った場合のダウンタイムを最小限にとどめることが必要とされている．よって，障害を検知してから，早期に障害原因を把握するための調査が重要になり，調査にはログ情報が必要不可欠とされている．

現在，運用されているシステムでは，大規模化

に伴い各ホストやネットワーク機器から何万行という膨大なログ情報が出力される．それらのログ情報は互いに関連性があるにもかかわらず，根本的に解決する方法は未だ見当たらない．そのため運用管理の現場においては，システムに精通している上級管理者のスキルや経験を元に，ログ情報の関連性を推測し，障害調査が行われている．しかし，スキルや経験のない初級の管理者にとって，管理区域外のログ情報の中から障害原因の特定に必要なログ情報を発見することは難しく，安定したシステム運用を妨げる一因になっている．

そこで本稿では，ログ情報による障害管理支援として，初級の管理者でもシステム全体のログ情報から，障害原因の候補となるログ情報を容易に把握することが可能な方法を提案する．

本稿では，2 章において従来の手法の特徴と問題点に触れ，3 章では提案方式について述べ，4 章で具体的な動作例を示し，5 章では議論を行う．

2　関連研究

従来の製品化された統合運用管理ツールとログ情報解析に出現頻度を利用した方式について，特徴と問題点を述べる．

2.1　既存の統合運用管理ツール

既存の製品化された統合運用管理ツール [1] では，各計算機及び周辺機器ごとのログ情報を収集し一元化するに留まり，システムに熟知し経験を持った上級管理者が，関連するログ情報を時系列毎に手動で比較する手法や特定の語句やパターンでマッチングした箇所を調査する手法がとられている．これらの手法は，自分が担当している管理区域外について知識や経験の浅い初級の管理者にとっては，特定の語句やパターン自体を思いつくのが困難だという問題がある．

2.2　ログ情報の解析に出現頻度を利用した方式

文献 [2, 3] は，複数の計算機や機器にログ収集管理サーバを配置し協調動作させログの統一管理が可能なシステムを提案し，ログ情報の解析に単語の出現頻度に基づく辞書を用いた．文献 [4, 5, 6] は，ログ情報にテキストマイニングを用い，異常事象は極少数であるという理由から出現頻度の低い情報に着目した．これらの出現頻度に着目する手法は，異常を検出させる点では有効であるが，障害の原因調査という点では有効ではない．また，システム内の全ログ情報の関連性を示すことは難しく，全て一人で管理する小規模なシステムでは有効であるが，複数の区域を複数人で管理するような大規模なシステムでは有効ではない．

3　サーバの依存関係を考慮したログ情報による障害管理支援

本章では，前章で述べた従来手法の問題点を解決するため，サーバの依存関係を考慮したログ情報による障害管理支援について述べる．

3.1　システム構成の依存関係の利用

本提案方式では，先行研究 [7] のシステム構成の依存関係を利用する．この研究は，大規模かつ複雑なシステムにおいて，システム内の各計算機や周辺機器から構成する情報を収集，解析することで，ディスクや仮想ディスクといった低いレイヤから，アプリケーションが使用しているディレクトリやサービスといった高いレイヤまでの依存関係を抽出し，図 1 のような形で構成情報の依存関係を示してくれる．このことにより，例えば，Web アプリケーションを担当している管理者が，Web アプリケーションが使用しているディレクトリは，中央データセンターのディスクアレイ装置のどのディスクに依存してるのかという，システム全体の複雑な構成情報を容易に把握可能となる．この依存関係はオブジェクトと呼ぶ構成情報であり，具体的にはディスクや仮想ディスク，ファイルシステム，ディレクトリ及びサービス等をオブジェクトとしている．本提案方式では，この方式によって抽出されたオブジェクトの依存関係を利用する．

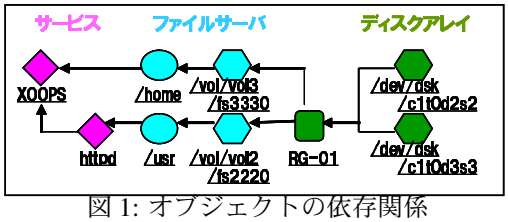

図 1: オブジェクトの依存関係

3.2　提案方式の概要と構成

はじめに前節で述べたシステム構成のオブジェクトの依存関係を利用する．次にオブジェクトと関連するログ情報をリンク (図 2) させ，そのリンクの関係をたどることでログ情報間の依存関係を示す (図 3)．管理者はシステム内で障害を確認すると，そのログ情報とリンクから，障害の原因と考えられるログ情報にたどりつくことで，障害調査において早期的な解決が可能となる．本提案方式はログ情報収集部，リンク情報作成部，出力部の 3 つの処理部から構成される．以下に各処理部の詳細について述べる．

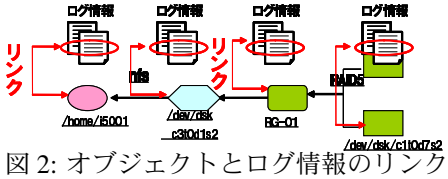

図 2: オブジェクトとログ情報のリンク

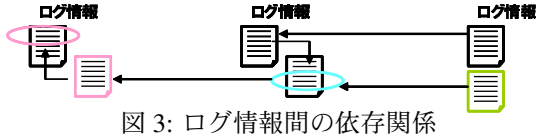

図 3: ログ情報間の依存関係

3.2.1 ログ情報収集部

ログ情報収集部は，次節で述べるリンク情報作成部からログ情報の収集に関するリクエストを受け取り，各計算機からリクエストに対応するログ情報を収集し，統一した形式に変換を行い，その結果を返す．ログ情報収集部は 2 つの処理部から構成される．ログ情報対応モジュールと汎用ログ形式変換部である．前者は，各ホストやネットワーク機器から各種ログファイル，ログ行を収集する役割を持つ．ログファイルは，OS の種類やバージョン，システム構成毎にファイルパスやファイル名が異なる．モジュールを種類毎に用意し，違いを吸収する．後者は，汎用的なログ形式を定義し様々なログ出力形式を一定のフォーマットに統一する．ログ行もログファイル同様，OS の種類やバージョン，ベンダ独自形式などによって出力形式が異なるので，その差異を吸収する役割を持つ．

3.2.2 リンク情報作成部

リンク情報作成部は，オブジェクトとログ情報をリンクさせる処理を行う．システムを構成している全オブジェクトとシステム内の全ログ情報との間でリンクが可能かどうかのマッピングを行い，可能ならばリンク情報を作成する．オブジェクトには，一意に識別する事が可能なキーが定められている．そのキーを構成している各キー値が，ログ事象のなかに出現しているかどうかでリンクが可能かどうかの判断を行う．ここでいうログ事象とは，ログ行のログ ID やプロセス ID が同値な複数のログ行，つまりあるサービスにおいて 1 事象として複数行に渡って記録したログ情報と定義する．次節で述べるオブジェクトの種類とログ情報の種類の関連表を参照し，オブジェクトがどのログ情報をサーチ対象とするかを決定する．サーチ対象のログ情報が決定したならば，それぞれのログ情報に対して順にオブジェクトのキー値でサーチを行い，リンク情報を作成する．

表 1: ロググループとオブジェクトの種類分類表の例

オブジェクトの種類	ロググループ
Service.sendmail	syslog.sendmail
Service.httpd	messages.httpd access_log error_log
FileSystem.fs40	messages.fs40
Disk./dev/dsk/c1t0d4	messages./dev/dsk/c1t0d4

3.2.3 出力部

出力部は，リンク情報，オブジェクト情報及び依存関係情報が格納された各データベースから，ログ情報同士の依存関係を管理者が必要とするログ情報のみを出力する処理を行う．その際，キー値の出現率を基にリンク情報の精度を考慮し，出力すべきログ情報の判断を行う．

3.3 ロググループとオブジェクトの種類の分類表

本提案方式では，ログ情報とオブジェクトの種類毎の関連性を示す分類表を作成する．分類表は，リンク情報を作成する際にオブジェクトがサーチ対象とすべきログ情報を絞り込むための用途に使う．これは従来方式において，管理者が単一の計算機上の複数のログファイル，及びログ行を大まかに分類し判断していた作業に相当する．分類表はシステムに熟知した管理者が，初期設定時に記述するものとする．分類表の例を表 1 に示す．

オブジェクトの指定は，オブジェクトの種類とキー値の 2 つの値で指定する．オブジェクトの種類にはディスクや仮想ディスク，ファイルシステム，サービス毎の値を用い，キー値にはオブジェクトのキー値での指定を行う．ログ情報は，ロググループと呼ぶ単位での指定を用いる．ロググループとは，特定のログファイルから特定のキーワードで抽出した複数のログ行と定義する．特定のファイ名と抽出するキーワードで指定する．この指定方法により，より詳細で柔軟な指定が可能となる．

3.4 リンク情報作成部の処理の流れ

オブジェクトとログ情報がリンク可能かどうかは，オブジェクトを一意に特定するためのキー値が，ログ行の中に全て出現しているかどうかで判断を行う．システム内の全オブジェクトのキー値で全ログ情報に対してサーチを行い，その結果を

リンク情報としてデータベースに保存しておく．

大規模なシステム内では，何万個というオブジェクトと何万行というログ情報が存在することが想定される．その際，管理者が使用する度にサーチを行いリンク情報を作成し，その結果を返していては，近年の計算機が高速になったとはいえ，大幅な待ち時間が予想される．また，ログ情報というものは，システムが稼働する限り情報が刻々と追記され，動的に増加するものである．予めリンク情報を作成していたとしても，管理者が使用する際には，新たに追記され増加した分のログ情報についてもリンク情報を作成し直す必要がある．

そこで本提案方式では，これらの相反する両者に対応するため，以下の 2 つの手法を用いる．

- 予め全オブジェクトに関するログ情報のリンク情報を作成
- オンデマンドで依存関係のあるオブジェクトのみリンク情報を作成

過去のログ情報に関しては，一度リンク情報を作成すれば，その後ログ情報が変化することは無い．よって，リンク情報を作成する際の対象とするログ情報は，新たに増加した分だけとする．差分更新の有無に関しては，以前チェックした最終の行番号を用いる．

3.4.1 オブジェクトとログ情報のリンク情報作成手順

はじめに，1 つのオブジェクトを引数として受け取る．次にそのオブジェクトがどのような種類のログ情報と関連性を持つのか，前節で述べた分類表を参照し，ロググループを決定する．前回，リンク情報を作成した時点からの差分更新の有無を確認するために，オブジェクトが前回までにチェックしたロググループの最終行と，現行のロググループログ行の最終行を比較する．もしロググループに新たに追加記録されたログ行があるなら，その追加分のログ行に対して，汎用ログ形式への変換とログ ID・プロセス ID を基にしたログ事象の連結処理を行う．その後，オブジェクトのキー値でログ事象をサーチし，キー値の出現率を求め，リンク情報をデータベースへと記録する．最後に，オブジェクトに対するロググループの最終チェック済行番号を保存する．またリンク情報として以下の値を記録する．

- オブジェクト番号
- ロググループ名
- 最終チェック済行番号
- キー値出現率

アルゴリズム - makelink(object)
配列:
(1) loggroup[ ]-ロググループ
(2) key[ ]-キー値
関数:
(1) GetLastLineNum( )-最終行番号を得る
(2) LineSearch( )-ログ事象にキー値が存在するかサーチ
begin
  loggroup[] ← 分類表参照 (object)
  foreach loggroup[i] **do**
    最終行 ← GetLastLineNum(loggroup[i])
    汎用ログ形式への変換
    ログ事象への複数行連結処理
    現在行 ← 最終チェック済行 [object][i]
    while(現在行 ＜ 最終行)
      line ← ログ事象の読み込み
      foreach key[j] **do**
        if(**true**)==(LineSearch(line, key[j]))
          キー値出現率の計算
          write LinkDB(object, loggroup[i], 現在行, キー値出現率)
    最終チェック済行 [object][i] ← 現在行
end

3.4.2 オンデマンドで依存関係のあるオブジェクトを対象としたリンク情報作成手順

管理者が使用する際に検索入力値として，ホスト名やサービス名，ディレクトリ名といったオブジェクトのキー値，もしくは，ログファイル名とログ行番号等のログ情報を求める．それぞれの場合に応じ，オブジェクト，リンク情報の各データベースに問い合わせ，その検索入力値と最も関係のあるオブジェクトの候補を割り出す．次に，依

アルゴリズム - makelink_depend
配列:
(1) object[ ]-オブジェクト
(2) depend_object[ ]-依存関係があるオブジェクト
関数:
(1) Ask{Object, Link, Depend}DB( )-{ オブジェクト，リンク情報，依存関係 }DB への問合せ
begin
  input ← 検索入力値の読込
  if (input == オブジェクのキー値)
    object[ ] ← AskObjectDB(input)
  elsif (input == ログファイル名と行番号)
    object[ ] ← AskLinkDB(input)
  depend_object[ ] ← AskDependDB(object[ ])
  foreach object[ ] **do**
    call makelink(object[i])
  foreach depend_object[ ] **do**
    call makelink(depend_object[i])
end

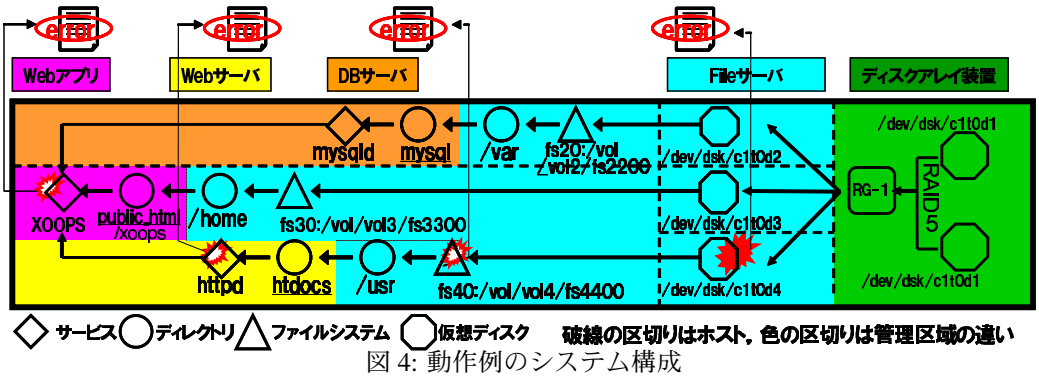

図 4: 動作例のシステム構成

存関係データベースに，候補のオブジェクトと依存関係のあるオブジェクトについての問い合わせを行う．候補のオブジェクトと依存関係のあるオブジェクトそれぞれに対し，ログ情報の差分更新があるかどうかを確認する．

4　提案手法の動作例

本章では本研究室で実際に稼働しているシステムにおいて，提案方式を使用した動作例を示す．

図 4 のシステムについて述べる．ディスクアレイ装置は各 RAID グループで構成されており，RAID グループ (RG-1) 内は多数の固定ディスク (/dev/dsk/c1t0d{1, 2}) で構成する複雑な階層構造となっている．ファイルサーバが 3 台稼働し，SAN を通して RG-1 をマウントしている．データベースサーバ (mysqld) が稼働しているホスト，Web アプリケーション (XOOPS) と WWW サーバ (httpd) が稼働しているホストが，ファイルサーバ 3 台の各仮想ディスク (/dev/dsk/c1t0d{2, 3, 4}) をファイルシステム (fs20, 30, 40) としてマウントしている．2 台のホストは，各ファイルシステムをディレクトリ (/var, /home, /usr) としてエクスポートし，各サービス (mysqld, XOOPS, httpd) がディレクトリを使用している．XOOPS は httpd 上で動作し，mysqld に依存している．また XOOPS のデータは mysql ディレクトリに格納している．破線の区分はホスト，色は管理区域の違いを現している．

このように互いにレイヤの異なるサービスが複雑に連携し，依存関係が深いシステムにおいて，1 台のファイルサーバとディスクアレイ装置を結ぶ SAN のファイバーチャネルスイッチの故障で，仮想ディスク/dev/dsk/c1t0d4 のマウントができなくなる障害が発生したとする．それが原因で，XOOPS にクライアントがアクセスしても画面が表示されない障害が発生し，XOOPS の管理者はエラーのログ情報出力を確認した．しかし，エラーのログ情報の記述内容では，その根本となる原因箇所が XOOPS 自体なのか，依存先である mysqld が動作しないからなのか，httpd の設定の問題なのか，又はファイルサーバの人的ミスか，ディスク本体のハードウェア故障なのか障害の原因箇所を特定できない．つまり，依存関係が深ければ深い程，障害の原因箇所の候補は増加する．また，このようなシステムでは複数人の管理者が管理区域を設け，分担して管理している．そのため，原因箇所の追跡には，他の管理区域でのサービスやログ情報に関する詳細な知識や経験が必要とされるため，さらに原因箇所の特定は困難になる．

本提案方式では，XOOPS の管理者が XOOPS によって出力されたログ情報のエラー箇所 (ログファイル名とログ行番号)，あるいは「XOOPS」というオブジェクトのキー値を入力値として，検索を行う．オブジェクト，リンク情報を格納してある各データベースから入力値でサーチが行われ，候補となるオブジェクト XOOPS が決定する．次に，XOOPS が依存している全オブジェクトがデータベースから検索され，結果のオブジェクト (18 つ) にたいしてログ情報の差分更新を行い，リンク情報を作成し直す．オブジェクト (18 つ) のログ情報の中から，エラーが出力されている 4 つのオブジェクト (XOOPS, httpd, fs40, /dev/dsk/c1t0d4) のログ情報を障害の原因候補として，XOOPS の管理者に向けて出力する (図 5)．XOOPS の管理者は出力結果から，エラーのログ情報が連動していることを察知し，ファイルサーバのオブジェクト/dev/dsk/c1t0d4 が根本の障害原因であると判断可能になる．

5　考察

本章では，サーバの依存関係を考慮したログ情報による障害管理支援について議論を行う．

本提案手法は，システム内の依存関係を構成する各オブジェクトとログ情報をリンクさせる事により，障害発生を示すログ情報と障害原因にあたるログ情報を関連づけた．その事によりシステム

```
XOOPS: /home/h-gotou/public_html/xoops/log/xoops.log
    123: Aug 22 06:00:03 error
httpd    : /usr/local/apache/logs/error_log
    224: Aug 22 06:00:03 error: File does not exist
           /home/h-gotou/public_html/xoops/file1.html
fs40:/vol/vol4/fs4400: /var/log/meesages
    292: Aug 22 06:00:03 /usr: nfs server not responding,still trying
FileServer /dev/dsk/c1t0d4: /var/adm/messages
    243: Aug 22 06:00:02 hard error reading fsbn 360723
```

図 5: ログ情報の出力例

全体での障害管理に必要なログ情報による調査支援を可能とする．これは複数の計算機が互いに連携してサービスを提供している複雑な関係を持つシステムや，複数の管理区域や管理者によって分担されているような大規模なシステムにおいて，ログ情報から障害の原因を調査する上で有効である．

従来の方法では，grep 等によるキーワード検索，高度なルール記述やパターンマッチングを用いるため，管理者が障害の場合に応じて複数のログファイルを開き，経験や知識からキーワードを推測し検索しなければならない．しかし，初級管理者は管理区域外のログ情報についてのキーワード自体を推測することが困難であり，負担を強いられる．障害の発生箇所から原因箇所までの依存関係が遠ければ遠い程，原因の解決はさらに困難になる．

しかし，本提案方式では障害発生箇所のログ情報から，関係するログ情報を自動で追うことにより障害原因箇所と思われるログ情報の候補を示してくれる．表示された候補が，依存している順に連続してエラーログ情報が表示されていれば，管理者は障害の原因箇所を判断しやすくなる．つまり自分の管理区域内でのエラーログ情報が，管理区域外のどの部分のエラーログ情報と関係しているか把握できる．また，管理区域内で発生している障害が，他の管理区域外に提供しているどのサービスにまで影響しているかどうか把握でき，連絡通知を行うことも可能である．この支援により，初級管理者でも自分が担当する管理区域内の原因対策をすべきなのか，他人が担当する管理区域外に連絡通知すべきなのかという判断が容易になる．

しかし，問題点も含んでいる．本手法では多様化するシステム構成に対応するためのモジュールを事前に複数用意し，特異なシステムに合わせて記述する必要がある．また，ロググループとオブジェクトの種類を分類し記述する必要もある．この煩雑な作業は，従来の上級管理者の経験や知識に相当するものである．だが，一般的なシステムにおけるモジュールは共通に使う事が可能である．

また，ログ情報とオブジェクトの種類分類において，指定方法をさらに細かく分類し記述することにより，検索すべき対象のログ情報をさらに削減することが考えられる．例えば，ログファイルから特定のキーワードで抽出するだけでなく，出力される行のパターンやエラーの種類毎に分類することなどが挙げられる．

6 おわりに

本稿では，先行研究のシステム構成の依存関係を用い，オブジェクトとログ情報をリンクさせ，ログ情報間の依存関係を示して障害管理を支援する方式の提案を行った．初級管理者に対し障害と関係のある絞り込まれたログ情報を提供し，障害管理の調査負担を軽減することが可能である．今後は，分類表において，エラーの出力パターンを詳細に分類することで，サーチ対象とするログ情報を削減する支援を行う．

参考文献

[1] 株式会社日立製作所, “HITACHI JP1”, http://www.hitachi.co.jp/Prod/comp/soft1/jp1/

[2] 神尾 正和, 石田 常竹, 箱田 貴久, “分散データベースを用いた大規模ログ管理システム”, 電子情報通信学会，データ工学ワークショップ論文集， pp.1-8, 2005.

[3] 神尾 正和, 石田 常竹, “ログの統一管理及び異常検出に関する研究”, 情報処理学会研究報告, コンピュータセキュリティ研究会, pp.77-82, 2004.

[4] 江端 真行, 小池 英樹, “不正侵入調査を目的とした複数ログの時系列視覚化システム”, 情報処理学会論文誌, Vol.47, No.4, pp.1099-1107, 2006.

[5] 高田 哲司, 小池 英樹, “ログ情報視覚化システムを用いた集団監視による不正侵入対策手法の提案”, 情報処理学会論文誌, Vol.41, No.8, pp.2216-2227, 2000.

[6] 高田 哲司, 小池 英樹, “見えログ: 情報視覚化とテキストマイニングを用いたログ情報ブラウザ”, 情報処理学会論文誌, Vol.41, No.12, pp.3265-3275, 2000.

[7] 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.41, No.12, pp.940-948, 2005.