

サーバの依存関係を利用したシステム構成管理の支援法

森 一[†], 敷田幹文[‡]

概要

近年、学校内や企業内でのネットワークは大規模化、複雑化が進み、各業務に情報ネットワークの利用は不可欠となっている。このようにネットワークの需要が増すにつれサービスを提供するシステムは大規模・複雑化する傾向にある。それに伴い管理には高度な知識や豊富な経験が必要となってきた。しかし、システム運用の環境によってはこれらの変化に管理者の理解度がともなわず、システム管理の大きな負担になっているという現状がある。本稿では、対象の各計算機・周辺機器からシステムを構成する情報を収集し、解析することでシステム内に存在するサーバの依存関係の抽出を行う。これら依存関係に関する情報を元に、管理者がシステム構成を把握するのに必要な情報を提供する。これによって、管理区分が異なっていたり、システムに熟知していない管理者に一律なシステム全体への理解度を与えることを目的としている。

The method of supporting system configuration management for using the dependency of a server.

Hajime Mori[†], Mikifumi SHIKIDA[‡]

Abstract

In recent years, the network system are becoming large scale and complicated in the school or the company. The network must be in each business. The demand of a network is increasing. The system which offers service tends to become on a large scale or complicated. Therefore, advanced knowledge and abundant experiences are needed for management. However, depending on the environment of the system management, an administrator cannot understand change of the system. it is the big burden of the system management. In this paper, the information which constitutes the system from each target computer and peripheral equipment is collected and analyzed. The dependency of the server which exists in the system is extracted. Information required for an administrator to understand the system configuration is offered using dependency information. This gives the same degree of comprehension as the administrators who do not understand the system.

1 はじめに

近年、コンピュータの普及、及びネットワーク技術の進歩に伴い学校内や企業内での各業務に情報ネットワークの利用は必要不可欠となっている。こ

れらネットワークシステムを用いて多岐にわたる様々なサービスを提供するために、複数のサーバが稼働している。こうしたネットワークサービスの需要に伴い、組織内ネットワークでのシステムの大規模化 [1] 複雑化が進んでいる現状がある。

[†]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
[†]School of Information Science, Japan Advanced Institute of Science and Technology
[‡]Center for Information Science, Japan Advanced Institute of Science and Technology

特に重要なサーバなどに関しては安定した稼働や、高い信頼性を求められるため、組織内ネットワークにおいて、複数の計算機間、周辺機器間にまたがって存在したり、幾つものサーバ同士が連動し

稼働するなど複雑な構成をもって運用されている。こうしたシステム内での複雑なサーバ間の関係を把握することは一般の管理者には困難となっており、システム構成の管理にかかるコストや負担は増加する傾向にある。

従来、システム構成の把握はシステム全体のポリシーなどを既に理解できている管理者によって、計算機本体やその周辺機器などの情報を収集することによって行われてきた。これはネットワークが小規模であり、管理者が少数の場合は有効な手段である。しかしながら大規模化、複雑化したシステムが広く普及している現状では、全ての管理者が管理対象のシステムに対する専門的な技術と適切なスキルを持っていない場合も存在する。このような場合、専門的知識を持つ上級の管理者は、自分の管理区分だけでなく他の不慣れた管理者のフォローを行う必要がある。

そこで本稿では、システムの運用管理において、全ての管理者が基本的に把握している必要があるシステム全体の構成管理に関する支援法を提案する。これは対象システムの構成を把握するのに必要な情報を収集し、それらに関連付けて管理者に提供を行うことによって、レベルの異なる管理者へシステム全体に対する一様の理解度を与えることを目的としている。

以下、2章において従来の手法を考察した後、3章で提案手法について述べ、4章でその動作例を示し、5章では議論を行う。

2 従来の手法

現在、一般的なシステム全体の構成把握は、各計算機・周辺機器単体ごとの情報を、システムツール等 [2] を用い収集を行い、管理者がそれらの情報から総合的に判断している。またネットワーク機器に関しては SNMP などの管理プロトコルや、ベンダーごとの独自の技術、またそれらを統合した構成の把握 [3] 等が用いられている。これらの方法は各周辺機器・計算機単体やネットワーク機器に関する管理においては十分に有効である。

しかしながらシステム全体を見た場合、どのサービスによって計算機・周辺機器間が依存関係を持っているのか、また逆にサービスがどの部分に依存して稼働しているか等の各計算機・周辺機器をまたいだシステム内での広域な関係に関しては、上記の

方法だけでは容易に判断することは困難である。そこで、これらの関係が複雑に組み合わせられた問題に関しては、事前にシステム構成を理解している管理者の判断に頼っているのが現状である。これらの管理者は、各システムツール・管理プロトコル・ベンダーごとの独自の仕様等の専門的な知識と、システム運用の豊富な経験によりの確なシステムの構成把握によって問題を解決し、柔軟な運用管理を行っている。

このような従来の管理者の経験と能力に依存したシステム構成把握の方式は、現在増加傾向にある複雑かつ大規模なシステムなどにおいては適しているとは言えなくなっている。その理由として以下の点があげられる。

まず大規模・複雑なシステムにおいて、一部の管理者の能力に頼ったシステム構成の把握は、管理対象が管理者の能力ではカバーできないほど広範囲となり過度な負担となっている。

またシステムが大規模になり、企業や学校などの業務などに密接に使用されるのに伴い、システムの運用管理を業務ごとやサービスごとなどに分割し、複数人の管理者によって区分を分担し管理する方式がとられている。この場合、各管理者は自分の管理区分に関してはシステム構成を把握しているが、関連を持つ他の管理者の区分に関しては把握するのは困難である。このことは、具体的に自分の管理する計算機・周辺機器と、他の管理区分の機器がどのような依存関係を持つのかを判断が行えず、これによって生じるシステム全体の理解度の違いにより運用管理に支障がでている。

また各区分ごとの管理者を統括する部署では、管理者ごとにシステム全体の構成把握の理解に対してばらつきがあるため、障害報告などの管理情報を適切に共有する [4] 等の対応が必要である。

以上のように、組織内ネットワークの拡大にともない大規模化、複雑化していく現状においては、従来方式ではシステムの全体構成に関する理解度は管理者のレベルによって異なっている。その結果、システム構成への理解度の高い管理者へ、運用管理に関する問題が集中する傾向がある。こうした複数の管理者の理解度の違いと、上級の管理者への負担の増大は、全体の運用管理に影響を及ぼし、ひいては利用者へ提供するサービスの品質低下につながる重要な問題となっている。

3 システム構成管理の支援法

本章では、レベルの異なる管理者に一樣なシステム構成管理の理解度を与えるための支援として、サーバの依存関係を利用した構成管理の支援法について説明を行う。

3.1 提案手法の構成

従来の問題で述べた通り、高い専門知識を持つ管理者に負担の増加する傾向にある原因の一つが、レベルの異なる管理者のシステムへの理解度の不足である。

そこで、システム構成管理の理解度の向上を支援する方法として、サーバの依存関係を利用し、システム内の計算機・周辺機器からシステムを構成する情報を自動及び一部手動で収集する。そこから依存関係に関する情報の抽出を行い、管理者に提示する方式の提案を行う。

これにより、知識や経験などの管理レベルの異なる管理者でも、共通のシステム構成に関する知識を共有することを可能とする。

本方式はサーバ間の依存関係に基く静的なシステム構成を対象とし、ターゲットとする領域は計算機・周辺機器のサービスによる関係のみとしている。そのためネットワーク越しのクライアント・サーバ間の関係は取り扱うが、これらの中で通信可能であればサービスが提供可能であるとし、各ネットワーク機器間に関しては本方式では対象の範囲外としている。また、収集するシステムの情報は静的な構成情報に限り、システムの稼動中に変化する依存関係については取り扱っていない。なお、本稿では特に明記しない限り、ホスト名は各ノードに固有につけられた名称であり、インターフェイスに依存する名称とは異なる意味を持つ。

本提案方式は、システム構成情報の収集部、依存関係情報の抽出部、依存関係情報の出力部と大きく分けて3つの部門で構成されている。その構成図を図1に示す。

- システム構成情報の収集部

依存関係情報の抽出部から構成情報の収集に関するリクエストを受け取り、各計算機・周辺機器においてリクエストに対応する構成情報を収集、その結果を返す。収集に関しては、各

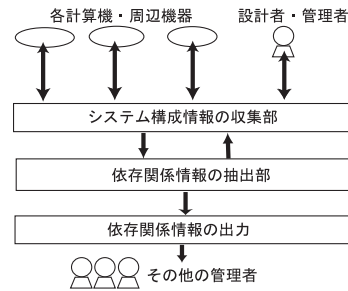


図 1: 構成図

計算機・周辺機器の持つ設定ファイル、ログを対象に行う。さらにこれらの情報から得ることのできない設計上の情報や、管理者のポリシーなどに関する事項は、システムを理解している管理者によって手動の入力を行うことにより、システム構成情報を収集する。

- 依存関係情報の抽出部

システム構成情報の収集部から、システム構成情報を受け取り、システム内での依存関係の抽出を行う。各種のシステム構成情報に対応したモジュールが用意されている。

各モジュールではシステム構成情報の内容を解析し、あらかじめ設定されたクラスの分類に従いオブジェクトを生成した後、データベースに格納する。その際、オブジェクトの属性等をオブジェクト情報とし、オブジェクト間の依存関係を示す情報を依存関係情報として分割して格納する。また格納したオブジェクトに他の構成情報へ依存関係がある場合、システム構成情報の収集部へリクエストを送る。

- 依存関係情報の出力

データベースに格納されたオブジェクト情報及び依存関係情報から、管理者が必要な情報のみを取り出し表示する。その際、あらかじめ設定されたクラスの分類や各情報間関係を考慮して情報の整理を行う。

3.2 クラスの分類

本提案方式では、システムを構成する情報を依存関係情報の抽出部において解析し、全ての情報をクラス分けする。これは従来の管理者がシステム情報を確認した際、大きくどのような分類に所属する内容であるかを判断している作業に相当する。

クラスは複数の属性をもち、基本的に共通の属性を持つものを一つのクラスとして分類する。さらに各クラスが複数の属性をもつ場合、オブジェクトを一意に決定できる最低条件の属性を主属性とする。主属性は1つでも複数でもかまわないものとする。なおクラス分類は依存関係の抽出時のチェックの効率化をはかるためだけでなく、今後、属性を増やすだけで様々な情報を取り扱えるといった拡張性を考慮して定義を行う。分類の例を表1に示す。

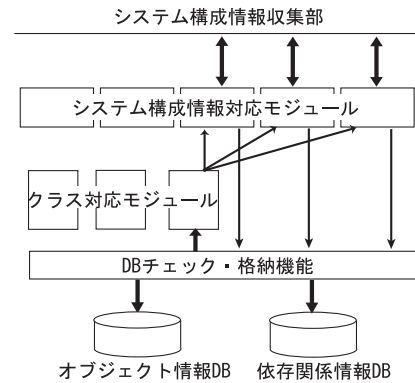


図 2: 構成図

3.3 依存関係の分類

生成されたオブジェクト間の依存関係に基づき、分類を行う。依存関係を示す方向によってベクトルを持つため、そのベクトルの向きによって単方向と双方向の二種類の依存関係が存在する。ディスクやファイルシステムのマウント、アプリケーションが特定のサービスを使用するなどが前者であり、ディスクのミラーリングなどが後者に相当する。

なお、本提案方式では、複数のオブジェクト間の関係において、全ての属性が一致したオブジェクトを同一オブジェクトとし、主属性のみが一致し他の属性が一致しなかったオブジェクトを準同一オブジェクトとし、双方向の関係に分類する。

3.4 依存関係情報の抽出機構

この機構は依存関係情報の抽出部を持つ機構であり、システム構成情報からオブジェクトを生成し、システム内での依存関係を抽出してデータベースに格納する。データベースに格納された情報に関して、依存関係のある情報が不足していた場合、リクエストをシステム構成情報の収集部に送信する機能をもっている。

依存関係情報の抽出機構は、データベースのチェック・格納機能、複数のシステム構成情報対応モジュール・クラス対応モジュールとオブジェクト情報と依存関係情報を格納する2つのデータベースから成り立っている。

本機構の概要を図2に示す。

以下に、依存関係情報の抽出機構の工程を示す。

1. 初期条件として事前に Host, Service 等の情報をオブジェクト情報データベースに未調査フラグを立て格納する。
2. データベースに未調査フラグがついたオブジェクト

がある場合は、以下の工程を繰り返し、無い場合は終了する。

3. 依存関係のあるシステム構成情報を収集する必要があるとして、そのオブジェクトのクラスに対応したクラス対応モジュールが呼び出される。
4. クラス対応モジュールを呼び出したオブジェクトは未調査フラグが消える。
5. クラス対応モジュールから、オブジェクトの属性に対応した各種システム構成情報対応モジュールが呼び出される。
6. 対応モジュール内で設定されたシステム構成情報の収集に関するリクエストを選択する。
7. リクエストがすでに収集済みかを確認後、未収集の場合はシステム構成情報の収集部にリクエストを送信し、収集済みの場合、2に戻る。
8. システム構成情報の収集部から結果が返信されると、対応したシステム構成情報対応モジュールで受け取り、構成情報の内容を解析する。
9. 解析結果を元に、システム構成情報対応モジュール内で設定されたクラスからオブジェクトを生成する。またオブジェクト間の依存関係の分類を行い、分類結果とその関係の詳細な内容からなる依存関係情報を同時に生成する。
10. 生成された全てのオブジェクトは、固有の ID と未調査フラグを付けオブジェクト情報データベースに格納する。
11. 依存関係情報は ID によってオブジェクト情報と連結し、依存関係情報データベースに格納する。
12. 同一オブジェクトが既に格納されていた場合、オブジェクトの格納は行わず同一 ID に変更する。また準同一オブジェクトの場合、異なる ID を付け準同一の依存関係情報を追加する。
13. 同一オブジェクト、及び準同一オブジェクトが存在しない場合、エラーの可能性を示すダウトフラグをたてる。フラグはモジュールを呼び出したオブジェクトと、その結果から生成されたオブジェクトのうち、呼び出したオブジェクトと同クラスのオブジェクトにつけられる。

表 1: クラス分類の例

クラス名	該当情報	主属性	属性
Host	ホスト情報	ホスト名	OS 名, Ver
Directory	記憶装置上のデータ	ホスト名, パス	
Disk	物理的記憶装置	ホスト名, (仮想) ディスク名	FS, サイズ
Service	提供するサービス	ホスト名, サービス名, ポート番号	ソフト名, Ver
NIC	NIC に関する情報	ホスト名, IP	NIC 名, MAC

14. 2 に戻る。

なお、システム構成情報の解析・オブジェクト生成・依存関係の分類・リクエストの選択等は全て対象のベンダーやバージョンによって固有のルールが存在し、それぞれ異なっている。本提案方式では多様なベンダー、バージョン毎に各種モジュールを用意することでこれを取り扱う。

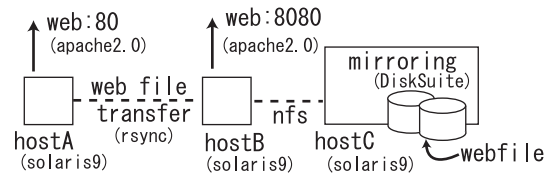


図 3: システム構成の例

3.5 依存関係情報の出力

管理者に管理対象が関連している計算機・周辺機器の構成情報を的確に伝える。各オブジェクト情報をノードとし、オブジェクト間の依存関係情報をリンクとして表される。出力の際のキーとなったオブジェクト情報から依存関係で分類された関係の方向に沿って順次追いかけることによって、キーとなったオブジェクトに依存関係をもつオブジェクト情報を収集する。その際、単方向の依存関係は順列、もしくは逆順のどちらか一方のみで依存関係の探索を行い、双方向はどちらの方向にでも依存関係を追いかける。

出力される結果は任意に指定したクラスもしくは各オブジェクト情報から、オブジェクトの主属性やクラス単位など限定した依存関係の出力が可能である。またダウトフラグや、収集時のエラーフラグを表示することでシステム構成上の間違いが発生している可能性を指摘する。

4 提案手法の動作例

本章では図 3 の構成をもつシステムにおいて提案手法を使用した動作例を示す。

図 4 の出力例 1 が hostA の web サービスが依存関係を持つシステム内のディスク (下部 4 行) と、そこまでの関係のつながり全て (上部) の表示結果であり、図 5 の出力例 2 は hostC のディスクに依存するサービスとその属性の表示結果である。

```

web:80 (hostA) << /usr/lab/public_html :documentroot
/usr/lab/public_html<< /dev/dsk/c0t0d0s6 :mount
/usr/lab/public_html<< /export/public_html (hostB) :filetransfer

/export/public_html << /usr/web/public_html (hostC) :nfs

/usr/web/public_html<< /usr :mount
/usr << /dev/md/dsk/d4 :mount
/dev/md/dsk/d4 << /dev/dsk/c0t0d0s4 :metadevice
/dev/dsk/c0t0d0s4 == /dev/dsk/c0t1d0s4 :mirroring

Host (hostA) Service (web:80) >> Host (all) Disk (all)
hostA: /dev/dsk/c0t0d0s6
hostC: /dev/dsk/c0t0d0s4
hostC: /dev/dsk/c0t1d0s4
    
```

図 4: 出力例 1

```

Host (hostC) Disk (/dev/dsk/c0t0d0s4) << Host (all) Service (all)

hostA: web:80: apache2.0
hostA: filetransfer: rsync
hostB: web:8080: apache2.0
hostB: filetransfer: rsync
hostC: nfs
    
```

図 5: 出力例 2

出力例では web の管理者は自分の管理するサービスがシステム内のどのディスクにまで関連して稼働しているのかを把握することができ、また逆にストレージシステム管理者は具体的にどのサービスがディスクを使用しているかの判断を行うことができる。このように各管理者がシステム全体の依存関係を必要なレベルで出力することが可能である。

5 議論

提案手法は、システム内の依存関係を収集し、サービスによる依存関係を利用して収集された情報を関連付け、システム全体の構成把握に必要な情報の提供が可能である。これは、特に複数のサーバが乱立した複雑な関係をもつシステムや、複数人の

管理者によって分担して管理される大規模なシステムにおいてレベルの異なる管理者のシステムに対する理解度の支援を行う上で有効である。

これにより、従来手法ではシステムへの理解度の足りない管理者への負担を軽減し、よりレベルの高い運用管理へとコストを振り分けることが可能となる。このような特徴により、本提案手法は特に大規模・複雑なシステムの運用管理性を従来よりも向上させている。

しかしながら、多様な構成をもつシステムの対応に関し幾つかの問題点を含んでいる。

本手法は多様化するシステム構成に対応するため、各ベンダーやバージョンごとにその構成情報を読み取るモジュールを用意している。そのため対応モジュールを事前に複数用意する必要があり、特異なシステムに対してはモジュールを独自に作成する必要がある。このような煩雑さを伴う各種モジュールは、従来での高いスキルをもつ上級の管理者の持つ豊富な知識に相当している。また、一般的なシステムでのモジュールはほぼ共通のものであり、特異なシステムにおいてはその特性を理解している上級の管理者によりモジュールの作成が可能である。

さらに、システムを構成する多数の情報を、サービスによる依存関係で関連付けて出力するだけであり、構成情報が大量になった場合、これらを見やすく出力できるとは言いがたい。このことは大規模・複雑なシステムでの管理者の理解度を向上させる効率を低下させる原因となる可能性がある。本手法では、管理者が必要な関係のみを限定して取り出すことで、大量の構成情報を管理者の要望に合わせて提示している。今後は、オブジェクトを依存関係情報のみで繋げるだけでなく、エラー内容や、クラス、ホストごとでグループ化し、視覚的に訴える仕組みを取り入れる必要がある。

また初期条件として、対象システムの既知の情報の格納を行っている。この初期条件がほとんど無い場合や、ある特定のノード・クラスに関してのみなど著しい隔たりがある場合、システム構成情報の取りこぼしが発生する可能性がある。しかしながら、本手法は、システムを熟知した管理者によって入力され、その結果を他の管理者に提供するという仕組みをとっている。そのため初期条件での、入力不足や隔たりは問題にはならない。

本手法の拡張性としては、システムの保守や改良にも、収集する情報の幅を広げる事によって対応す

るなどの応用も可能である。静的なシステム構成情報だけでなく、動的に変化する状態情報、ログから抽出される情報 [5] や障害情報等を、依存関係情報と連動させることにより、障害監視を支援を行うことができる。これにより、どのサービスが安全に稼動しているのか、また致命的なダメージを受けているのか、さらにはサービスは稼動しているが障害の影響を受けている状態であるなど、障害がサービスに与える影響をシステム内の構成管理情報を使用して明示的に示すことも可能となる。さらに従来のシステム管理ツール等で提供されている CPU の稼働率、ディスクの使用率などの利用頻度を示す情報などにシステム構成の管理情報、障害履歴加えることにより、システム構成上の重要度等を分析し、今後のシステムの拡張や変更など改良の指針とするといった利用法なども考えられる。

6 おわりに

本稿ではサービスによって依存関係をもった計算機本体・周辺機器の関係を利用してシステムの構成管理の支援法の提案を行った。管理者は管理対象から依存関係をもつサービス・計算機・周辺機器を容易に知ることが可能であり、システムの構成管理性を大きく向上させることができるであろう。今後は、構成情報の記載ミスへの対応、動的なシステム情報、障害監視情報、などを取り入れることで、システムの改良から保守作業などの支援を行う。

参考文献

- [1] 敷田幹文, 井口寧, 三輪信介, 丹康雄, 松澤照男. 大規模高可用性サーバの設計と運用. 情報処理学会 分散システム/インターネット運用技術シンポジウム, pp. 57-62, 2001.
- [2] 日立製作所. JP1 Version6 JP1/Base マニュアル. 2001.
- [3] 長田智和, 谷口祐治, 玉城史郎. 大規模分散ネットワーク運用管理システムの提案. 情報処理学会 DSM 研究会, Vol. 20, No. 6, pp. 31-36, 2000.
- [4] 泉裕, 上原哲太郎, 國枝義敏. 柔軟な管理情報の共有によるトラブルチケットシステムの構築. 情報処理学会 DSM 研究会, Vol. 20, No. 10, pp. 55-60, 2000.
- [5] 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.