

# 大規模ネットワークでのフレキシブルな階層構造による サーバ監視システムの提案

笠原 浩介<sup>†</sup>, 敷田 幹文<sup>‡</sup>

## 概 要

近年, 大規模かつ複雑化しているネットワークでは, 様々なサービスを提供するサーバが多数存在しており, それらを管理するコストも増大する傾向にある. また, 構成の変更等, 状況の変化に柔軟に対応できるサーバ監視システムが求められている. このような状況より, 単一の組織内で多数のサーバを管理・運営する場合では, 管理コストの削減や知識の集約, サービスの向上などを目的として管理業務を一括して行う傾向にある. そこで本研究では, 管理者が膨大なシステム全体の構成を把握, 維持しておかなければならないという負担を軽減させ, 構成変更や障害発生時に迅速に対応させるため, 周囲の機器の性能やネットワークの状態から, より良い階層構造を自動的に構築するサーバ監視システムを提案する.

## The Proposal of Monitoring System for Servers by the Flexible Layered Structure in a Large Scale Network

Kousuke KASAHARA<sup>†</sup>, Mikifumi SHIKIDA<sup>‡</sup>

## Abstract

The network systems become large scale and complicated in recent years, and many servers which provide various services exist. The cost of monitoring those network systems also tends to rise. And, monitoring system which can respond to change of a situation flexible, for example the change of network structure, is demanded. In this paper, we propose monitoring system for servers which builds a better layered structure automatically using the performance of surrounding apparatus and the state of a network. It makes the burden that an administrator has to grasp and maintain the composition of a vast quantity of whole system reduced.

## 1 はじめに

近年, インターネットの普及および急速な発展に伴い, 企業や学校などのネットワークの大規模・複雑化が顕著に見られる. メールや WWW などといったサービスが日常生活の中でなくてはならないものになってきており [1], それらのサービスを提供するサーバに対して, 非常に高い信頼性が求められるようになってきた [2]. このようなサーバは年々増加傾向にあり, 企業などにおいては, 多数のサーバの

可用性を維持するために障害の早期発見, 迅速な復旧作業を効率的に行う必要がある.

このような問題を解決するために, ネットワーク管理の処理の一部を記述した管理プログラムを被管理システムに投入して実行する方式 [3] や, 一つ以上の被管理システムを巡回して管理処理を行うモバイルエージェントを用いたネットワーク管理 [4] やサーバ管理 [5] といった大規模ネットワーク管理の提案がこれまでにある.

しかしこれらの研究はネットワーク管理を主眼に置いたものが多く, サーバ監視を考慮したものでも, 大規模・複雑化しているシステム全体の構成, つまりどの場所にどのような性能の機器があるのか, どれぐらいの数の機器があるのかといったことを管理

<sup>†</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>‡</sup> 北陸先端科学技術大学院大学 情報科学センター  
Center for Information Science, Japan Advanced Institute of Science and Technology

者が把握し、その情報を元に効率のよい監視が行えるよう振り分けを行う必要がある。

本研究では、このような問題を解決するため、管理者が逐一指示を出さなくてもより良い階層構造を自動的に構築し、構成変更や障害発生時に柔軟に対応できるサーバ監視方式を提案する。

本稿では、2章で既存の SNMP を用いた監視方式とモバイルエージェントを用いた監視方式の利点および問題点について述べ、3章では提案する方式のシステム構成や動作等について述べ、4章で提案方式についての考察を行う。

## 2 既存のサーバ監視方式

従来の SNMP を用いた監視方式とモバイルエージェントを用いた監視方式について、それぞれの特徴と問題点を述べる。

### 2.1 SNMP を用いた方式

従来の SNMP を用いた集中型監視は、マネージャが複数のエージェントに対して定期的に 1 対 1 のポーリングを行って情報を収集し、またエージェントが異常を検知した場合にマネージャに通知するトラップ機能によって障害を把握する。

大規模なネットワークで運用を行う場合は、マネージャへの通信量および処理負荷が増大するので、解決法として各拠点にマネージャを置き、さらにそれらを統括管理するマネージャを置く必要がある。よって大規模なシステムになると構成変更が困難となる [4]。またサーバ監視においては、そのサービスの状況に関する様々なメッセージが重要であるので、MIB での監視は不向きであるし、トラップ機能を用いた場合についても各監視対象に専用のプログラムが必要となり、管理コストが高くなる。

### 2.2 モバイルエージェントを用いた方式

既存の研究 [4-6] ではマネージャから各監視対象へモバイルエージェントを送り、巡回させ情報を収集、解析し必要な結果だけをマネージャに持ち帰ることで、大規模なネットワークに適した運用を行っている。

大規模なネットワークへ適用する場合、巡回するエリアをグループ分けした複数のモバイルエージェントを用いたり、SNMP と同様に複数のマネージャ

を用いる。この際、巡回するエリアの分け方や、すべてを統括しているマネージャの下についている各拠点のマネージャをどの機器にするのか、どういった順番でモバイルエージェントを巡回させていくのかという設定が、システム全体のパフォーマンスに大きく影響してしまう。

よってシステムの構成を一通り把握しておかなければ、効率が悪くなる恐れがある。

## 3 状況変更柔軟に対処可能な階層構造自律組み替え型サーバ監視方式

提案システムの通常時の動作、障害発生時に自律的に組み替わる動作および管理者の指示による組み替えの動作を述べ、マネージャやエージェントの個別の動作を述べた後、階層構造を構築するための指標について述べる。

### 3.1 通常監視時の動作

本提案方式では、複数の監視対象サーバをツリー型に配置し、それらを根のマネージャによって監視作業を行う。

監視対象のサーバ上にはエージェントが常駐している。図 1 のような構成になっているとする。エージェントは子エージェントからの報告を一定時間ごとに受け取る。また、エージェントは監視項目についての監視を行い、子から受け取った報告とともに親へ報告を行う。このように下位の情報は順に上位に渡されてゆき、最終的にマネージャまで報告が行われる。

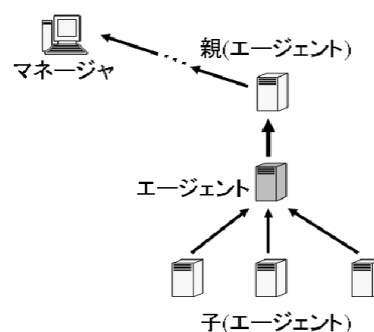


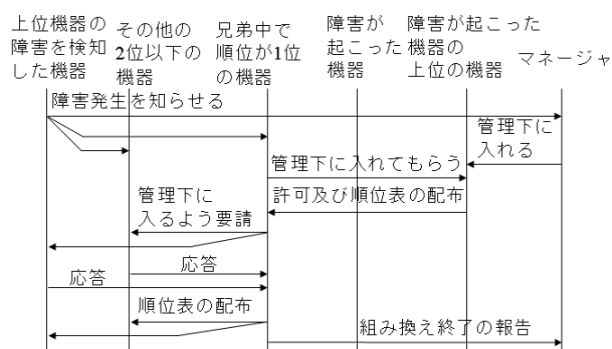
図 1: 基本動作例

監視項目はサービスの稼働状況である。サービスの稼働状況とは、サービスのプロセスの存在や要求に対する応答が帰ってくるか、エラーログの詳細、通

### 3.2 ツリーの組み換え

### 3.2.1 自律的なツリーの組み替え

その手順を図 2 に示す.



### 3.2.2 マネージャの指示によるツリーの組み替え

| マネージャ  | 組み換え対象の<br>機器                          | 行き先の<br>上位機器     | 行き先の<br>下位機器 |
|--------|----------------------------------------|------------------|--------------|
| 組み換え指示 | 管理下へ入れてもらう<br>許可及び順位表の配布<br>管理下に入るよう要請 |                  |              |
|        |                                        | 管理下から<br>はずしてもらう |              |
|        |                                        | 許可               |              |
|        |                                        | 管理下に入る           |              |
| 終了報告   | 許可及び順位表の配布                             |                  |              |

### 3.3 システムの構成要素の動作

### 3.3.1 マネージャの動作

図4にマネージャの状態遷移図を示す.

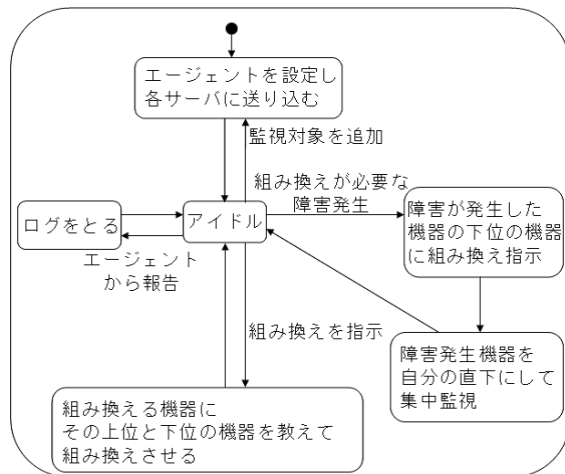


図 4: マネージャの状態遷移図

### 3.3.2 エージェントの動作

エージェントは、一定時間ごとにそのサーバの稼働状況を監視し、異常を検知する。異常を検知した場合は、速やかにマネージャに報告を行う。

自分自身の障害を監視する場合の状態遷移図を図 5 に示す。

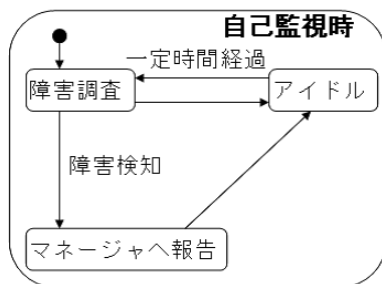


図 5: 自己監視時のエージェントの状態遷移図

エージェントは子からの報告を一定時間ごとに受け取り保存しておき、自分が親への報告を行う際に、子から受け取った報告もあわせて報告する。

この際、すべての状況が正常であれば、すべて正常であるという報告のみを行い、異常な項目があるのであれば、その内容をメッセージに加工して報告を行う。

子からの報告は一定時間ごとに行われるのであるが、報告が一定時間を経過しても行われない場合には、何らかの障害が発生したものと子に対して調査を行う。また、親への報告も一定時間ごとに行うのであるが、この際にも通信が行えないなどの深刻な障害を検知することができる。このように、エージェントは自己を監視するだけでなく自己の周囲、つまり親や子の障害の有無についても監視を行

うことができる。

これらの検知された障害情報は速やかにマネージャに報告され、障害が発生した機器が親であった場合には、3.2.1 節の手順に従い自律的にツリー構成を組み替える。

以上の、自己の周囲についての監視を行う場合の状態遷移図を図 6 に示す。

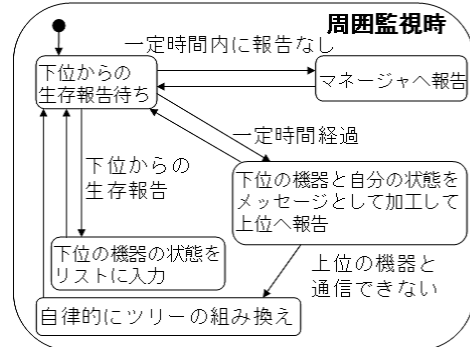


図 6: 周囲監視時のエージェントの状態遷移図

障害発生時における自律的な組み替えでは、3.2.1 節で述べたように順位表が必要であるので、親と子は同じ内容の順位表を共有する。また、親に障害が発生した場合には、親のさらに親、つまり 2 つ上の階層の機器に子から選出された代替機を繋ぐ必要がある。よって、すべての子は親だけでなく 2 階層上位の機器も知っておく必要がある。加えて、2 階層上位の機器も確実に繋がるという保証はないので、さらに上位の機器についても知っておく必要性が出てくるので、親から最上位のマネージャまでの直系の機器のリストが必要となる。

以上より、各エージェントは自分の子のリスト、順位付きの兄弟のリスト、マネージャから自分までの直系の機器のリストの 3 種類のリストを持つ必要がある。

### 3.4 ツリーの評価指標

ツリー構造を構築する際に、機器を最適な位置に配置するためには、ツリー上の機器配置に対する評価の指標が必要である。

しかし、機器を配置する際に指標となる要素は、機器の性能やネットワークの伝送速度など複数存在し、管理者のポリシーによって重要視される要素が変わる。

そこで、管理者のポリシーを考慮した、ツリー上の機器配置に対する評価の指標を示す。

### 3.4.1 評価指標に用いる要素

指標となる要素は、以下の3つを用いる。

- ネットワークの伝送速度：子を何台まで繋げても通信に支障がないかを判断する。
- 通信の遅延：繋がっている機器同士がネットワーク的に近いツリー構造がより良いので、遅延は少ないほうが良い。
- 機器の性能：処理能力やリソース量を測定して、何台の子を繋いでもよいかを判断する。

### 3.4.2 各機器に接続可能な子の限度数 $L$

一般に、機器の管理操作に用いる帯域はネットワークの最小帯域の5%以下であることが望ましいといわれている [7]。

これに従い、実効帯域  $B[\text{bps}]$ 、1つのエージェントがやり取りするデータ量  $S[\text{bit}]$  とすると、帯域により制限される接続可能な子の限度数  $L_{band}$  は式 (1) となる。

$$L_{band} = \frac{B}{20S} \quad (1)$$

機器の性能については簡単なベンチマークを行い、子からの報告が何台分までなら処理できるかを計測し、処理能力やリソース量により制限される接続可能な子の限度数  $L_{per}$  を算出する。

各機器に繋がる子の数の上限は、前述した接続可能な子の限度数  $L_{band}$ 、 $L_{per}$  以下とする。ただし、この限度数まで繋げるのか、もしくはある程度の余裕を持たせるのかといった判断は管理者が設定すべきであるので、管理者が設定する接続限度率  $P(0 \leq P \leq 1)$  を設けて、接続可能な子の限度数  $L$  は式 (2) となる。

$$L = \begin{cases} L_{band} \times P & (L_{band} < L_{per}) \\ L_{per} \times P & (L_{band} \geq L_{per}) \end{cases} \quad (2)$$

限度数  $L$  を超える数の子が繋がろうとした場合には、超過分の子が、限度数を超えていない他の機器に繋がることになる。

また、各エージェントが持つ兄弟の順位表を求めるときに、この限度数  $L$  を用いて子の順位付けを行うことにより、障害時における自律的な組み替えが行える。

### 3.4.3 接続の優先度

ツリー構造全体を見て各機器同士の接続がネットワーク的に近い方が、より良いツリーであると考ええる。

通常、ネットワーク的に近いということ「経由するルータやネットワークの数が少ない」「RTTが短い」の2つの意味がある。

ここでは通信の遅延を評価の尺度に用い、RTTの値が少ないものをより評価の高い接続先とする。

接続先との通信の遅延  $D[\text{ms}]$  とすると、 $D$  の値が少ないものほどより評価の高い接続先ということになるのだが、計測された遅延の値が同じである機器が出てくるものと考えられる。また、遅延の値は動的に変化するものなので、常に最も遅延が少ない機器に繋がるように設定していると、絶えず接続先を変えなければならないという可能性も考えられる。

このような絶え間のない移動を抑制するためには、常に遅延の値が最小のものを選ぶのではなく、遅延の値がある程度の範囲内に収まっているものを選ぶべき。

繋がるべき親を探している機器はまず、ネットワーク的に近い機器がマネージャから複数台挙げられる。これはあらかじめ管理者が、ネットワーク的に近い機器のうち上位  $N$  台までと設定しておき、マネージャがネットワーク図と照らし合わせて、近くだと思われる上位  $N$  台の機器についてを通知する。次に、その  $N$  台の中から1台を選択する。

ここで、 $N$  台の中から1台を選択する方法は管理者のポリシーの違いにより複数考えられる。本研究では2つの方法について考える。1つは、接続可能な子の限度数  $L$  の値が大きなもの、つまり機器の性能およびネットワークの性能が高いものに優先的に子が繋がるという方法。もう1つは、接続可能な子の限度数  $L$  に対する現在繋がっている子の数  $C$  の割合  $R$  を算出し、その割合  $R$  が少ないもの、つまり機器の性能およびネットワークの性能に対して現在接続されている子の処理負荷の割合が比較的少ないものに優先的に繋がるという方法である。 $R$  の算出式を式 (3) に示す。

$$R = \frac{C}{L} \quad (3)$$

よって接続先の優先度は、まず接続先との通信の遅延  $D$  を計測し、ネットワーク的に近い機器のうち上位  $N$  台を挙げ、管理者のポリシーにより、限度数  $L$  の値が大きなものか、 $R$  の値が小さいものかのどちらかを選ぶことで決定される。

この評価指標を用いることにより、監視開始時にまだツリー構造を構築していないところから組み上げてゆくときはもちろん、ある機器をツリーに挿入する時に、よりよいツリーとなる順に挿入地点の候補を挙げることができる。

## 4 考察

ネットワーク管理においてよく用いられる SNMP は、監視項目が少ないものや単純なものであれば非常に有用であるが、サーバ監視を対象とする場合には多数の MIB を取得しなければならなかったり、MIB では対応しきれないといったことが考えられる。この点については、提案方式や従来のモバイルエージェントによる方式などの、メッセージを用い、それを加工してマネージャへ通知するものであれば、通信量の削減やすべての監視項目についての対応を期待できる。

従来の SNMP やモバイルエージェントを用いる方式を大規模ネットワークで適用する場合、マネージャが複数存在し、それらを統括するマネージャを置くような構成となり、統括しているマネージャの下に置かれるマネージャをどの機器にすればよいのか、複数あるマネージャのうち、どのマネージャにどの機器を繋げればシステム全体の効率が良くなるのかといったことを、ネットワークの状態や各機器の性能などを管理者がすべて把握した上で決定しなければ、効率の悪いシステムになる恐れがある。また、単一のマネージャによるモバイルエージェント方式の場合も同様の問題を孕んでおり、巡回すべき機器をいくつかのグループに、どういう分け方をするのか、一つのグループ内でどのような順番で巡回するのかを、ネットワークの状態や各機器の性能を管理者が把握したうえで効率を考えて設定する必要がある。これに対して提案方式では、3.4 節に挙げたツリーの評価指標を用いることにより、管理者の介入なしにサーバ群をツリー構造に組み上げたり、ツリーの途中に新規にサーバを挿入する際に、より良い挿入先の候補を自動的に順位付けして示せるので、管理者がネットワークの状態や各機器の性能をすべて把握し設定を行わなくとも、それらの状況に柔軟に対応できる。

## 5 おわりに

従来方式では、管理者がシステム全体の構成を詳細まで把握し、維持する必要があった。さまざまな場所に分散されている機器の各々の性能の把握や、どのマシンをどのマシンに監視させるかという効率を考慮した振り分けを管理者がよく吟味して設定する必要があった。

本稿では、機器の性能やネットワークの状態などから、より良いツリー構造を自動的に構築することで、これらの問題点を解決するサーバ監視方式を提案した。本提案方式は特に、大規模ネットワーク上にある、管理者が詳細まで把握し切れていないほどの多数のサーバを集中監視する場合に有効であると考えられる。

本方式では、ツリー構造の部分木に着目して機器をより良いツリーになるよう配置したのだが、ツリー全体を見たときの局所的な偏りや、ツリー全体に対する評価などは今後の課題である。

## 参考文献

- [1] 磯川弘美, 萱嶋信, 寺田真敏, 伊藤武. インターネットサービスの特徴を考慮したサーバ稼働監視システムの提案. 情報処理学会 研究報告 DSM, No. 009, pp. 1-6, 1998.
- [2] 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.
- [3] 吉原貴仁, 磯村学, 堀内浩規. 分散ネットワーク管理のための動的負荷分散方式. 電子情報通信学会論文誌 B, Vol. J86-B, No. 3, pp. 438-448, Mar. 2003.
- [4] 知念真也, 長田智和, 谷口祐治, 玉城史朗. 分散オブジェクト技術を用いたネットワーク監視システムの設計. 情報処理学会 研究報告 DSM, No. 020, pp. 37-42, Dec. 2000.
- [5] 清水雅司, 敷田幹文. モバイルエージェント技術を用いたサーバ監視システムの設計. 情報処理学会 研究報告 DSM, No. 027, pp. 13-18, Oct. 2002.
- [6] 三好優, 釜洞健太郎, 朴容震, 浦野義頼, 富永英義. モバイルエージェントによる大規模・分散形ネットワーク管理法の一提案. 情報処理学会 研究報告 DPS, No. 097, pp. 61-66, 2000.
- [7] IETF RFC 1157. A Simple Network Management Protocol(SNMP). May 1990.