

# 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

清水 雅司<sup>†</sup> 敷田 幹文<sup>‡</sup>

北陸先端科学技術大学院大学  
<sup>†</sup> 情報科学研究科 <sup>‡</sup> 情報科学センター

## 概要

近年のネットワークシステムの大規模化・複雑化に伴い、それらを管理していくコストも増加する傾向にある。このような状況において従来用いられている SNMP を用いた一般的な監視方式でサーバを監視するとなると管理にかかる通信量が多い、管理性が悪くスケーラビリティを悪化させてしまうなどの問題を持っている。そこで、本研究ではモバイルエージェント技術に注目し、3種類のエージェントを協調作業させて集中管理されたサーバ群に対して柔軟に監視を行うことでそれらの問題を解決できる監視システムを提案する。

## The Design of Monitoring System for Servers with Mobile Agent Technology

Masashi SHIMIZU<sup>†</sup> and Mikifumi SHIKIDA<sup>‡</sup>

<sup>†</sup> School of Information Science, <sup>‡</sup> Center for Information Science  
Japan Advanced Institute of Science and Technology

## Abstract

The network systems become large scale and complicated in recent years, and as a result, the cost of monitoring those network systems also tends to rise. A general monitoring system in SNMP under such a situation includes some serious problems, especially for monitoring servers; for example, the increase of packets for monitoring, the decline of system scalability, and so on. In this paper, we propose the new monitoring system for servers using the mobile agent technology. This monitoring system consists of three agents and a manager, and their agents work cooperating with each other, which realize a efficient monitoring system for servers under the central management.

## 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よってネットワークシステムの大規模化が進んでいるが、それにつれてそのネットワークシステムの構成自体も複雑なものへととなり、今後もこの大規模・複雑化の傾向は続いていくと思われる。また、学校や企業でもこれらのネッ

トワークシステムを用いて教育・業務が日常的に行われており、ネットワークファイルシステムやメール、WWWなどの基幹ネットワークサービスの可用性を確保する必要が出てきた。そして、必然的にそれらのサービス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 [1]。この信頼性を保つために、ハード・ソフト両面で冗長性を実現するだけでなく、万一障害が発生した場合には、

その障害を有効的に検知するシステムの設計が急務である。

監視システムの設計にあたって、監視対象であるサーバ及びサーバが提供するサービスの大きな特徴を考慮すべきである。それらの特徴とは多様性と頻繁なシステム変更・更新である。よって監視システムは多種多様なものに対応し、かつ、頻繁に更新される可能性がある。今後も提供されるサービスは増えていく事が予測できそれに伴って監視対象や監視項目が増加し、従来の監視方式では単純に監視にかかる通信量が大きくなるだけでなく、管理性で深刻な問題が起こると考えられる。以上のことを踏まえて、円滑なネットワークシステム運用を考える上で、その監視システムの管理性は大変重要になってくる [2]。現在、管理性に最も優れたネットワークシステム構成の一つとして基幹サーバ群を高性能な大型ルータに広帯域なネットワークで局所的に集中させたものが注目されつつある。

そこで本稿ではこのような集中管理に適した局所的に配置されたサーバ群に対して監視を行うシステムを提案し、監視にかかる通信量や管理者側から見た管理性についての有効性を検討する。

続く 2 章では既存の 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、問題点について述べ、3 章で管理性の向上と通信量の低減などを目的とした提案するサーバ監視方式の概要について述べ、4 章でそれらのシステムの利点や問題点について議論を行う。

## 2 既存の監視方式

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をとりあげ、それぞれの利点や問題点を述べる。

### 2.1 SNMP を用いた監視方式について

集中監視型の SNMP 方式はマネージャ (監視ホスト) が複数のエージェント (監視対象機器) に対して定期的に 1 対 1 でポーリングを行うことによって、もしくは何らかの異常を検知してエージェント側からのトラップ機能を使うことによって、情報を取得し、その情報から障害を判断するというシンプルな構成で成り立っている。各フィールド

に対しての問い合わせに値を返すという簡素な仕組みであるため、他のアプリケーションからの利用が容易であり、また、それ単体の動作は非常に軽い。現在、監視ツールのデファクトスタンダードとなっている [3]。

しかし、ネットワークの複雑化・大規模化に伴い、以下のような問題が生じてきた。

- 各監視対象機器に対して往復の通信路が必要であり、マネージャにかかる通信の負担が非常に多い
- 取得する MIB のオブジェクト数は増加傾向にあり、SNMP での監視にかかる通信量も比例して増加する傾向にある
- 機器単体が正常であったとしても必ずしもユーザがサービスを受けることができるとは限らない (クライアント側からのサーバへの経路調査が必要である)
- 数多くの監視ネットワークができてしまい、それらを統合するための専用のマネージャを配置する必要があり、監視システムが階層化してシステムのスケラビリティを低下させてしまう
- 各ベンダ間で独自機能があり、ベンダ差をうまく埋めるのが難しい
- 複雑な監視を行う際は、専用のプログラムを各監視対象に設け、それからトラップ機能を用いて障害の検知をするといった複雑な仕組みになるために、そのプログラムの管理コストが高く、また、障害が起ったときのトレースが複雑な作業となってしまう

### 2.2 モバイルエージェントを用いた監視方式について

一般的な SNMP の他にモバイルエージェントを用いた監視方式があり、それらに関する既存研究 [4, 5] でいくつかの利点が明らかになっている。

従来の SNMP による監視方式では非常に多くの MIB オブジェクトを通信によってマネージャ側で受取り、その受け取った値からマネージャが障害を判断している。しかし、モバイルエージェントを用いた監視方式では基本的に通信はエージェン

ト自体の移動だけであり、さまざまな値はその移動先の監視対象上でエージェントが参照することができる。そして、その後障害判断をエージェントが行い、必要な結果だけをマネージャに報告するという事で管理パケットの低減を実現している(図1参照)。今後も取得するMIBオブジェクト数の増加が予想され、ますますこの利点は活かされると考えられる。

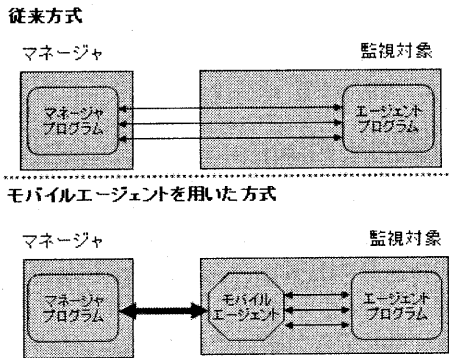


図1: エージェント移動による恩恵

そして、もし管理しているネットワークシステムに変更があったとしても、従来方式では複雑な監視を行っている場合、監視対象に専用のソフトウェアを別途インストール、チューニングしていなければならない、それらのソフトウェアを必要に応じてチューニングしなければならないのに対して、モバイルエージェントを用いた監視方式では移動するエージェントの障害調査プログラムの移動先や監視項目などの変更を行うだけでよく、監視対象サーバに対しては何ら手を加える必要がない。よって管理コストを抑えることができ、結果としてシステムのスケラビリティも向上させることができる。

以上、モバイルエージェントを用いることの利点を大きく2つ挙げてみた。しかしながら既存のモバイルエージェントを用いた監視方式ではいくつかの問題点もあり、以下にそれらを挙げる。

- エージェント自身に対する障害対策  
自律的に動き回るモバイルエージェント自身の位置を把握することが難しく、エージェント自身に対して何らかの障害が起ってしまった場合、そのエージェントの障害が監視システム全体に影響してしまう。

- エージェント移動による障害情報の遅延  
エージェントが障害情報を格納し、マネージャまで移動して報告するといった方法では、逐次変化していく障害状態を把握することが難しく、また障害情報の遅延にもつながってしまう。

そこで、本研究ではこれらの問題点に対して機能別にモバイルエージェントを複数用いてそれぞれを協調作業させることによって解決し、かつ新たな特徴として、刻々と変わる障害状況の把握とクライアント側からの調査を実現したサーバの監視方式を提案する。

### 3 提案方式

提案するサーバ監視方式の基本システムの概要及び各エージェントの働きを述べ、最後に信頼度を向上させるための追跡モードについて述べる。

#### 3.1 各エージェント説明

以下の3つのエージェント群とマネージャから本方式は構成される。また、それら各エージェントとマネージャの基本動作の概要図を図2に示した。

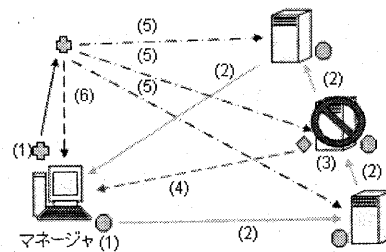


図2: 基本動作概要図

##### 3.1.1 定期巡回エージェント

###### 「最初の障害検知」

巡回すべき監視対象のサーバリストをマネージャから受取り、それを見ながら自律的にサーバ群を順番に巡回して行って調査を行う(図2(2))。サービスの存在や応答、CPU・メモリ・NICなどの負荷

を監視項目とし、異常を検知した場合はそのサーバ上に集中監視エージェントを生成して後は完全に復旧するまでその集中監視エージェントにそのサーバに関する全ての監視業務を任せる(図2(3))。集中監視エージェントが障害の起ったサーバを監視している間は定期巡回エージェントはそのサーバへは移動しない。また、MACアドレスまでも調べることで例えばHAシステムなど従来では障害検知が難しい複雑なシステムにおいても障害の検知を行うことができる。マネージャと各サーバ間の往復を避けて、巡回行動をとることでマネージャにかかる通信の負担を軽減することができる。以下に定期巡回エージェントの状態遷移図を示す(図3参照)。

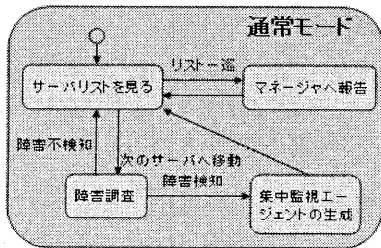


図3: 定期巡回エージェントの状態遷移図

### 3.1.2 集中監視エージェント

#### 「障害状況の把握」

自らは移動することなく、サーバ上で障害状況を監視し、改善・悪化などの変化があればマネージャへ通信を行って報告する(図2(4))。サーバ上に常駐するために定期的なポーリングでネットワーク上にトラフィックが発生せずに、通信量を抑えることができ、かつ、障害の変化を逐次マネージャへと伝えることができる。完全にサーバが障害から復旧すれば役目を終え消滅する。

### 3.1.3 経路監視エージェント

#### 「クライアント側から調査」

マネージャからルータリストと経路表を受け取り、ルータリストを見ながらクライアントとサーバ間でクライアント側に近いルータやゲートウェイ

に順次移動して、そこから各監視対象サーバへ問い合わせを行う(図2(5))。この時、受け取った経路表を元に、実際にそこからクライアントがサーバまで正常な経路で、かつ、指定時間内に通信を行うことができるかどうかを調べる。もし、異常を検知した場合は即時にマネージャへ通信を行って報告する(図2(6))。一方、正常な値はエージェントが保持してルータリストを一巡した後、マネージャへと戻り正常値を報告する。

### 3.1.4 マネージャ

#### 「障害状況の収集」

まず、定期巡回エージェントと経路監視エージェントを生成し、自分の持っているサーバリストとルータリスト・経路表をそれぞれに渡す(図2(1))。そして、それら2つのエージェントを生成した後は一か所に留まり、各エージェントからの報告によってログの収集を行い、障害情報を管理者へ知らせる。

ルータリストには巡回すべきルータ・ゲートウェイが記述されており、経路表はそのルータリストに記載されている各ルータ・ゲートウェイからの各監視対象サーバへの経路及びタイムアウトが記述されている。

サーバリストには巡回すべき監視対象サーバと、その各サーバの基本情報として、そのサーバのOS名、サービスしているプロセス名、サービスプロセスのバージョンが記述されている。一例を表1に記した。このリストを元に定期巡回エージェントはそのサーバに適した行動をとる。

表1: 各サーバの基本情報例

Host	OS	Process	Version
nameserver	Solaris	bind	9.2.1
www	FreeBSD	apache	1.3.26

基本的に、各監視対象に対して正常報告は定期的にエージェントが移動でマネージャに戻ってきた時のみ行われ、障害報告は障害を検知したら即時に各エージェントは通信によってマネージャに行く。よって、障害報告だけではできる限り早くマネージャは知ることができる。

### 3.2 追跡モード

モバイルエージェント自身に対する障害に対応するために通常時の通常モードの他に、追跡モードを取り入れ監視システム自体に対して信頼性を高める。この追跡モードでは監視にかかる通信量は多くなってしまいが、障害の発生箇所をつきとめることに全力を挙げる。

#### 3.2.1 モードの切り替え

定期巡回エージェント、経路監視エージェントは共にマネージャによって巡回するインターバルが決められている。もし一定時間内にこれらのエージェントがマネージャへと戻ってこなかった場合に、マネージャは追跡モードへと切り替わる。逆に、追跡モード時にそれら巡回している各エージェントが一定時間内に戻ってくれば追跡モードから通常モードへと切り替わる。このように、マネージャだけがモードの切り替えを行うことができる。

#### 3.2.2 追跡モード時の動作例

追跡モード時の定期巡回エージェント及び経路監視エージェントは各サーバ/ルータへ移動する度にマネージャへ自分の位置を報告し(図5(1))マネージャはそのデータベースを更新する(図5(2))。もし一定時間内にエージェントが戻って来なかった場合はマネージャはそのデータベースを用いて問い合わせを行う(図5(3))。反応が無かった場合、その位置のサーバ/ルータに集中監視エージェントを送信し(図5(4))、巡回リストからそのサーバ/ルータを除いてエージェントを再生成する。一定時間内にエージェントが戻ってきた場合は、マネージャは通常モードへと切り替わる。なお、追跡モード時の定期巡回エージェントの状態遷移図を図4に、通常モード時の基本動作と違う部分を図5に示した。

## 4 ディスカッション

前章で述べた提案方式と既存方式を様々な観点から比較することによって、本提案方式の利点や問題点を明らかにし有効性について議論していく。

監視にかかる通信量について各方式別に比較してみる。監視にかかる総通信量はモバイルエージェ

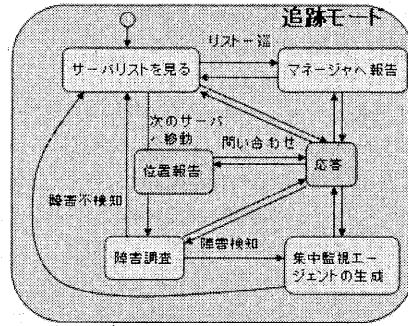


図 4: 追跡モード時定期巡回エージェント状態遷移図

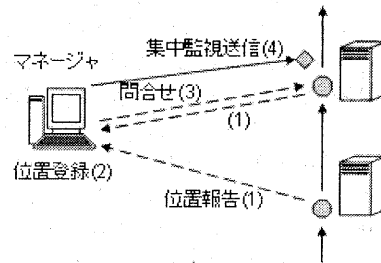


図 5: 追跡モード時の動作

ントを用いた方式ではエージェント自身の実装に依存してくるのに対して、SNMP方式では取得するMIB数に依存する。フィールドにもよるが、一つのMIBオブジェクトを取得するには約200バイトの通信量が必要である。これは非常に少ない値である。よって、もし取得するMIB数が少ないのであればエージェント方式よりもSNMP方式の方が管理パケットを低減させることができる。しかしながら、SNMPを用いた監視方式であるSun Management Center 3.0で実際に取得するMIBオブジェクト数を調査したところ、基本情報だけで約50ものオブジェクトを取得していることがわかった。複雑な監視を行うともなればこれよりも多数のMIBオブジェクトが必要とされる。そのため、多くの情報を必要とするサーバ監視には本方式による通信量の低減が期待できる。また、既存のエージェント方式に比べ本方式ではエージェントメッセージを多用するためにより通信量がかかると考えられる。これに関連してマネージャの負

担についても考察してみると、取得する MIB 数の他に監視対象数が影響を与えることになる。監視対象が多ければ大きいほど SNMP 方式ではマネージャの負担が大きくなってしまいが、モバイルエージェントを用いた方式ではマネージャからのエージェントの送信・受信の 2 回だけで済む。しかし既存のエージェント方式では後述のマネージャ同士を統合するための通信が必要となり、本方式よりもマネージャ負担は大きいと考えることができる。

スケーラビリティについての比較では、SNMP 方式と既存のエージェント方式では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設けてマネージャの情報を統合するために監視システムの階層化が起こり、それらの管理コストが非常に高い。一方、本提案方式では各監視対象グループを複数の定期巡回エージェントが担当し監視システムの階層化を避けることで単一のマネージャが監視情報を収集する。そのため、集中管理されているシステムにて本提案方式を運用すると管理性・スケーラビリティを向上させることができる。

そして、本提案方式の新しい特徴としてクライアント側からサーバへの調査が挙げられる。従来は主にネットワーク機器単体での監視だったものをクライアント側からのサーバ周辺の経路状態も監視することによって、クライアントが正規の経路を辿って指定時間内にサーバまでアクセスしているかを調査できる。

また、監視システムの設計に当たって信頼性については考慮しなければならない。既存のエージェント方式ではエージェントに対する障害に何ら保証が無いが本方式では追跡モードを設けてエージェント自身に対する障害を検知することで、信頼性を上げている。

以上をまとめると、本提案方式はサーバ監視など必要な情報量が多い場合に通信量の低減を実現し、またそれらの監視対象が集中管理されているシステム形態では、マネージャの単一化による管理性、及びスケーラビリティの向上が可能である。また、クライアント側からも調査することでサーバ周辺の経路状態も監視できる。

## 5 おわりに

近年のネットワークシステムの大規模化・複雑化に伴い、従来の SNMP を用いた監視方式では通

信量や管理コストの面などで様々な問題が生じてきた。そこで、本稿では分散オブジェクト技術であるモバイルエージェントに注目し、エージェントの協調作業によってこれらの問題点を解決する監視方式を提案した。本提案方式は特に、集中管理されているネットワークシステムでの運用が有効であると考えられる。また、サービスプロセスの自動アップデートやライセンス管理などへの応用も考えられる。

現在、提案してきた方式を Aglets[6, 7] を用いて実装中である。今後の課題として、そのプロトタイプの実装を完成させて、取得する情報やネットワーク構成によって通信量の変化を計り本提案方式の有効性について考察する。

## 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] 磯川弘実, 萱島信, 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実装. 情報処理学会 DSM 研究会, Vol. 9, No. 1, pp. 1-6, 1998.
- [3] Andrew S. Tanenbaum. コンピュータネットワーク. ピアソン・エデュケーション, 第 3 版, 1999.
- [4] 知念真也, 永田智和, 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 DSM 研究会, Vol. 20, No. 7, pp. 37-42, 2000.
- [5] 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・分散型ネットワーク管理法の一提案. 情報処理学会 DPS 研究会, Vol. 97, No. 11, pp. 61-66, 2000.
- [6] Danny B. Lange and Mitsuru Oshima. *Programming and Deploying JAVA Mobile Agents with Aglets*. Addison-wesley, 1998.
- [7] IBM 東京基礎研究所. Aglets project. <http://www.trl.ibm.com/aglets/>.