2K－1

ユーザレベル・アプリケーションレベルに着目した ネットワーク・システム管理方式の提案

近藤 祐志　勅使河原 可海
yuji@euclid.t.soka.ac.jp
創価大学 工学研究科

1. はじめに

近年、ネットワークの効率的で安全な運用を目指してネットワーク管理がますます重要になっている。

今まで行われてきたネットワーク管理は、SNMP を利用した構成管理・障害管理と RMON を利用したルータやスイッチの性能管理の併用が主であった。しかし、この方法ではネットワーク層までのプロトコルと IP アドレスまでしか管理・監視できず、誰がどこでどのアプリケーションを使っているのか、何のアプリケーションがパケットを送出したのか、どのくらいのリソースを使用しているのか、といったような詳細な情報を動的に得ることは不可能であった。アプリケーションがエラーのためにネットワークに大量のパケットを送出し続けるような事態が起きたときに、どこのホストで誰が何のアプリケーションを使っているのかを動的に把握できれば早急に対処することが可能である。そこで本論文では、どのユーザがどのようなアプリケーションを使って様々なリソースを消費しているかを管理マネージャが管理対象マシン上のエージェントから情報を収集・統計処理し、統合的な管理・監視をする新しいネットワーク・システム管理方式を提案する。

2. 提案

本論文で提案する管理方式は以下のような項目が管理・監視できることとする。

(1)ユーザ管理
・誰がどのくらいパケットを送出しているのか
・誰がどのくらいの時間、ホストに login しているか
・誰がどのアプリケーションをよく使うか
(2)アプリケーション管理

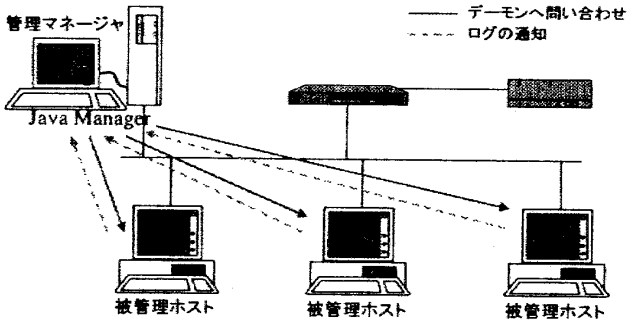

図 1:本提案による管理方式

・どのアプリケーションがどのくらいパケットを送出しているのか
・どのアプリケーションがよく使われているのか
(3)ホスト管理
・パケットの送受信状況
・CPU ロードアベレージ
・ハードディスクの使用率
また、以下のような特徴も有する。
・SNMP と RMON の併用方式では被管理ホストから応答がなくなったときに原因がホストにあるのかネットワークにあるのかが特定できなかったが、本論文で提案する方式ではどちらに原因があるのかも特定可能
・誰がどういうアプリケーションをよく使うかを記録することでなりすましを防ぐデータとすることが可能

3. 方式

(1)実現方法

2 で述べた詳細な管理・監視を実現するために、被管理対象ホストでのユーザやアプリケーションのふるまいをイベントログとして逐一記録するモニタデーモンと、モニタデーモンへの定期的な問い合わせやモニタデーモンが通知してきたイベントログの管理・編集・表示などを行う管理マネージャを用いる。モニタ

A Study of Network and System Management Methods Focusing on User Level and Application Level
Yuji Kondo, Yoshimi Teshigawara
yuji@euclid.t.soka.ac.jp
Graduate school of Engineering, Soka University

デーモンと管理マネージャ間のやりとりは SNMP で行う。

(2)SNMPのふるまい

イベントが発生した際にイベントログを取り逐次通知する情報と 定期的に問い合わせて取得する情報とに分かれる。逐次取得するべき情報はイベントが発生した際にモニタデーモンがイベントログとして記録し管理マネージャに通知する。定期的に取得するべきデータは短期間・長期間での統計を取る必要のあるデータで、管理マネージャがモニタデーモンへ定期的に送信要求を出し、モニタデーモンはこれを受けて情報を送信する。これらは独自拡張したMIBで定義され、SNMPはそれに基づいてふるまう。

(3)モニタデーモンのふるまい

・管理マネージャが集めている統計情報の定期的な問い合わせに対する報告
・被管理ホスト上に常駐し、ユーザの login やアプリケーションの起動などのイベント発生のたびにイベントログとしてホスト名、ユーザ名、イベント内容、イベント発生時間、などを逐一記録し、SNMP を用いて管理マネージャへ通知
・重大なエラーが発生した場合のホストと管理マネージャへのアラームの通知

(4)管理マネージャのふるまい

・独自拡張した MIB に基づいて SNMP で定期的にモニタデーモンに問い合わせを行い、統計情報を収集・処理
・モニタデーモンが送信してくるイベントログを処理

4. 実装

(1)Java

以上の方式を Java で実装し評価する。その理由としては以下のものが特に挙げられる。

・JMAPI(Java Management API) や WBEM (Web-Based Enterprise Management)の利用を検討中
・プラットフォームに依存しないコードが生成可能
・ネットワークプログラミングが容易
・オブジェクト指向であるため機能の追加が容易
・画像表示などのライブラリがあらかじめ利用可能

(2)実装

モニタデーモンの実装方法は、以下のようにいくつかの方法が考えられる。

・login、logout については logind を監視
・アプリケーションの実行などのユーザのふるまいについては、新しいプロセスの生成を監視する、shell がコマンドを解釈する際にイベントログを取る、など

login の監視やプロセスの生成など OS に大きく依存するため、プラットフォームに依存しないという Java の利点は失われるが、ローカルホストの管理情報にもブラウザでアクセスできるようにするためには Java で実装することが最も好ましいと思われる。

管理マネージャはモニタデーモンへの問い合わせとモニタデーモンからの通知の処理を行うだけなので、Java で実装すれば全くプラットフォームに依存しない形で実現できる。これにより、現在はスペック不足のために使用されていないホストも Java の実行環境があれば管理マネージャを動かすことが可能である。

(3)セキュリティ

外部からのアクセスは、管理マネージャ上の情報へも被管理ホスト上の情報へも厳しく制限されるべきである。具体的にはパスワードによる認証や同一ドメイン以外からのアクセス拒否などで実現する。

同一ドメイン内からの管理マネージャ上・被管理ホスト上の情報へのアクセスは、特にユーザ管理情報については自分の情報へのみ許可され、他のユーザの情報へは厳しく制限されるべきである。アプリケーション管理情報・ホスト管理情報については制限の必要はないと考えられる。

5. おわりに

本論文では、SNMP を利用した新しいネットワーク・システム管理方式の提案を行い、この提案を実装することにより、より詳細なレベルでの管理・監視が可能になる可能性を示した。

今後は、現在標準化が進められている JMAPI や WBEMなどの動向も念頭に置いて実装・評価したい。また、LDAP(Lightweight Directory Access Protocol)というディレクトリサービスを利用したユーザ管理の可能性も検討したい。

6. 参考文献

Mark A. Miller, "Managing Internetworks with SNMP", M&T Books, 1997.