

ネットワーク管理とシステム管理の統合

3R-4 ユーザレベル・アプリケーションレベルに着目した管理方式の実装と評価

近藤 祐志 勅使河原 可海
創価大学 工学研究科

1. はじめに

これまでに筆者らは、クライアント・サーバ等を含めた分散コンピューティング環境を管理するための方法として、ユーザレベル・アプリケーションレベル・ホストレベルという3つの管理レベルを定義し、SNMPと拡張MIBを用いて各レベルの管理情報を取得し、それらの管理情報に基づいて、詳細な管理を行うという方法(図1)を提案した[1]。さらに、各レベルの管理オブジェクトを拡張MIBに定義し、SNMP Agentを拡張・実装して、各レベルごとの管理情報の取得を行った[2]。

本研究では、さらに詳細な管理情報を取得するために、以前、実装した拡張SNMP Agentを改良した。また、ポーリングで管理情報を取得する管理マネージャを実装し、実際に管理情報を取得しグラフ化して、拡張したSNMP Agentの実装を確認した。

2. 背景

近年、インターネットへの接続が急激に増え、学校や企業におけるネットワーク環境の整備も急激に行われた。さらに、コンピュータの低価格化により、その勢いは加速され、今では、1人が1台のコンピュータを保持する環境も珍しくなく、これらの多くがネットワークに接続されている。

しかし、現在のネットワーク管理技術は、おもにルータやスイッチなどのネットワーク機器を対象としているため、クライアント/サーバや、ユーザ、ネットワーク・アプリケーションなどを含む分散コンピューティング環境を統合的に管理・把握することはできなかった。

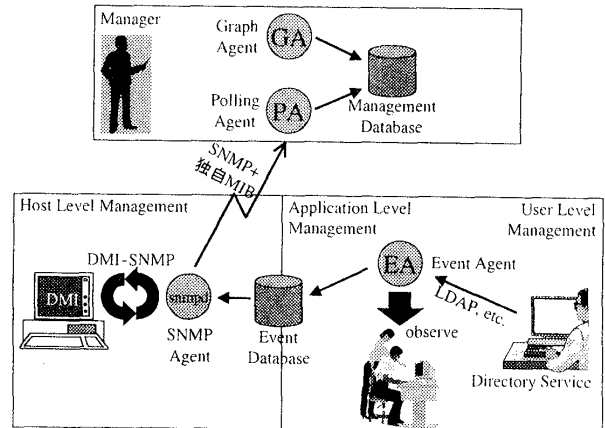


図1 本提案による管理方式

そこで、筆者らはユーザレベル・アプリケーションレベル・ホストレベルという管理レベルを定義し、各レベルの管理情報を取得して、分散コンピューティング環境の詳細な管理を行う方法を提案、実装してきた。

3. 実装方式

今回の実装では、管理マネージャ部には Advent Net SNMP Release 2 Package を、SNMP Agent の拡張には AgentBuilder 2.0 を利用した[3]。

(1)管理マネージャ

管理マネージャは、ホスト、オブジェクト ID、MIB、ポーリング間隔などの設定が記述されたファイルを読み込んで解析し、ポーリングを行う。取得した管理情報は host+OID という名前のファイルに記録される。

(2)拡張 SNMP Agent

今回の拡張では、テーブル型のデータを実装した。これにより、構造的なデータの取得が可能になった。

4. 実装結果

今回の実装では、以下のような管理情報を取得することができた。

An Integration of Network and Systems Management - An Implementation and Evaluation of Management Methods Focusing on the User Level and the Application Level -
Yuji Kondo, Yoshimi Teshigawara
yuji@euclid.t.soka.ac.jp
Graduate school of Engineering, Soka University

(1) ユーザレベル

- ・ユーザアカウント、過去の login 回数、最後に login した日付・時刻、logout した時刻
- ・現在ログインしているユーザ数、ユーザアカウント、ログインした日時、ログイン時間、使用プロセス名

(2) アプリケーションレベル

- ・総プロセス数、ユーザアカウント、CPU 使用時間、プロセス名

(3) ホストレベル

- ・CPU の負荷、使用状況
- ・メモリの状況、スワップの状況
- ・ディスクの使用量、使用率、マウントポイント
- ・現在時刻

今回、取得した管理情報の中から、CPU 負荷率とユーザのログイン頻度に関するログを Excel を用いてグラフにした(図 2、3)。

5. 評価

今回、取得することのできた情報は静的な情報と動的な情報の 2 種類に分類できるが、動的な情報については、管理者がシステムの状況を把握するためには必要不可欠なものを取得するように実装した。

各レベルの動的な管理情報と静的な管理情報をそれぞれ組み合わせて取得することにより、システムの状況をより詳細に把握できる。たとえば、CPU の負荷が異常に高くなったときに、誰がログインしていたのか、どのプロセスが負荷をかけているのかなどを検出することができる。このような仕組みは侵入検知などにも応用することが可能であると考えられる。

6. おわりに

本研究では、分散コンピューティング環境を管理するための管理方法の実装と評価を行った。

今後は、システム運用上、重要なリソース(パスワードファイルやネットワーク・アプリケーションなど)へのアクセスを、SNMP TRAP を利用して管理マネージャに通知する仕組みの実装を検討している。この仕組みにより、誰がいつどのリソースにアクセスしたのかを監視することができる。たとえば、どのアプリケーションがどのくらい利用されているのか、誰がいつバ

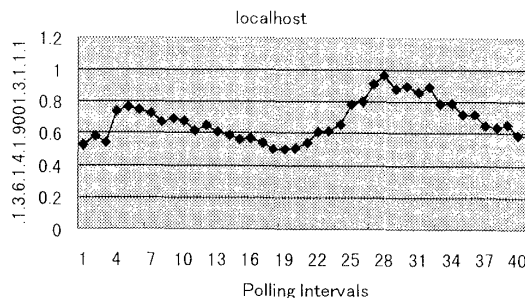


図 2 CPU 負荷率

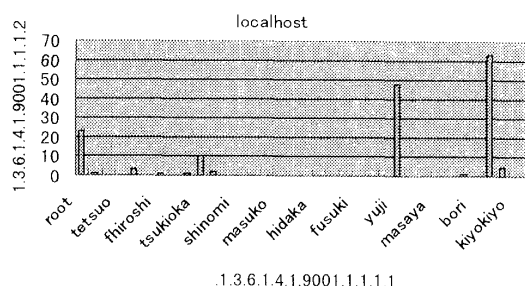


図 3 ログイン頻度

スワードファイルにアクセスしたのかなどを、管理情報として取得することができるため、障害時の原因究明やセキュリティの保持に有効であると考えている。

また、管理マネージャを使いやすくするためには CUI ベースから GUI ベースに移行する必要がある。グラフ化を行う部分を Applet で作成すれば、リモートからも監視を行うことが可能なので、非常に有効であると考えられる。

7. 参考文献

- [1] 近藤, 勅使河原, “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”, 第 56 回情報処理学会全国大会講演論文集, Vol.2K-1, pp.3-657-658 (1998)
- [2] 近藤, 勅使河原, “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・システム管理の実現方式”, DICOMO'98 シンポジウム論文集, pp.109-114 (1998)
- [3] AdventNet Inc., “AdventNet Agent Builder 2.0”, “AdventNet SNMP Release 2.0”, <http://www.adventnet.com/>