

分散オブジェクトプラットフォーム上に構築したネットワーク 管理システムの評価

藤崎智宏 蔭山克禎 浜田雅樹

NTT ソフトウェア研究所 広域コンピューティング研究部

ネットワークの発達に伴い、ネットワーク管理システムも広く用いられるようになってきたが、既存のネットワーク管理システムには、管理ネットワーク機器の増大に対する対策、ネットワーク管理システム自身の信頼性、個々のネットワークに適合した管理の提供の難しさ、といった問題点が存在する。これらの問題点を解決するネットワーク管理システムとして、分散オブジェクトプラットフォーム上に『ネットキーパー』を構築してきた。『ネットキーパー』においては、分散オブジェクト通信機構によるスケーラビリティの確保、オブジェクト監視による信頼性の向上、オブジェクト指向フレームワークの構築によるネットワーク管理ソフトウェアのカスタマイズの容易化を行っている。本稿では、ネットワーク管理システムのプラットフォームとして分散オブジェクト通信機構を使用することによる利点・欠点について考察し、『ネットキーパー』を例として、分散オブジェクトプラットフォームを用いることによるスケーラビリティの向上に関しての評価について報告する。

Evaluation of a Network Management System based on a Distributed Object Platform

Tomohiro Fujisaki Katsuyoshi Kageyama Masaki Hamada

NTT Software Laboratories, Global Computing Laboratory

This paper describes impacts upon a network management system (NMS) by using the distributed object technology, and a scalability evaluation of an NMS based on a distributed object platform. The features of the distributed object platform such as flexibility, scalability and fault tolerance are considerably fit to a network management system. One example is our NMS *NetKeeper*, which is built on a distributed object platform. We explain the fitness from our development experience and evaluate the scalability of the distributed object platform used by *NetKeeper*.

1 はじめに

ここ数年でコンピュータネットワークは社会的に浸透し、コンピュータの専門家以外でもネットワークを利用することがごく一般的になってきている。利用の形態も、従来の電子メールや電子ニュースといったアプリケーションに加え、遠隔教育、電子

商取引などの高度なアプリケーションが増加しており、コンピュータネットワークの重要性は増大の一途である。一方でコンピュータネットワークを運用する側には、これまで以上に安定したネットワークの管理・運用が求められているが、現状ではコンピュータネットワークの運用は一部のエ

キスパートといった人的資源に負うところが非常に大きい。昨今、多くの組織が参加しており、商用利用もすすんでいるインターネットを例に取って見ても、組織におけるネットワークの運用・管理を円滑に行うにはかなりのスキルを有した担当者が必要であり、トラブルへの対応など担当者の経験に頼る場面が多い。しかしながら、管理対象ネットワーク中に存在する機器の数が増加するにつれ、少数のエキスパートがネットワーク全体を管理することは困難になってくる。

適切なネットワーク管理システム(NMS)を導入することで、上記のような人的資源への依存度を減らし、少ない担当者数でも円滑にネットワークの管理・運用を行うことを可能にできる。商用/フリーなものなど数多くのNMSが利用可能であるが、現状のNMSには、大規模ネットワークを管理することが困難である、カスタマイズや機能拡張が難しい、分散環境を生かした管理がしにくい、といった問題点がある。分散環境を利用して管理をするために、NMS どうして通信を行う機能を持つものもあるが、通信内容が限定されており、分散環境を十分に利用した管理することは難しい。

本稿では、分散オブジェクト技術を利用することで、NMSにおける大規模ネットワークへの対応、個々のネットワークやネットワーク管理者向けのカスタマイズの容易化、分散環境を利用した管理が容易に行えることを、我々が分散オブジェクトプラットフォーム上に開発したNMS『ネットキーパー』[Arano95]を例として説明する。

2 ネットワーク管理システム

ネットワーク管理の省力化、ネットワーク異常の早期発見などのために、ネットワーク管理システム(NMS)を導入する組織が増加している。商用/フリーなものなど数多くのNMSが存在し利用可能であるが、現状のNMSは以下のような問題点を抱えている。

1. 大規模なネットワークを管理することが困難である
既存のNMSは基本的に一つのGUI付きの単一のプログラムである。多数のネットワーク機器で構成される大規模なネットワークを監視するためには、複数のNMSを配置して領域分割などにより管理することになるが、複数のNMSでの情報交換は容易ではなく、統合的に、柔軟な管理を行うことは難しい。
2. カスタマイズや機能拡張が難しい
OSIの管理構造モデルなど、ネットワーク管理に必要であると標準化された機能をベースにシステムが作成されており、この標準以外の機能を必要とするような拡張が難しい。このため、特定の組織内の運営ポリシーなどを反映した管理システムを構築することが困難である。
3. 分散環境を生かした管理がしにくい
障害が発生した場合には、障害の起きた場所でNMSによる情報収集、機器のテストなどを行うことができれば、障害回復に有効であるが、既存のネットワーク管理システムは管理用のマシンのある場所でのみ使用可能な場合が多い。

これらの問題に対し、以下のような解決のアプローチがとられているが、それぞれのアプローチは十分ではない。

1. 複数のネットワーク管理ステーション間での情報伝達機構の作成
NMS間で情報をやり取りする機構を導入し、複数のネットワーク管理ステーションを使用して大規模なネットワークを管理することを可能にしている[RFC1451]。しかしながら、NMS間で受け渡しできる情報は十分ではなく、また、分散管理を行うための管理システムの論理的構造が固定されており、管理目的に合わせて柔軟に複数NMSの論理構造を変えるといたことができない。

2. パラメータ設定などによるカスタマイズ
規定した枠組みの中で、ある程度のカスタマイズを可能にしているが、パラメータの設定などでは対応できない要求が多い。
3. 構造を規定した分散監視
ネットワーク管理ステーションを木構造に配置し、上記の NMS 間通信を使うことで分散監視を可能にしている。静的な木構造に基づき分散監視が可能であるが、木構造の同じノードの下にない NMS 同士では通信が出来ないため、管理できる内容に制限があるなどの問題が生じている。

上記の問題点を解決するために、分散オブジェクト通信環境を用い、この上に NMS を構築した。分散オブジェクト通信環境を用いることで、この環境の特徴であるスケラビリティの利用による NMS の大規模ネットワークへの対応、オブジェクト指向技術を用いた個々のネットワークやネットワーク管理者向けのカスタマイズの容易化、分散オブジェクト通信環境を利用した分散監視の実現が可能である。

次章では分散オブジェクト環境に関して簡単に説明し、構築した NMS 『ネットキーパー』を例として、NMS を分散オブジェクト通信環境上に構築することにより、上記の問題点が解決出来ることを示す。

3 分散オブジェクト環境と NMS

3.1 分散オブジェクト環境

分散オブジェクトとはオブジェクト指向の概念を分散環境に拡張したものである。従来のオブジェクト指向実行環境では、目的システムを構成するオブジェクトが一つの計算機中の一つのプロセス内に存在していた。分散オブジェクトの実行環境では、システムを構成するオブジェクトをネットワーク中に分散配置することが可能である。システムは、オブジェクトどうし

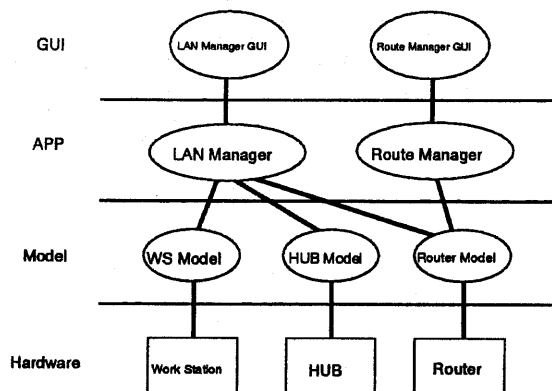


図 1: Structure of the NetKeeper

がネットワーク中でメッセージを送受信することにより動作する。オブジェクトどうしのメッセージ送受信は、オブジェクトの“名前”とそのオブジェクトに対し動作を指示する“メソッド”を指定することにより行われ、オブジェクトが実際にどのマシン上にあるかを気にしなくてよい。また、既存の socket 等を用いたプロセス間の通信に比べ、非常に簡便に使用できるため、開発も容易である。

3.2 ネットキーパーの構成

分散オブジェクト環境上に構築した NMS, 『ネットキーパー』の構造を図 1 に示す。

NetKeeper では管理対象であるネットワークを 4 層構造に分けてモデル化している。特徴的なのは、管理対象機器に対しその代理エージェントである“モデル”を構築し、モデルに分散オブジェクト呼出によるインタフェースを持たせるという構成をとっていることである [藤崎 95]。モデルは管理対象機器に 1 対 1 に対応して存在し、複数のプロトコルを用いて実際にネットワーク機器にアクセスするが、上位のアプリケーション層に対しては、複数プロトコルの存在を分散オブジェクトメッセージ交換機構で隠蔽する。

ネットキーパーでは、ROF (Remote Ob-

ject Facility) と呼ばれる分散オブジェクト通信機構を利用している。ROF は、ネットワーク上に分散したオブジェクトに対し、オブジェクトの名前を指定することによりメッセージが送信可能である。

以下、ネットキーパーの各層について詳述する。

Hardware 層

ネットワーク上に存在するルータ、ハブ、ワークステーションなどの管理対象機器そのものの層である。

Model 層

管理対象となるネットワーク機器からの情報の取得、機器の制御を行う“モデル”の層である。“モデル”は各機器のエージェントであり、管理対象機器一台に対し一つ存在する。機器からの情報を保持し、各種設定の指示を機器に送る。

APP(Application) 層

Model 層に存在する管理対象機器のモデルにアクセスし、ネットワークの管理をするアプリケーションの層である。管理対象機器のモデルを組み合わせ、組織のネットワーク管理ポリシーにあったアプリケーションを構築できる。

GUI 層

管理情報の表示、機器への指示を人間が行う部分である。GUI とアプリケーションを切り離したことにより、ユーザに合わせて容易に GUI を変更できる。また、GUI を複数の場所で実行し、同じアプリケーションにアクセスすることで離れた場所で同一の情報を共有することができる。

ネットキーパーでの NMS の問題の解決方法について説明する。

1. 大規模ネットワークを管理することが困難である
分散オブジェクトプラットフォームで

利用できるホストを付け足し、分散オブジェクトを再配置することで目的システム全体としてのスケーラビリティを向上させることが可能である。

2. カスタマイズや機能拡張が難しい
オブジェクト指向で用いられる MVC モデル [Cunningham86] の利用により、管理アプリケーションとその GUI を分離することで GUI 部分のカスタマイズが可能である。GUI を作成する際には、あらかじめ用意されている部品を用いることができる。また、GUI のみでなく、部品を組み合わせることにより既存アプリケーション本体の変更も容易に行える。
3. 分散環境を生かした管理がしにくい
分散オブジェクトの位置透過性を利用することで、オブジェクトに名前だけでアクセスできるため、分散オブジェクト指向環境がカバーするネットワーク中のどこにいても、同じ環境でのオブジェクトアクセスが可能である。すなわち、管理ネットワーク中の場所に依存せずに、同じ環境で NMS を利用することが可能となる。

4 分散オブジェクト通信の評価

分散オブジェクト環境を利用することにより、大規模なネットワーク管理に対応できるスケーラビリティを得ることを示すための実験を行った。

4.1 分散オブジェクト環境の性能評価

分散オブジェクト (ROF)、SNMP [Rose94]、リモート手続き呼び出し (RPC) の各通信機構の性能を比較するための実験を行った。

4.1.1 実験環境

二つのサブネットを一台のルータでつないだ図 2 のようなネットワーク上で、監視

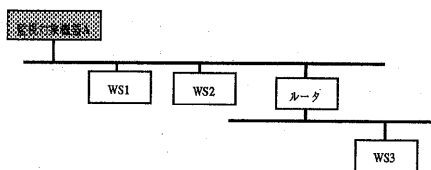


図 2: 分散オブジェクト性能評価実験

WS2 上から,WS2 のモデル経由	3.48 ms
WS2 上から,WS1 のモデル経由	6.95 ms
WS2 上から,SNMP	1.98 ms
WS2 上から,RPC	480.37 ms
WS3 上から,SNMP	3.67 ms

表 1: 通信時間測定結果

対象機器 A の管理を行うモデルを WS1, WS2 で起動する. このモデルは, 分散オブジェクト通信機構による要求を受け取ると, SNMP を用いて監視対象機器 A から情報を得ようになっている.

4.1.2 実験内容・測定結果

WS2, 3 上から, 分散オブジェクト通信機構, SNMP, RPC を用いて監視対象機器 A より情報を取得する. 測定した結果は表 1 の通りである.

表 1 より, 分散オブジェクト通信の所要時間は,

- ローカル 1.50 ms
- リモート 4.97 ms

となる.

4.1.3 考察

以上より, ROF を用いることによって, SNMP のほぼ 3 倍程度の時間がかかることがわかった. これは, ROF には, 名前空間の維持管理機構によるオーバーヘッドが存在することと, 信頼性を確保するために, 下位プロトコルとして TCP/IP を使用しているためである. RPC による通信が, 非常に時間がかかっているが, これは

今回の測定では, 呼び出し時間に管理対象機器 A 上で返答のためのプロセスが起動される時間が含まれているためである.

隣接した別セグメントからの SNMP によるアクセス (WS 3 上からのアクセス) は, 3.67ms である. 複数のルータが管理対象機器との間に介在するような環境では, ローカルマシンに立ち上げたモデルをキャッシュとして用いることにより, パフォーマンスを向上させることが可能であることがわかる.

4.2 スケーラビリティの評価

分散オブジェクト環境を利用することにより, スケーラビリティを確保できることを確認するための実験を行った.

4.2.1 実験環境

実験を行ったネットワーク構成を図 3 に示す. 単一ネットワーク上に, ネットワーク管理用マシンを 3 台配置しそのマシン上で動作するモデルを用いて, 同一セグメント上にある監視対象機器 (ルータ) を監視する.

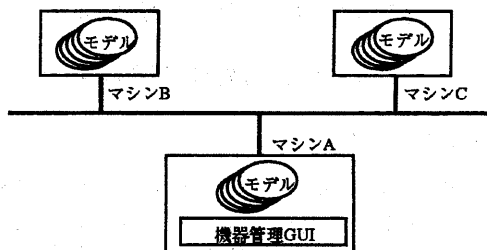


図 3: スケーラビリティ評価実験

4.2.2 実験内容・測定結果

マシン A 上のネットワーク機器管理 GUI が, 機器管理を行っているモデルに対し, アクセスするためにかかる時間を測定する. 機器管理 GUI は, 各モデルから, 各モデルが監視する監視対象機器の起動時

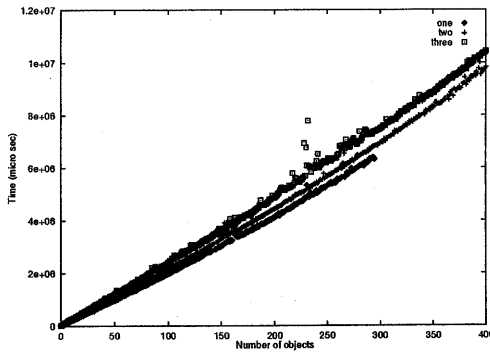


図 4: 測定結果

間(sysUpTime)を取得する。その際の、全てのモデルから一通り情報を取得するのにかかる時間を、

1. 単一マシン(マシン A)上でネットワーク管理 GUI とモデルを動作させた場合(one)
2. モデルを、マシン A とマシン B で均等に分割した場合(two)
3. モデルを、マシン A、マシン B、マシン C で均等に分割した場合(three)

について測定した。測定結果を図 4 に示す。

4.2.3 考察

図 4 で、one のグラフが途中でとぎれているが、これは、オブジェクト数が 292 で、評価用マシンがクラッシュしたためである。

モデルを動作させるマシンを増やしていくと、応答時間が増加していく。これは、分散オブジェクトの管理用通信、オブジェクト間通信の増大から、通信コストが大きくなっていくためである。一台のマシンでは 291 オブジェクトしか処理できないが、マシンを複数使用することで、限界オブジェクト数を増やすことができた。

この実験では、各モデルは要求に返答、管理対象機器への 30 秒ごとのポーリング

以外には何も行っていない。モデルが、収集したデータに基づくトラフィックの予想などの計算量の多い作業をする場合には、ネットワークの通信コストよりも CPU の使用コストの方が大きくなり、オブジェクトを分散させることによってシステム性能を向上させることができると思われる。

5 まとめ

本稿では、既存のネットワーク管理システムが抱えている問題に関し、分散オブジェクト通信機構を用いることで解決できることを、NMS『ネットキーパー』構築の経験より示した。また、分散オブジェクト環境を利用することにより、大規模なネットワークの管理への対応といった、スケーラビリティを確保できることを、実験により確認した。

今後は、個々のオブジェクトの計算量と通信コストの比較、分散させるオブジェクトの最適配置に関する評価を行っていく。

参考文献

- [Arano95] Takashi Arano et al. "A Computer Network Management System Platform based on Distributed Objects", IFIP/IEEE DSOM '95
- [Rose94] MARSHALL T. Rose "THE SIMPLE BOOK second edition" Prentice Hall, 1994 ISBN 0-13-177254-6
- [藤崎 95] 藤崎 他 "分散オブジェクト指向プラットフォームに置けるルータの運用・管理" 情報処理学会分散システム運用技術研究グループ研究会 9501028
- [Cunningham86] Ward Cunningham "Smalltalk-80 によるアプリケーションプログラムの作り方" bit, Vol.18 No.4, pp.379-395, 1986. ソニー・テクノロジクス(株)AI 営業部 抄訳
- [RFC1451] J. Case et al. "Manager-to-Manager Management Information Base", Request for Comments 1451, Apr 1993