

イベント駆動型管理戦略に基づく ネットワーク性能管理方式の提案

斉藤 裕樹 中野 武司

{hiro-s, chusho}@cs.meiji.ac.jp

明治大学大学院理工学研究科基礎理工学専攻

概要

TCP/IP インターネット技術の普及にともない、サーバへのアクセス速度や応答性能といったユーザの利便性を確保するための性能面でのネットワークサービスの品質が求められている。

本報告では、ネットワークシステムにおける性能問題を分析し管理を支援する方式を提案する。システムをサブシステム同士の依存関係に分割し、トップダウンの分析を行う方法を提案し、この方式に基づいたイベント駆動型の管理戦略を用いることで、これまで経験を積んだネットワーク管理者が手作業で行っていた問題の発見、切り分け、対処という一連の作業自動化することができ、管理作業を低減することができると考えられる。

Performance Management for Network Systems with Event-driven Strategy

Hiroki SAITO and Takeshi CHUSHO

Graduate School of Science and Technology, Major in Science, Meiji University

Abstract

One of major issues in Internet is performance management for usability such as access speed and response. This paper proposes an analysis method for performance management of network systems, and its application. We consider network systems as subsystems and their dependence by focusing on functions and data flow. The problem can be handled automatically by the event-driven strategy which is based on the subsystems and the dependence. This method reduces the network systems management work.

1 はじめに

インターネットの展開とその技術を用いた企業、大学などのネットワークの増加と規模の拡大に伴い、データ転送が遅い、転送が中断されるなどのネットワークシステムの性能面での問題が顕在化している。スループットの保証や応答速度の改善といった、ユーザの立場でのネットワークサービスの性能を確保するネットワーク性能管理技術が望まれている。

従来ネットワーク管理システムは、端末間の到達

性の保証や、リンクの利用状況の監視を目的とし、主にネットワーク層の機器管理を中心としたものであった [1]。しかし、インターネットの主要アプリケーションで用いられる TCP の性能はネットワーク層の遅延など特性によって大きく悪化することが知られている [2, 3]。

さらに、ネットワークシステムの性能は、ネットワークの帯域やルータの処理能力といった性能だけではなく端末間のスループットやサーバシステムやクライアントシステムの性能などに左右され、要因

は多岐にわたっている。

従来、性能管理はこのようなネットワークの知識を持った管理者によって行われてきた。しかし、ATM、Gigabit Ethernet などをはじめとする新しいネットワーク技術が次々導入されシステムは複雑化している。にもかかわらず、今後ネットワークに精通していない者が管理を行う必要性が高まると思われる。

本報告では、このような問題を解決するためにネットワーク性能管理作業を支援する技術について検討する。

2 ネットワークの性能管理

ネットワークシステムの性能管理とは、何らかの問題で性能を発揮できないネットワークシステムに対して、問題を分析し特定し、その対策を行うことである。

従来、SNMP やその応用技術 [4, 5, 6] を用いネットワークの各ノードに注目し、ping, traceroute, ttcp やその他様々なツール [7, 8, 9, 10] を用い端末間の通信性能を測っていた。またアプリケーションの性能測定の方法はそれぞれ異なるため、特に考慮されていないか、アプリケーション独自の方法を用いることが多い。ネットワークシステムの性能は、各リンクの利用状況、端末間のスループット、アプリケーションの性能などによって左右されるため、必要に応じてそれら全てに注目しなければならない。

ここでは、ユーザの感じる性能に基づいた管理が行えるネットワークシステムの管理方式を提案する。

まず、管理対象となるサーバ、クライアント間でアプリケーションの応答速度や実行速度を測りアプリケーション層での性能を測定する。そして、アプリケーション層から利用されるネットワーク層の性能と比較対照することで、性能問題がアプリケーション層を原因とするものなのか、ネットワーク層以下を原因とするものなのかを判断する。

ネットワーク層に問題があれば、そのアプリケーションが利用しているネットワークのうち性能劣化が起きているネットワークを特定する。

次に、問題のあるネットワークに対してデータリンク層の問題なのか、ネットワーク層の問題なのかを特定する。これにはルータやスイッチなどの中継

装置のトラフィック量やエラー率などを調べ、過去のデータやあらかじめ設定したしきい値などと比較することで問題の有無を判断する。

このようなトップダウンの問題分析は、従来管理者の手作業によって問題を切り分けることで行ってきた。しかし、このような作業は煩雑であり、正しく原因を突き止めるには、アプリケーション層、ネットワーク層、データリンク層の各技術やネットワーク構成に関する知識が要求される。したがって、これを支援するシステムが望まれる。システムには以下の機能が求められている。

1. ネットワークシステムを構成するアプリケーション、ネットワーク、データリンク各層のサブシステムを明確にしその分析手順を定義する機能
2. 各サブシステムのエラー率やトラフィック量などの性能を測定し記録する機能
3. サブシステムの性能データとサブシステム同士の関係から、問題発生時に原因と考えられる個所を分析、特定する機能
4. 管理者に警告を発生する、代替機器に切り替えるなどの問題発生個所に応じて対処を行う機能

これらの機能を利用することで、性能問題が発生した場合の分析と対処を自動化することができると考えられる。また、監視データの記録からボトルネックを解消するためのネットワーク拡張の指針を立てることができる。

3 性能管理問題の定式化

本手法における性能管理の定式化手順を以下に示す。

1. 性能管理モデルの構築

- (a) 対象システムの明確化と依存関係の定義
クライアント、サーバ、およびその間のネットワークによって構成されるネットワークシステムから、性能に影響を与える要素をサブシステムとして洗いだし明確化する。システム間のデータのやり取りを中心とした依存関係を抽出し、システムとサブシステムの間の関連づけを行う。

(b) 性能測定方法の定義

前段階で明らかになったサブシステムに対して、性能を測るためには、どのようなデータを測定すればよいのか測定内容と測定方法を定める。

2. イベント駆動による管理戦略の定義

管理戦略とは、性能管理モデルに対しての分析方法と異常時の対処方法である。

問題が起ったときには、性能管理モデルで定義したサブシステム間の依存関係に従いイベントを伝搬させることで問題分析を行う。

次章以降で各手順について説明していく。

4 性能管理モデルの構築

4.1 対象システムの明確化と依存関係の定義

ネットワークシステムには、サーバアプリケーションやクライアントアプリケーションのようなソフトウェアシステム、端末計算機の OS のネットワーク通信機能、ルータやデータリンクメディアのようなネットワーク網の各要素、DNS や DHCP などの基幹サービスなどの多数の構成要素が含まれている。そして、これらの間にはデータのやり取りを中心とした機能の依存関係が存在している。

これらの構成要素をどのように捉えるかを明確にするために、性能管理モデルを構築する。まず、クライアントアプリケーションとサーバアプリケーションとのやり取りの観点から、関係のあるサブシステムを再帰的に抽出する。次に、システムとサブシステムの依存関係に基づいた関連づけを行う。

各システム間の関係には以下の 3 通りが考えられる。

1. 計算機内のサブルーチンコールやモジュール機能間の依存関係

アプリケーションから通信を行う場合、OS のシステムコールを用いる。OS 内のネットワーク機能は、インターネットのレイヤ化アーキテクチャに基づいたモジュール構成をとっている。そこで、システムコールやモジュール間の機能の依存関係に注目する。

2. 端末内の各層同士の相補的な依存関係

インターネットはレイヤ化アーキテクチャによって構成されているため、端末間での各層独立で対等な論理的な通信に注目する。例えば、ネットワーク層での通信は、下位のデータリンク層とは独立の論理的な IP データグラム

3. ネットワーク構成機器の通信機能の依存関係

実際の通信は、物理的なネットワークメディアやルータ、スイッチなどの中継機器を通して行われているので、各構成機器間でのデータのやり取りに注目した機能の依存関係に注目する。

図 1 に性能管理モデルの全体図を示す。従来のツールがネットワーク網や個々の通信ノードに注目していたのに対し、本モデルは、直接ユーザが接するクライアントアプリケーションの視点でのシステムの性能に注目したモデルであることが大きな特徴である。

抽出された対象システムの依存関係は有向グラフで表される。このような依存関係を抽出することによって、あるシステムに対して性能に影響を与えるサブシステムを明確にすることができる。すなわち、システム S において性能劣化が起った場合、 S を構成するサブシステム $S_0, S_1, \dots, S_n$ または S 自身に原因があると予測することができる。

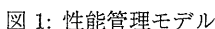
4.2 性能測定方法の定義

明確になったサブシステムに対し、性能に影響を与える測定データを明らかにする。

トランスポート層でのデータ通信機能、IP データグラムの交換機能、データリンクドライバはノード間でのデータ通信を行うサブシステムである。これらの性能は一般に、スループット、遅延の 2 つの観点から表される。

また、IP ネットワーク、データリンクメディアはノード間の伝送を行うものであるため、トラフィック量とエラー率を測ることで性能を知ることができる。一般的にネットワークは負荷によって伝送効率が低下するので、ネットワークやそれをつなぐ中継装置の利用率を調べることで性能が把握できる。また、エラー率を測定することで問題を発見することができる。

アプリケーションゲートウェイ、ルータおよびデー



行い、評価し、その結果どのような対処を行うのかを定義することである。

5.1 管理対象オブジェクト

有向グラフによって表された依存関係はトップダウンによる問題分析手順と一致する。管理アプリケーションでは、管理対象の各サブシステムをオブジェクトとして表し、異常の原因であるシステムを探し出す。これは、依存関係を表すツリー構造を幅優先探索することで実現することができる。また、分析を行っている途中で不要部分の枝刈りを行うことでシステムへの不用意な測定を防ぐ。

管理対象オブジェクトはシステムが正常か異常かを表す内部状態を持ち、状態遷移の際に下位のオブジェクトにイベントを伝搬する手法を用いる。対象

管理対象オブジェクト
state:enum{正常, 異常疑いあり, 異常} name:String 性能測定のための各種パラメータ (ネットワークアドレス, SNMPコミュニティ名)
probe():void 「正常」時の定期的な監視手続き evalProbe(watchedData):bool 「正常」時監視項目に対し「異常疑いあり」を判定 probeAgain():void 「異常疑いあり」時の監視手続き evalProbeAgain(watchedData):bool 「異常疑いあり」時監視項目に対し「異常」を判定 actProblem(watchedData):bool 「異常」時の対処手続き

図 2: 管理対象オブジェクト

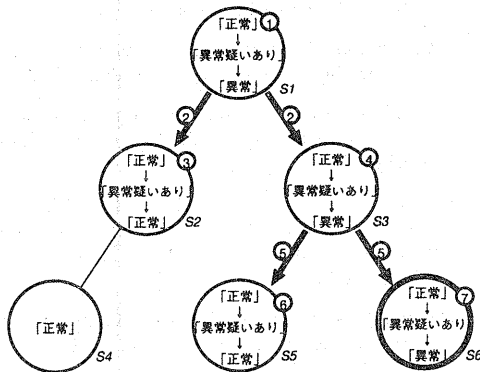


図 3: イベント伝搬の例

システムは、「正常」、「異常疑いあり」、「異常」の3状態を持つこととし、各状態間の2段階評価を行うこととした。

管理対象オブジェクトの構成を図2に示す。

5.2 イベント伝搬による問題分析

図3に示すシステム間の依存関係を例にあげる。性能問題がシステムS1で顕在化し、原因はシステムS6であった場合、問題分析は次のような手順で行われる。

1. S1が通常測定手続きS1.probe()を実行しS1.evalProbe()で評価し結果から、「正常」から「異常の疑いあり」に遷移し、

S1.probeAgain()を実行する。

S1.evalProbeAgain()による評価の結果異常と判断し「異常」に遷移。

2. S1から下位のサブシステムS2,S3に対して異常イベントを通知
3. S2はイベントを受けて「正常」から「異常の疑いあり」に遷移し、S2.probeAgain()を実行する。S2.evalProbeAgain()による評価の結果異常がなく「正常」に戻る
4. S3はイベントを受けて「正常」から「異常の疑いあり」に遷移し、S3.probeAgain()を実行する。S3.evalProbeAgain()による評価の結果異常と判断し「異常」に遷移。
5. S3から下位のサブシステムS5,S6に対して異常イベントを通知
6. S5はイベントを受けて「正常」から「異常の疑いあり」に遷移し、S5.probeAgain()を実行する。S5.evalProbeAgain()による評価の結果異常がなく「正常」に戻る
7. S6はイベントを受けて「正常」から「異常の疑いあり」に遷移し、S6.probeAgain()を実行する。S6.evalProbeAgain()による評価の結果異常と判断し「異常」に遷移。

5.3 イベント伝搬による問題対処

異常時には各対象オブジェクトのactProlem()を実行し対処を行う。

さらに、ログ記録、可視化機能、監視レポートや警告をメールで送信などの管理上必要な諸機能や、代替機への切り替えなどの異常時の対策機能を持つ管理ユーティリティオブジェクトを用意する。異常イベントをこれらのオブジェクトへ伝えることによって問題発生時の対処を自動的に行う。

5.4 オブジェクトのカスタマイズ

ルータ、スイッチ、Ethernetなどネットワーク管理上標準的な対象オブジェクトや、頻用する管理ユーティリティオブジェクトは、あらかじめライブラリとして用意し必要に応じてカスタマイズし利用する。また、新たにオブジェクトを作成する場合にも、一

一般的な形のオブジェクトのひな形を用意し、必要な部分を作成するのみにする。

カスタマイズ方法としては、システムを測定するのに必要なネットワークアドレスやSNMPコミュニティ名などのプロパティの設定、測定のための手続きの再定義、評価関数のしきい値や関数自体の定義が行えるようにする。

6 現状と今後の課題

現在、上記管理戦略に基づくネットワーク性能管理アプリケーションを作成・評価中である。トップダウンモデル、分析方法、性能評価の妥当性および、性能測定における実運用への影響、管理作業の省力化効果の評価などを行う予定である。

今後の課題と検討事項を以下に示す。

1. 多クライアント多サーバへの対応

本研究での性能管理モデルでは1クライアント対1サーバによる分析を行っている。しかし、実際の運用形態は多クライアント対多サーバであり、さらに多システムが1つのネットワーク上で運用されている。

しかし、これは1対1のモデルの考え方を拡張することで実現可能であると考えられる。同じシステム間の依存関係が多くのシステムで適用されるようなケースでは、オブジェクトの再利用が有効であると考えられる。

2. 評価関数の設定

システムの性能問題の有無の判定は管理対象オブジェクト内の評価関数で決定される。理論上の最大性能が決まっているものは、しきい値を決めることは容易であるが、例えばサーバの最大同時処理能力などは様々な要因を考えなければならないので、運用状況に応じて適当に調整しなければならない場合もありうる。

7 おわりに

本報告では、TCP/IP インターネット技術を用いたネットワークシステムにおける性能問題を分析し管理を支援する方式を提案した。

システムをサブシステム同士の依存関係に分割し、

トップダウンの分析を行う方法を提案した。この方式に基づいた分析と対処を自動化するためのイベント駆動型の管理戦略を用いることで、これまで経験を積んだネットワーク管理者が手作業で行っていた問題の発見、切り分け、対処という一連の作業自動化することができ、現在のネットワークで頻発しているネットワーク性能の問題を軽減することができると考えられる。

参考文献

- [1] Leinwand, A.: Accomplishing Performance Management with SNMP, Proc. of INET'93 (1993).
- [2] Mark Allman, Chris Hayes, Hans Kruse, Shaen Osterman: TCP Performance over Satellites Links, Proc. of the 5th International Conference on Telecommunications Systems (1997).
- [3] V. Jacobson, R. Braden, D. Borman: TCP Extensions for High Performance, RFC1323 (1992).
- [4] D. Harrington, R. Presuhn, B. Wijnen: An Architecture for Describing SNMP Management Frameworks, RFC2271 (1998).
- [5] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser: Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1907 (1996).
- [6] S. Waldbusser: Remote Network Monitoring Management Information Base Version 2 using SMIV2, RFC2021 (1997).
- [7] Netperf: A Network Performance Benchmark Rev.2, Hewlett-Packard, (1995).
- [8] Jonkman, R.: NetSpec: A Network Performance Evaluation and Experimentation Tool, <http://www.tisl.ukans.edu/Projects/AAI/products/netspec> (1995).
- [9] Keshav, S: REAL: A Network Simulator, Computer Science Technical Report, University of California, (1988).
- [10] 村山 公保, 門林 雄基, 山口 英: TCP 性能評価システム DBS の構築, コンピュータソフトウェア Vol.15, No.2, pp.24-37 (1998).
- [11] 北橋 雅子, 野口 正一: オブジェクト指向によるネットワーク管理問題分析, 情報処理学会論文誌, Vol.35, No.6, pp.1162-1171 (1994).
- [12] 布川 博士, 堀切 敬一, 新井 恒博, 福田 健一, 村田 美穂, 風間 敬一, 増永良文: オブジェクト指向に基づくネットワーク環境情報システム, 電子情報通信学会論文誌, D-1, Vol.J79-D-1, No.10, pp.873-883 (1996).