

分散システムにおけるアプリケーション QoS 管理手法

加藤 由花 箱崎 勝也

電気通信大学大学院情報システム学研究科

概要

本稿では、分散システムにおけるアプリケーション QoS (Quality of Service) に着目し、ポリシーに従ったシステム運用を可能とする、QoS 管理ポリシー設定手法を提案する。本手法は、アプリケーション QoS を基に決定されるアプリケーションレベルの制御ルールと、アプリケーションのトラフィック特性を基に決定されるネットワークレベルの制御ルールの組み合わせにより実現される。その結果、管理ポリシーの決定支援、複数アプリケーション間の QoS ネゴシエーションが実現され、システム全体として管理ポリシーに従ったアプリケーション運用が可能となる。本稿ではまた、この QoS 管理ポリシー設定機能の実装例を紹介する。

Application QoS Management for distributed Computing Systems

Yuka KATO and Katsuya HAKOZAKI

Graduate School of Information Systems, University of Electro-Communications

Abstract

This paper proposes a setting method of QoS (Quality of Service) management policies from a viewpoint of application QoS in distributed computing systems. In this method, the QoS control is done by combination of the control rule on an application level based on the application QoS and the rule on a network level based on the characteristics of the application traffic. The method supports the decision of the management policy, and makes it possible to negotiate QoS among many applications. As a result, application management based on the policies becomes possible in the system. An implementation example of the setting function also shows in the paper.

1. はじめに

CORBA (Common Object Request Broker Architecture)¹⁾等の分散オブジェクト技術の発達により、コンピュータネットワーク上には多くの分散アプリケーションが導入されるようになってきた。特にビジネス分野では、戦略的に情報システムが利用され、数時間のシステム異常、障

害が企業に大きな損失を与えるようになってきた。一方、社内 LAN 上には様々なアプリケーションが共存しており、基幹業務アプリケーションの性能が、他の優先度の低いサービスの影響で劣化してしまうケースも起きている。そのため、各アプリケーション毎に管理ポリシーを設定し、このポリシーを満足するよう情報システムを運用する、という考え方が一般化しつつある。そこで

は、各アプリケーションが、利用者とシステム間であらかじめ合意されたサービス品質 (Quality of Service: QoS) を満足しているかが常に監視され、このサービス品質に基づくシステム管理が行われる。

ここで監視対象となるサービス品質は、各アプリケーションにおいて利用者に対して提供されるサービス品質、つまりアプリケーションの QoS である (以後、アプリケーション QoS と記す)。このアプリケーション QoS は、利用者ごと、アプリケーションごと、さらに利用環境ごとに異なる評価基準を持つ。従って、分散アプリケーションの効率的な運用には、アプリケーションへの要求条件、利用環境、利用者の選好への要求を考慮した、情報システムアーキテクチャ、およびアプリケーション運用手法が必要となる。アプリケーション QoS の例を表 1 に示す。

表 1 アプリケーション QoS の例

アプリケーションの例	アプリケーション QoS
電子メールを送る	メールの到達性
オンラインショッピング	受注確認時間
Webへのアクセス	ページ表示時間
ビデオの視聴	画質
IP電話	会話の遅延時間
ダイヤルアップ接続	接続率

近年、このアプリケーション QoS に着目し、QoS 制御を実現するためのシステムアーキテクチャ、制御手法等の研究が盛んに行われている。IETF では、ネットワーク上で特定コネクションの帯域保証を行う RSVP²⁾や、ネットワークが複数の QoS を柔軟に提供するためのメカニズムである DiffServ³⁾の標準化が行われている。また、ネットワーク利用者が能動的にネットワークを制御する技術として、Active Network⁴⁾に関する研究がある。さらに、Check Point Software Technologies の FloodGate-1⁵⁾のように、管理ポリシーに基づきネットワーク制御を実現するポリシーサーバの製品化が行われている。これらの技術では、アプリケーション毎にルールの記述が

必要であり、管理ポリシーを事前に決定しておく必要がある。しかし、システム全体でポリシーに従った運用を行うためには、どのように管理ポリシーを決定したら良いか、また複数アプリケーション間の QoS ネゴシエーションをどのように実現するか、という問題を解決する必要がある。

一方、エンド-エンドでのアプリケーション QoS を制御対象とする研究としては、QoS プロトコル⁶⁾や QoS を考慮したやわらかいマルチメディアシステム⁷⁾がある。これらの研究は、リソースの割り当て方法や、システムによるアプリケーション QoS のマッピング方法を研究対象としている。しかし、システム構成が複雑化し需要変動予測の困難な実システムにおいては、より単純な制御方式が望ましい。

これらの観点から、我々は、分散アプリケーションの適応的運用に着目し、測定駆動型の簡易な制御を実現する、アプリケーション QoS 管理システム (application QoS Management System: QMS) の研究を進めている⁸⁾。QMS は、アプリケーション QoS 管理のロール毎に定義される 3 種類のモジュールから構成され、ネットワーク上に分散配置されるモジュール間の通信によって、適応的な運用を実現する。本稿では特に、QMS における QoS 管理ポリシー設定機能に着目し、その設定手法を提案する。本設定手法により、管理ポリシーの決定支援、複数アプリケーション間の QoS ネゴシエーションが実現され、システム全体として管理ポリシーに従ったアプリケーション運用が可能となる。

以下、2 章で QMS の構成について述べた後、3 章で QoS 管理ポリシー設定手法を提案する。4 章でその実装例を示し、5 章で本稿をまとめる。

2. 管理システムの構成

2.1 システム概要

QMS は、アプリケーション QoS を制御対象とし、分散アプリケーションの適応的運用を目的に設計される。その設計コンセプトは、(i)アプリケーション QoS を制御対象とする; (ii)システム環

境の変化に柔軟に対応できる、適応的な運用を行う; (iii)複雑なモデル化を伴わない、簡易な制御手法を採用する、の3点である。QMSは、アプリケーションQoSを制御対象とするが、これを各リソースにマッピングすることなしに制御を実現する。そのため、システム環境の変化を検知した場合、複雑なモデル化を伴わない簡易な制御方法によって、フィードバック的な制御を行う。

QMSは、以下の3種類のモジュールで構成される。(i)Managerモジュール:アプリケーションのQoS管理ポリシーを保持し、制御方法を決定する;(ii)Notificatorモジュール:アプリケーションQoSパラメータを測定、監視する;(iii)Controllerモジュール:管理ポリシーに基づき、制御を実現する。

2.2 システムアーキテクチャ

QMSのシステムアーキテクチャを図1に示す。

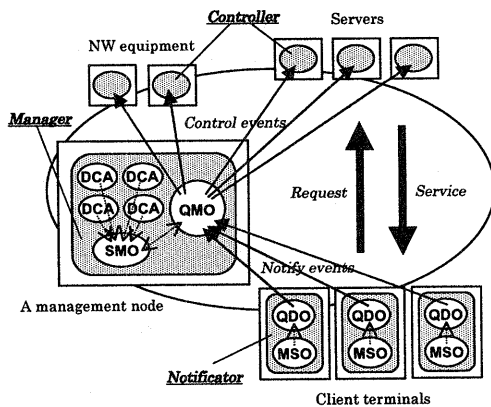


図1 システムアーキテクチャ

(1) Managerモジュール

アプリケーションQoS管理ポリシーに従って、制御方法を決定するモジュールである。運用対象システム毎に存在し、通常、システム管理ノード上に実装される。Managerモジュールは3種類のオブジェクトで構成される。各アプリケーションのQoS管理ポリシーを保持し、制御方法を決定するQoS管理オブジェクト(QoS Management Object: QMO)、各種性能データを収集するデータ収集エージェント(Data

Collection Agent: DCA)、収集されたデータをシステム状態として保持する状態管理オブジェクト(State Management Object: SMO)である。QMOはManagerモジュールに1つ存在し、運用対象である全アプリケーションのQoS管理ポリシーを保持する。

(2) Notificatorモジュール

アプリケーションQoSを測定、監視するモジュールで、通常クライアント端末上に実装される。Notificatorモジュールは、2種類のオブジェクトで構成される。監視対象アプリケーションQoSの測定を行うQoS測定オブジェクト(QoS Measurement Object: MSO)と、アプリケーションの管理ポリシーに基づき、QoSの劣化を検知するQoS監視オブジェクト(QoS Detection Object: QDO)である。これらのオブジェクトは、Notificatorモジュールに対して1つずつ存在し、利用者が当該サービスを受けている間中、QoSの監視を行う。

(3) Controllerモジュール

Managerモジュールで決定した制御を実現するモジュールで、制御実施対象毎に存在する。各アプリケーションの管理ポリシーによって、制御方法は異なる。

3. QoS管理ポリシーの設定

アプリケーション毎のQoS管理ポリシーは、Managerモジュールに設定される。このとき、QMSの管理スコープ内には複数のアプリケーションが実装されるため、これら複数アプリケーション間のQoSネゴシエーションが必要である。ここでは、このQoSネゴシエーションを実現し、システム全体として管理ポリシーに従った運用を可能とする、管理ポリシー設定手法を提案する。

3.1 基本方針

QMSにおけるQoS管理ポリシーの設定は、以下の2段階で行われる。

- ① アプリケーション毎にQoSを定義し、その制御アルゴリズムを決定する。
- ② アプリケーションのトラヒック特性を基に、

ネットワークレベルの制御の優先順位を決定する。

①では、サーバでのサービス内容の変更等、アプリケーションレベルの制御アルゴリズムを決定する。②では、複数アプリケーション間のQoSネゴシエーションを実現するため、ネットワークレベルの制御の優先順位を決定する。

3.2 制御対象

QMSにおけるQoS制御は、測定駆動型の適応的制御である。複雑なモデル化を伴わないフィードバック型制御を採用する代わりに、制御周期を短縮し精度の低下を補う。そのためQMSでの制御対象は、システム運用中に準リアルタイムで変更可能な要素となる。ここでは、ネットワーク上を流れるトラフィックの制御、サーバの負荷分散、サービス内容の変更を制御対象とする。サーバやクライアントのCPU、メモリ、ディスクアクセス等の変更は考慮しない。

(1) アプリケーションレベルの制御

アプリケーションレベルの制御対象は、監視対象のアプリケーションQoSに合わせて設定され、それに基づき制御内容が決定される。アプリケーションレベルの制御対象、制御例を表2に示す。

表2 アプリケーションレベルの制御

制御対象	制御例
サービスレベルの変更	映像送出レートの変更
ロードバランシング	接続先サーバの変更
エラー訂正	冗長パケットの利用
提供サービスの変更	映像を静止画に変更

(2) ネットワークレベルの制御

ネットワークレベルの制御対象は、インターネットにおけるエンド-エンドサービス品質である、スループット、IPデータグラムの遅延、遅延ゆらぎ、損失⁹⁾の4種類とする。これらの品質を確保するため、QMSでは表3に示すネットワークレベルの制御を行う。

各制御方法は、制御対象毎の品質劣化要因から決定した。つまり、スループットは、通信相手先に到達するまでの通信経路の違い、経路上のルー

タの性能とその数、ルータ間の回線速度とその混雑度合など、IPデータグラムの遅延および遅延ゆらぎは、ルーティングテーブル検索時間、回線の伝搬遅延など、IPデータグラムの損失は、ルータのバッファあふれなどに依存するとした。

表3 ネットワークレベルの制御

制御対象	制御方法
スループット確保	RSVP等を利用した帯域保証
遅延を防ぐ	バッファでの送出優先
遅延ゆらぎを防ぐ	バッファでのシェーピング
損失を防ぐ	DiffServ等を利用したパケット廃棄優先度の設定

3.3 アプリケーションの分類

QMSのネットワークレベルの制御は、アプリケーションのトラフィック特性に応じて、その優先順位を決定する。この特性は、トラフィックの種類と応答時間に対する要求によって分類される⁹⁾。アプリケーションから発生するトラフィックの種類を表4に、応答時間に対する要求種別を表5に示す。

ブロック型のアプリケーションに要求されるQoSは、静止画の表示にかかる時間やファイルのダウンロード時間などである。そのため、ネットワークレベルの品質尺度としてスループットを用いる。ストリーム型のアプリケーションに要求されるQoSは、画像の動きの滑らかさ、音声のとぎれなどである。そのため、ネットワークレベルの品質尺度として、遅延ゆらぎに対する要求が大きい。トランザクション型のアプリケーションに要求されるQoSは、メールの到達度やデータベースの応答時間などである。そのため、ネットワークレベルの品質尺度として、遅延、損失を用いる。一方、応答時間に対する要求については、リアルタイム性の高いトラフィックに対して、パケット送出順序制御は適さない。そのため、パケット廃棄処理を用いる。

これらのトラフィックの特徴から、表3に示した4種類の制御の優先順位を決定する。各アプリケーションを、この2種類の尺度によって分類し、

制御の優先度を決定することによって、アプリケーション間の QoS ネゴシエーションを実現する。

表 4 トラヒックの種類による分類

トラヒックの種類	特徴
ブロック型 (B)	大きなサイズの情報が低い頻度で発生 (静止画など)
ストリーム型 (S)	中サイズの情報が周期的に発生 (映像など)
トランザクション型 (T)	小サイズの情報が不規則に発生 (コマンドなど)

表 5 応答時間に対する要求による分類

応答時間への要求	サービス例
リアルタイム (RT)	数100ms以下が必要 (電話など)
準リアルタイム (QRT)	数秒以下が望ましい (WWW, 情報検索など)
非リアルタイム (NRT)	100秒以上も許容 (電子メールなど)

アプリケーションによっては、複数のトラヒック種別を持つものも存在する。例えば、オンラインショッピングでは、発注データ送信 (トランザクション型)、静止画による商品情報 (ブロック型)、インターネット電話による問い合わせ (ストリーム型) などのトラヒックが混在する。この場合、これら 3 種類のサービスは、別アプリケーションとして登録する必要がある。

3.4 管理ポリシー設定方法

アプリケーションレベルの制御法として、測定駆動型の簡易なアルゴリズムを適用した後、アプリケーションのトラヒック特性に従って、ネットワークレベルの制御の優先順位を決定する。

QMS における制御対象毎の優先順位を表 6～表 9 に示す。これらの優先順位は、3.3 節で述べたトラヒック特性ごとの品質尺度を基に決定した、デフォルトの順位である。そのため、利用環境、利用者の状況に応じて、順位の変更を行う必要がある。

表 6 帯域制御の優先順位

優先順位	種類	応答時間	制御の有無
0	S	RT	○
1	B	RT	○
2	S	QRT	○
3	B	QRT	○
4	T	RT	×
5	T	QRT	×
6	S	NRT	×
7	B	NRT	×
8	T	NRT	×

表 7 パケット送出の優先順位

優先順位	種類	応答時間	制御の有無
0	T	RT	--
1	S	RT	--
2	B	RT	--
3	T	QRT	--
4	S	QRT	--
5	B	QRT	--
6	T	NRT	--
7	S	NRT	--
8	B	NRT	--

表 8 シェーピングの優先順位

優先順位	種類	応答時間	制御の有無
0	S	RT	○
1	S	QRT	○
2	B	RT	○
3	S	NRT	×
4	B	QRT	×
5	T	RT	×
6	B	NRT	×
7	T	QRT	×
8	T	NRT	×

表 9 パケット廃棄の優先順位

優先順位	種類	応答時間	制御の有無
0	S	RT	--
1	B	RT	--
2	S	QRT	--
3	B	QRT	--
4	S	NRT	--
5	B	NRT	--
6	T	RT	--
7	T	QRT	--
8	T	NRT	--

4. 実装例

QoS 管理ポリシー設定機能を Java アプリケーションにより実装した。実装例を図 2～図 4 に示す。新しいアプリケーションを追加する場合、新規アプリケーション追加画面において、アプリケ

ーション名,トラフィックの種類,要求される応答時間を設定する.すると,3章で規定した優先順位決定ルールに従い,ネットワークレベルの各制御の優先順位が決定される.この優先順位は,4種類の優先度設定画面において確認可能であり,必要に応じて順位の変更を行う.ここで決定された優先度は,各ネットワーク機器に設定され,ネットワークレベルの制御が実現する.

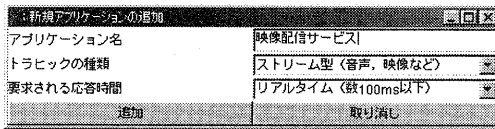


図2 新規アプリケーション追加画面

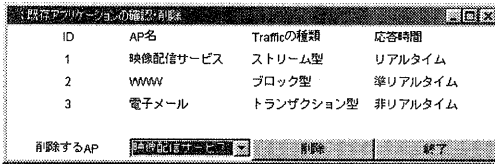


図3 既存アプリケーション確認画面

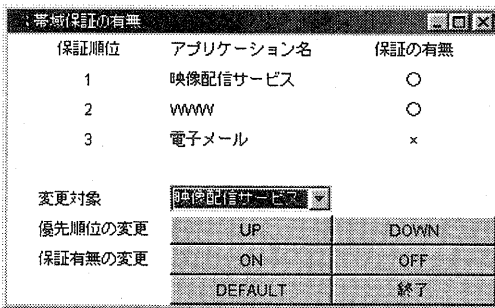


図4 帯域保証優先度の設定画面

本システムを用いると,アプリケーションのトラフィック特性を選択することにより,容易にQoS管理ポリシーが設定できる.また,4種類の制御の優先度を決定することにより,既存のアプリケーションとのQoSネゴシエーションが可能となる.

今回の実装例は,シミュレーション実験用のモジュールとして開発した.このモジュールに,ネットワーク機器への制御内容設定機能を追加することにより,実環境への適用が可能となる.

5. まとめ

分散システムにおけるアプリケーション QoSに着目し,ポリシーに従ったシステム運用を可能とする,QoS管理ポリシー設定手法を提案した.本手法は,アプリケーションレベルの制御ルールとネットワークレベルの制御ルールを組み合わせることにより,複数アプリケーション間のQoSネゴシエーションを実現する.その結果,システム全体として,管理ポリシーに従ったアプリケーション運用が可能となる.

今後,シミュレーション実験によるアプリケーション間QoSネゴシエーションの評価,実環境への本システムの適用実験を行う予定である.

参考文献

- 1) Object Management Group, "The Common Object Request Broker: Architecture and Specification", OMG, 1998.
- 2) RFC2205, "Resource reSerVation Protocol (RSVP)", 1997.
- 3) RFC2475, "Architecture for Differentiated Services (DiffServ)", 1998.
- 4) J.M. Smith, et al. "Activating Networks: A Progress Report", IEEE Computer, Vol.32, No.4, pp.32-41, 1999.
- 5) FloodGate-1 (<http://www.checkpoint.com>)
- 6) K. Nahrstedt, J.M. Smith, "The QoS broker", IEEE MultiMedia, Vol.2, No.1, pp.53-67, 1995.
- 7) 橋本, 野村, 柴田, 白鳥, "QoS保証を考慮したやわらかいマルチメディアシステム", 情処学会論文誌, Vol.40, No.1, pp.113-123, 1999.
- 8) 加藤, 金井, 中西, 箱崎, "分散システムにおけるアプリケーション管理フレームワークの提案", 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, Vol.99, No.3, pp.81-86, 1999.
- 9) 間瀬 編著, "マルチメディアネットワークとコミュニケーション品質", (社)電子情報通信学会編, コロナ社, 1998.