

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

加藤 由花 金井 秀明 中西 泰人 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

本論文では、分散システムにおけるアプリケーションのサービス品質に着目し、ユーザ指向型のシステム運用を可能とするシステム構成、およびシステム運用手法を提案する。本フレームワークは、ユーザに対して提供されるサービス品質の劣化を検知する *Notificator*、環境変化に適応した制御方法を決定する *Manager*、制御を実現する *Controller* の3モジュールで構成され、適応的な品質制御を実現する。この管理フレームワークの適用例として、Campus LAN 上で映像配信サービスの制御を行うプロトタイプシステムを開発した。このプロトタイプを用いて、管理フレームワークの実装検証と、実装性能評価を行った。

A Proposal of an Application Management Framework for distributed Computing Systems

Yuka KATO, Hideaki KANAI, Yasuto NAKANISHI, and Katsuya HAKOZAKI

Graduate School of Information Systems, University of Electro-Communications

Abstract

This paper proposes an application management framework for user oriented system management from a viewpoint of application quality of service in distributed computing systems. This framework consists of three modules, which are *Notificator* detecting deterioration in quality of service to provide for system users, *Manager* deciding on a control method adapting flexibly to a system environment and *Controller*. These modules realize adaptive control of performance. As an application of this framework, we develop a prototype for video service in Campus LAN. By using this prototype, we examine system implementation and evaluate system performance.

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、ネットワーク上には多くの分散システムが導入されるようになってきた。それらのシステムは業務の基幹を担うものとなり、アプリケーションレベルでの QoS (Quality of Service) 管理の重要度は年々増加してきている。例えば、プロトコルやホスト単位でパケットの流量制御を行う Packet Shaper [2]、Web サーバの負荷分散処理を行う Local Director [3] 等が製品化されている。また、インターネット上で帯域予約を行う

RSVP (Resource reSerVation Protocol) [4] を実装したネットワーク機器も登場してきた。

これらの技術は、専用のハードウェアが必要であったり、アプリケーションレベルの制御を、ネットワークレベルで実現しようとするものであり、新しいアプリケーションが開発されるたびに、ネットワーク管理ポリシーを変更する必要がある。また管理対象となる QoS は、ユーザが体感するサービスレベルの QoS (システム応答時間、画像のみだれなど) ではなく、ネットワークレベルでの QoS (パケット損失率、遅延など) であることが多い。技術レベルが十分でなかった時代には、このようにネットワークでのサービス品質が

ら、ユーザに提供できるサービス品質を決定しており、ユーザに技術的限界を許容してもらう必要があった [5]。しかし、技術レベルが十分成熟した現在では、技術主導型サービスからユーザ指向型サービスへのパラダイムシフトが必要である。

このような観点から、アプリケーションにシステムリソースを割り当てることによってアプリケーションレベルの QoS を制御する研究が行われている [6][7]。これらの研究は、リソースの割り当てに市場モデル等、複雑なアルゴリズムを用いており、実用レベルの研究にはなっていない。また Video on Demand 等、特定のアプリケーションのサービス品質を制御する研究も行われているが [8]、これらは対象とするアプリケーションの制御手法に着眼したもので、分散アプリケーションの管理に着目しているわけではない。

本論文では、分散システムにおけるアプリケーションのサービス品質 (以下、アプリケーション QoS と記す) に着目し、ユーザ指向型のシステム運用を可能とするシステム構成、およびシステム運用手法を提案する。このフレームワークは、ユーザに対して提供されるサービス品質の劣化を検知する Notificator、環境変化に適応した制御方法を決定する Manager、実際に制御を行う Controller の 3 モジュールで構成され、適応的な品質制御を実現する。今回、管理フレームワークの適用例として、Campus LAN 上で映像配信サービスの制御を行うプロトタイプシステムを開発し、フレームワークの実装検証と実装性能評価を行った。以下、2 章ではフレームワークの構成について、3 章ではプロトタイプシステムについて、4 章ではその性能評価結果について述べる。

2 アプリケーション管理フレームワークの構成

2.1 コンセプト

アプリケーション管理フレームワークの構築における設計コンセプトを以下に示す。

- (1) ネットワークトポロジやトラフィック等、システム環境の変化に柔軟に適応できる、動的運用手法を採用する。
- (2) ユーザ指向型の管理を目指し、ネットワークレベルの QoS ではなく、アプリケーション QoS を制御する。
- (3) 複雑なアルゴリズムを用いた精度の高い制御ではなく、簡易な制御を短周期で行う。

2.2 システム構成

上記のコンセプトをもとに、Manager、Notificator、Controller の 3 種類のモジュールで構成される、分散アプリケーション管理フレームワークを設計した。ここでは、環境変化に柔軟に適応できる動的運用を目指しているため、周期的にサービス品質を測定し、適宜システム制御を行う必要がある。そのため、Manager は常時、各種性能データを収集する。一方、性能評価はユーザに提供されるサービス品質で行うため、性能劣化の検出はクライアントに実装される Notificator が行う。Notificator から品質劣化イベントを受け取った Manager は、収集している各種データ、およびアプリケーション毎に定義された管理ポリシーに基づき、システム制御方法を決定する。実際の制御は、Manager からの要求を受けた Controller が、提供するサービス内容を変更することによって行う。

一般に、同一ネットワーク上には複数の分散アプリケーションが運用されているが、各アプリケーション毎に Manager を用意することで、アプリケーションや運用環境毎の管理ポリシーを設定することができる。Manager の実装を工夫することによって、複数アプリケーション間のネゴシエーションや優先順位の設定も可能になる。

各々のモジュールはネットワーク上に分散配置されるが、モジュール間の通信に CORBA 等の分散オブジェクト環境を利用することにより、実装の容易性と拡張性に対応する。

システム構成のイメージを図 1 に示す。

(PentiumMMX166MHz Windows98) 上に実装した。PC3はダイヤルアップ端末である。

2つの映像サーバ上にそれぞれ、異なるレートでエンコードされた4種類の映像ファイル(28kbps、56kbps、100kbps、200kbps)を用意し、そのときの状況に適した映像が配信される。

4.2 実装性能評価

実装性能として、システムのターンアラウンドタイムを測定した。結果を図5に示す。

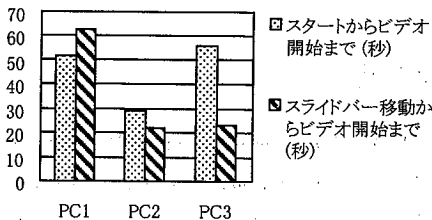


図5: ターンアラウンドタイム

処理時間はクライアントの性能とネットワーク条件により大きく異っている。そこで処理のボトルネック検出のため、スタートからビデオ開始までの時間の内訳を測定した。結果を図6に示す。

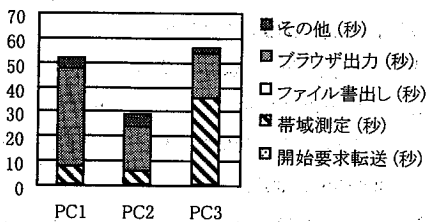


図6: ターンアラウンドタイムの内訳

図6から、ブラウザ出力と帯域測定が処理時間の大部分を占めていることがわかる。ブラウザ出力時間には、plug-inの読み込み時間、ビデオバッファリング時間等が含まれ、クライアント性能に大きく依存する。またダイヤルアップ端末であるPC3では、帯域測定に非常に時間がかかっている。スライドバー移動からビデオ開始までの時

間短いのは、帯域測定が行われないためである。本プロトタイプでは、帯域測定ツールにpathcharコマンド[11]を利用したが、ツール適用範囲の確認、および測定ツールの見直しが必要であろう。

これらの処理時間を除いた管理システムとしての処理時間は、制御周期30秒に対して十分短く、十分な実装性能といえる。

4.3 アルゴリズム性能評価

アルゴリズム性能を評価するには、様々な実験条件で複数アルゴリズムを実装し、比較を行う必要がある。今回の実験はプロトタイプによる実装検証であること、1種類のアルゴリズムだけを実装したことから、本来の意味でのアルゴリズム性能評価は行えない。そこで、アルゴリズムがシナリオに従って正しく動作するかを確認することにした。以下の3種類のシナリオを用意した。

- (1) WS1に負荷をかける。空き帯域はWS1の方が大きいとき、WS2上の映像が選択されるか。
- (2) ネットワーク条件の差によって、適した映像ファイルが選択されるか。
- (3) 視聴中にFTPを行うと、帯域は低下するか。

シナリオ(1)では、高負荷のWS1は選択されず、空き帯域が少ないWS2が選択された。結果を表1に示す。

表1: アルゴリズム性能評価結果(1)

CPU_idle	WS1	16%
	WS2	95%
空き帯域	WS1	36.9kbps
	WS2	36.6kbps
選択結果	サーバ	WS2
	ファイル	28kbps

シナリオ(2)では、LAN上の端末(PC1)では最も広帯域な200kbpsの映像が選択され、ダイヤルアップ端末(PC3)では28kbpsの映像が選択された。結果を図7に示す。

シナリオ(3)では、FTP前に十分な空き帯域が確保できなかったため、制御ファイルを強制的に書き換え(制御なしの状態)、56kbpsの映像を配

信している状態で FTP を行った。すると空き帯域が低下し、次周期には 28kbps の映像が選択された。結果を図 8 に示す。これは映像サービスより他のサービスの優先度が高いという設計ポリシーと一致する。

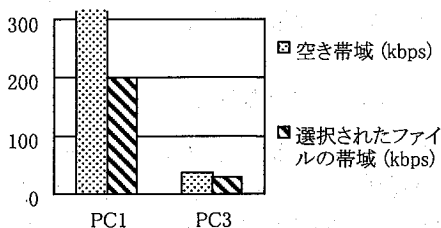


図 7: アルゴリズム性能評価結果(2)

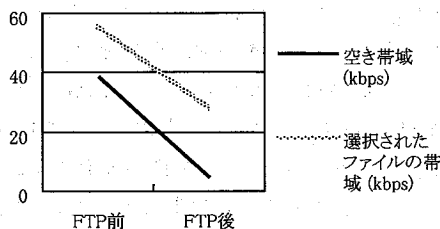


図 8: アルゴリズム性能評価結果(3)

以上の結果より、本アルゴリズムはシナリオに従って正しく動作することが確認された。今回採用したアルゴリズムは、複雑なモデル化をいっさい行っておらず、本フレームワークに適したアルゴリズムと言える。

5 まとめ

本論分では、分散システムのアプリケーション品質に着目し、ユーザ指向型のシステム運用を可能とするシステム構成、およびシステム運用手法を提案した。このアプリケーション管理フレームワークでは、管理のロール毎に 3 種類のモジュールを定義することにより、サーバおよびクライアントの実装をモジュール内に隠蔽し、アプリケーションや運用環境に適した管理ポリシーを設定

することが可能である。

管理フレームワークの適用例として、Campus LAN を対象とした映像配信サービスの QoS 制御を行う、管理システムのプロトタイプを開発した。このプロトタイプによって、管理フレームワークの実装性能およびアプリケーション性能の評価を行い、十分な性能を確認した。

今後、より広域なネットワークへの本フレームワークの適用、シミュレーション実験による様々なアルゴリズムの評価、映像配信サービス以外への適用について研究を進める予定である。

参考文献

- [1] OMG, "The Common Object Request Broker: Architecture and Specification.", OMG, 1998.
- [2] Packet Shaper (<http://www.packeteer.com>)
- [3] Local Director (<http://www.cisco.com>)
- [4] RFC2205, "Resource reSerVation Protocol"
- [5] 浅谷, "新しい通信品質の考え方—マルチメディアと GII 時代へ向けて—", 信学会誌, Vol.81, No.9, pp.924-931, 1998.
- [6] 八旗, ウェルマン, 石田, "市場モデルに基づくアプリケーション QoS の制御", 信学論 D-I, Vol.J81-D-I, No.5, pp.540-547, 1998.
- [7] 小菅, 酒井, 松田, "適応型情報通信アプリケーションのためのフレームワークの一考察—エージェントに基づくシステムアーキテクチャー—", 信学技報, CQ96-66, 1997.
- [8] J-C. Bolot, T. Turletti, "Experience with control Mechanisms for Packet Video in the Internet", ACM SIGCOMM, Vol.28, No.1, 1998.
- [9] BasicServer (<http://www.real.com/Solutions>)
- [10] S. Hirano, "HORB: a distributed object oriented language for worldwide programming", WOO'96, 1996.
- [11] Pathchar: Van Jacobson (<ftp://ftp.ee.lbl.gov>)