

5ZA-03 動的 QoS 制御に基づく資源管理システム『堤』の設計と実装

田中 大介[†] 芝 公仁[†] 大久保 英嗣^{††}

[†]立命館大学大学院理工学研究科 ^{††}立命館大学工学部情報学科

1 はじめに

ここ数年、計算機の高性能化やネットワークの広帯域化に伴い、電子会議システムや VOD といったマルチメディアアプリケーションが実用化されるようになってきた。また、RT-Mach, RT-Linux といったリアルタイム OS の出現、Diffserv や IEEE802.1p といったネットワークの帯域制御技術の出現にみられるマルチメディア処理に関する技術に大きな関心が集まっている。既存のシステムでは、アプリケーションの優先度を考慮したサービスの品質 (QoS: Quality of Service) を保証することが難しい。これは、個々のアプリケーションが、優先度、システム負荷、ネットワークの利用状況を考慮せず資源を要求するためである。

そこで、我々は、マルチメディアアプリケーションに有効な動的 QoS 制御に基づく資源管理システム『堤』の設計と実装を行っている。『堤』は、メディアスケーリング、QoS 交渉、資源予約の 3 つの概念を基本として、CPU、メモリ、ネットワーク帯域の計算機資源をシステム負荷に関わらずアプリケーションに提供する。

2 『堤』の概要

2.1 『堤』の目的

『堤』は、単一ノードのアプリケーションに対し資源管理を行うだけでなく、他ノードの『堤』と独自の資源管理プロトコルを利用して分散環境での資源管理を行う。このような資源管理は、電子会議システムといった分散マルチメディアアプリケーションに適用可能である。

2.2 『堤』の特徴

『堤』は、マルチメディアアプリケーションを支援する資源管理システムである。マルチメディアアプリケーションは一般に連続メディアデータを扱うが、連続メディアデータには以下の 2 つの特徴がある。

- 離散メディアのように値の正当性だけでなく時間の正当性が重要な要素となる。
- 意図的にデータの QoS を調節することができる。

The Design and Implementation of Resource Management System "Tsumi" based on Dynamic QoS Control
Daisuke TANAKA[†], Masahito SHIBA[†], and Eiji OKUBO^{††}
[†]Graduate School of Science and Engineering, Ritsumeikan University
^{††}Department of Computer Science, Faculty of Science and Engineering, Ritsumeikan University

我々は、メディアデータの持つ特徴を基に以下の 3 つの概念を『堤』に導入している。

● メディアスケーリング (Media Scaling)

ユーザがあらかじめ決めた優先度が低い連続メディアデータの品質を低下させ、システムに余計な負荷をかけないようにする。また、優先度の高いデータの品質を上げることで、ユーザの要求する QoS を保証できるようにする。

● QoS 交渉

ユーザの要求する連続メディアデータの品質 (以下、ユーザ QoS) を実際に利用する資源量 (以下、システム QoS) に相互変換し、優先度に基づいた交渉で資源配分を行う。ユーザ QoS は、データに対する数段階の優先度であり、ユーザがアプリケーションを介して決定する値である。このユーザ QoS をテーブルを使用してデータのフレームレートや解像度といった属性 (以下、アプリケーション QoS) に変換し、システム QoS を求める。データが VBR (Variable Bit Rate) の場合、経験的もしくは悲観的なアドミッション制御でシステム QoS を予測する。

● 資源予約

連続メディアデータの持つ実時間制約を満たすために、前もって資源 (CPU、メモリ、帯域幅) を予約する。資源予約は、メディアスケーリングと QoS 交渉の後に行うべきであり、実際に資源が使われるまで資源を確保する。それぞれの資源は、互いに独立せず何らかの関連を持って予約されることがある。例えば、ネットワーク間でデータトラフィックを転送する場合、十分な帯域とキャッシングに使用されるメモリをどちらも予約しなければならないといったことが挙げられる。このため『堤』では、総合的に資源を管理する機構を提供する。

3 『堤』の構成

3.1 構成要素

『堤』は、資源管理を行うミドルウェアである『堤』本体と、アプリケーションに提供される API 群である堤ライブラリの 2 つから構成されている (図 1 参照)。

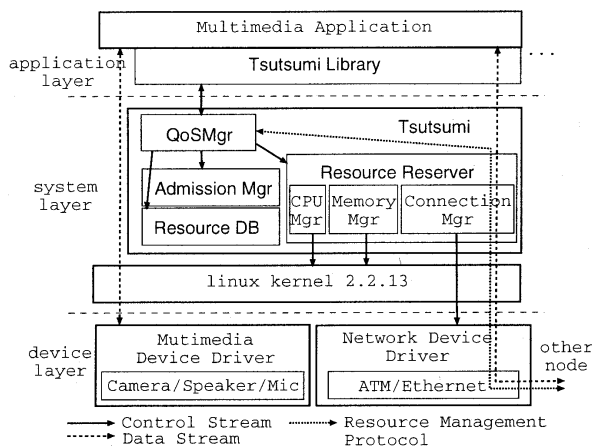


図1: 『堤』の構成

- QoS マネージャ (QoS Manager)**
 アプリケーション QoS とシステム QoS の相互変換, システム負荷状況の把握, 優先度に基づく QoS 交渉を行う。さらに, 自ノードの QoS 変更に伴い他ノードの QoS に影響が及ぶ場合, QoS マネージャがコネクションマネージャを利用して他ノードと協調した資源管理を行う。
- アドミッションマネージャ (Admission Manager)**
 資源の残量管理, および要求された資源が提供可能か否かを判定する。また, データが VBR である場合のシステム負荷の予測を行う。
- リソースリザーバ (Resource Reserver)**
 『堤』の資源確保機構である。リソースリザーバは, それぞれの資源を確保する CPU マネージャ, メモリマネージャ, コネクションマネージャを統括し自ノードの資源を確保する。アプリケーションが他ノードの計算機資源を要求する場合, コネクションマネージャが資源管理プロトコルを利用して他ノードと通信する。
- リソースデータベース (Resource DB)**
 ユーザ QoS, アプリケーション QoS, システム QoS, アプリケーション pid や利用者の uid, メディアデータの優先度といった資源管理に必要な情報のすべてを蓄積するデータベースである。ここにはデータベース操作に必要な関数群も含まれる。
- 堤ライブラリ (Tsutsumi Library)**
 『堤』が提供する静的ライブラリである。メディアスケール, 『堤』との通信, メディアデータの解析といった機能を提供する。このライブラリをリンクすることで, アプリケーションは『堤』の機能を利用することができる。

3.2 処理の流れ

アプリケーションが資源を要求し, 資源が確保されるまでの処理の流れを以下に示す。

1. アプリケーションが『堤』に login し, ユーザからユーザ QoS を受け取る。
2. 堤ライブラリがメディアデータの解析や QoS の変換を行い, QoS マネージャにアプリケーション QoS を通知する。
3. QoS マネージャが, アプリケーション QoS を受け取り, アドミッションマネージャによって現在や将来のシステム負荷を予測する。
4. QoS マネージャは, 予測結果から各ストリームの QoS 交渉を行う。
5. 交渉結果や QoS 情報をリソースデータベースに保存する。
6. リソースリザーバが交渉結果から資源を確保し, 利用するのに必要な情報を QoS マネージャに送信する。
7. 堤ライブラリが QoS マネージャから結果を受け取り, メディアスケールや資源を利用できるようにする。

3.3 資源確保方法

『堤』のリソースリザーバにおける資源の確保方法を述べる。CPU 資源は, メディアデータの再生を行うタスクをリアルタイムタスクとして扱い, EDF 法や RM 法といったリアルタイムスケジューリングを行うことで確保する。このとき, メディアデータのサンプリングレートやフレームレートの周期を求めてスケジューリングすることになる。

メモリ資源は, アプリケーションと『堤』との共有メモリを作成し, 一貫性制御およびアクセス制御を行い, 提供できる資源として確保する。このメモリ資源は, アプリケーションがメディアデータを再生するためのキャッシュとして利用される。

帯域資源を確保するためには, いくつかの方法が考えられる。ATM スイッチを使った帯域制御を利用する方法, イーサネットでは, パケットに優先度を付加するといった VLAN 技術, TCP/IP のウィンドウフレームを制御し無駄なトラフィックを減少させる方法がある。これらの手法で確保した帯域資源は, アプリケーションがデータトラフィックを転送するために使われる。

4 おわりに

本稿では, 動的 QoS 制御に基づく資源管理システム『堤』の設計について述べた。现阶段の『堤』では, ユーザがすべてのアプリケーションやデータの優先度を決め, レベルに応じた QoS 制御を実現している。今後は, アプリケーションと協調できるエージェントによって優先度をある程度予測し, ユーザの負担を減らすことを予定している。また, 分散した環境での有効な資源管理手法や交渉が『堤』の課題である。