

デコーダダウンロードダブルシステムにおけるスケジューリング方式の検討

A study on scheduling method of decoder downloadable system

宮澤 敏記^{*1}Toshinori MIYAZAWA^{*1}清水 直人^{*1}Naoto SHIMIZU^{*1}亀山 渉^{*2}Wataru KAMEYAMA^{*2}渡辺 裕^{*2}Hiroshi WATANABE^{*2}富永 英義^{*1*2}Hideyoshi TOMINAGA^{*1*2}^{*1} 早稲田大学 理工学部 電子・情報通信学科^{*1} Dept. of Elec. Info. and Comm. Eng., WASEDA Univ.^{*2} 早稲田大学国際情報通信研究センター^{*2} Global Info. and Tele. Institute, WASEDA Univ.

1. はじめに

ソフトウェアデコーダを配信することにより、様々なメディアフォーマット^[1]が復号可能となる。しかし、異なるメディアを連続再生する過程で、デコーダを動的にダウンロードするシステムは存在しない。そのため我々は、各コンテンツに最適なデコーダを動的にダウンロードするシステムを提案している^[2]。本稿では、デコーダダウンロードのスケジューリングと再生遅延時間に関する評価を行い、結果からユーザーの特性を基としたスケジューリング方式を提案する。

2. デコーダダウンロードダブルシステム概要

クライアントが各コンテンツに最適なデコーダを動的に見つけ、かつダウンロードする処理の流れは次のようになる。1) クライアントシステム内のスケジューラは、適切なタイミングでユーザーが指定したコンテンツ情報を解析する。2) 対象のデコーダが存在しない場合、コンテンツ情報をサーバに渡す。3) サーバは受信したコンテンツ情報を元に、クライアントに最適なデコーダ情報を検索しクライアントに送信する。4) デコーダ情報には、デコーダファイル名やデコーダ配信サーバアドレスが含まれており、これらの情報を元にクライアントはデコーダ配信サーバからデコーダを受信する。

3. スケジューリング評価

デコーダダウンロードのスケジューリングを評価するために、1) 蓄積するデコーダ数とメディア再生までの初期遅延時間、2) 蓄積するデコーダ数と総遅延時間に関して実験を行った。Pull型サービスにおいて、ユーザーは選択したコンテンツ間で恣意的にJumpすると考えられ、ユーザーの操作を考慮したコスト換算が必要となる。(実行環境-CPU:Pentium III 1GHz, RAM:256MB, OS:windows 2000, Java環境:JDK1.3.1)。

3.1 メディア再生初期遅延の評価

コンテンツ数を12、全てのコンテンツが4.25KBのエフェクト処理をダウンロードする必要があると設定し、一番目のメディア再生までの遅延時間を測定した(図1)。グラフの横軸に蓄積されるデコーダ数を取る。スケジューリング方式は、1) 既にクライアントにデコーダが存在する場合(local decoder)、2) 複数のデコーダを並列にダウンロードする方式(parallel download)、3) 複数のデコーダを再生予定順番に従って直列にダウンロードする方式(serial download)について比較した(対象デコーダダウンロード完了直後、メディア再生プロセス再開)。

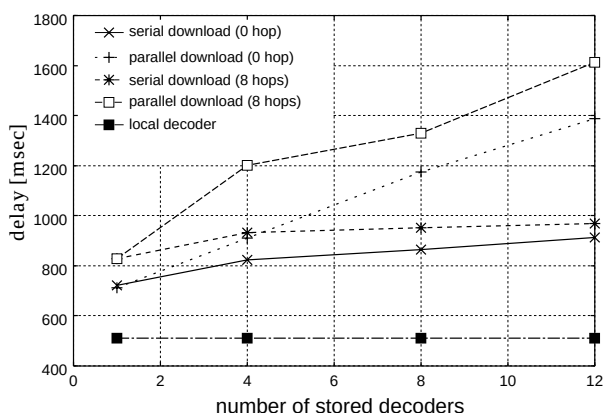


図 1: 蓄積デコーダ数とメディア再生初期遅延の関係

結果から複数のデコーダを一括してダウンロードする際、必要なデコーダを優先的にダウンロードする直列方式の方が並列方式よりも再生までの遅延が小さいことが判明した。また直列方式であっても、ダウンロードするデコーダ数が増えるにつれて遅延時間が増加するのは、ftpセッション処理がメディア再生処理のパフォーマンス効率を低下させるためと考えられる。

3.2 総再生遅延コスト評価

ダウンロードに伴う総遅延コストを $T_d(n)$ 、前節で求めた初期遅延を $d_s(n)$ 、1デコーダダウンロードに伴う遅延を d_d 、コンテンツ変更が有った際、新たなデコーダを必要とする確率を $p(k)$ 、恣意的にユーザーがコンテンツ変更する確率を $q(k)$ 、総コンテンツ数を N 、デコーダを必要とするコンテンツ数を M 、蓄積するデコーダ数を n とおけば、 $T_d(n)$ は次式で表すことができる(ただし、 $p(k) = \frac{M-n-k+1}{N-k}$ である)。

$$T_d(n) = d_s(n) + d_d \sum_{k=1}^{M-n} p(k)q(k)$$

$q(k)$ はある分布に従うものと想定されるが、本評価では $q(k)$ の値を1, 0.5, 0.25, 0に設定し、またその他の値は前節の評価で用いた値に従った。総遅延コスト T_d と蓄積デコーダ数 n の関係を図2に示す。結果よりユーザーによるコンテンツ変更が多い(少ない)程、蓄積デコーダ数が多い(少ない)方が総遅延が少ないことが分かる。ただし、ユーザーによる変更の確率が0.25である場合、デコーダ蓄積数が8で総遅延時間が最小となる。

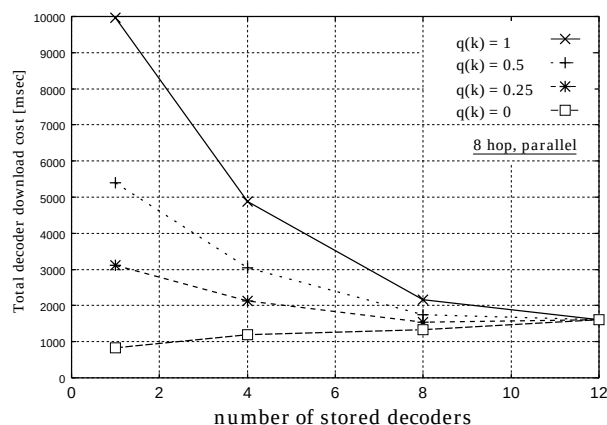


図 2: 蓄積デコーダ数と総遅延コストの関係

4. ユーザプロファイルに基づいたスケジューリング方式

総再生遅延コスト評価の前提として、ユーザーがコンテンツを変更する確率 $q(k)$ を定数として扱った。現実のモデルではある確率分布に従うと考えられるが、導出することは困難である。そこで我々は、ユーザーがコンテンツを変更する確率を統計的に求める方式を提案する。統計的に求めたユーザープロファイルに元、蓄積するデコーダ数を決定することで、総遅延時間が確率的に最小となり得るスケジューリング方式を導入することが可能となる。

5. まとめ

本稿では、現在実装中のデコーダダウンロードダブルシステムを基に、デコーダダウンロードのスケジューリングに関して評価を行った。本評価により導かれたことは、1) 蓄積デコーダ数が小さく、優先的にデコーダをダウンロードする手法が最も初期遅延時間が小さいという点、及び2) ユーザーによるコンテンツ変更確率による蓄積デコーダ数と総遅延時間の関係である。この結果から、我々はユーザープロファイルを統計的に求めることで、総遅延時間を確率的に最小とし得るスケジューリング方式を提案した。

参考文献

- [1] 宮澤, 亀山, 渡辺, 阪谷, 富永, "アニメーション画像符号化における輪郭線抽出と近似方式の検討", AVM32-12, 2001, pp.65-70.
- [2] 宮澤, 清水, 亀山, 渡辺, 富永, "Pull型サービスにおけるデコーダダウンロードダブルシステムの構成及び評価", PCSJ2001, 2001, pp.27-28.