

マルチメディア情報ネットワークのためのパケットビデオシステムの設計と性能評価

4V-5

清水省悟 瀬田直也 神原久夫 柴田義孝

東洋大学工学部情報工学科

1 はじめに

近年情報システムの発達により、画像や音声といった異なるメディアの情報を意味的に結合された情報としてユーザに提供するために、マルチメディア情報ネットワークを基本とするサービスの必要性が増加してきた。そこで筆者らは先に、パケットビデオシステムとして、マルチメディア情報ネットワーク上で効果的なビデオ通信(音響データと動画データ間で同期を取りながらの通信)を実現するための通信プロトコルと、そのプロトコルに適した動画及び音響処理方式、同期処理方式の設計及び実装を行ってきた[1]。本稿では、そのパケットビデオシステムの異なる実装手段(UNIXのプロセス及び、Machのタスク・スレッド)による性能の比較評価について報告する。

2 パケットビデオシステム

本システムは図1のような階層構造をしており、各層ごとに最適化を行なうことにより効率の良いパケットビデオシステムを構成している。Synchronization層では、動画フレームと音響セグメント間の同期処理を行ない、Strict, Relaxed, Silenceの3種類の同期処理を選択的に利用できる。Transform層では、各データの圧縮伸張及びフォーマット変換などを行なう。音響処理としてサンプリングレート・量子化ビット数・変調方式などのフォーマット変換機能や、有音データのみ転送処理を行なう無音検出機能をもつ。Transport Interface層では、サーバとクライアント間のパケット転送の制御を行ない、バッファの溢れを監視しながらパケットのフローを調整する。

2.1 同期方式

本システムでは、動画フレームと音響セグメント間の同期方式として、同期ポイントの決め方によって

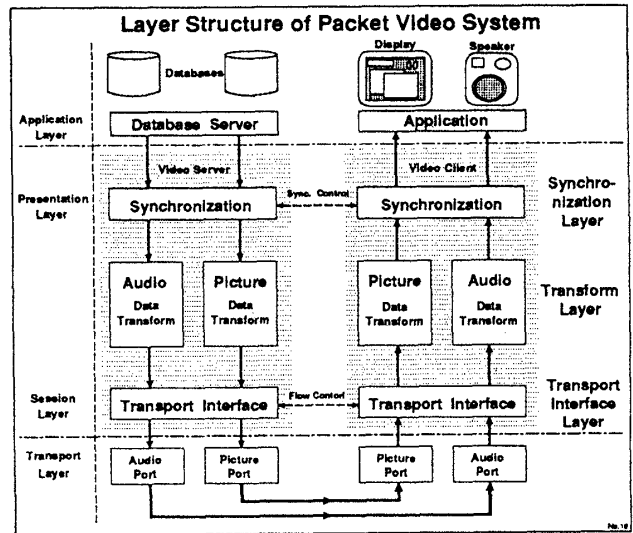


図1: パケットビデオシステムの階層構造

以下の3種類の方法[2]を選択できる。

Strict Synchronization 動画データフレームレートと音響データのセグメントレートを完全に一致させ、対応した動画1フレームと音響1セグメントの表示再生の開始時刻を揃える方式。

Relaxed Synchronization フレーム及びセグメントの合計時間が一致した時点で同期をとる方式。レート値がそれぞれ N_P [frames/sec], N_A [segments/sec] の動画及び音響データの時、 N_P, N_A の公約数でそれぞれのレート値を割った間隔で同期を取る。

Silence Synchronization 音響セグメントが無音であるか有音であるかを判定し、有音状態の場合のみ先の2つの同期方式のいずれかを用いて動画データと同期を取るか、又は有音部分の開始点だけ揃える方式。無音状態の場合は、動画を一定レートで表示するだけで、音響との間で同期処理を必要としない。

3 システムの実装

本システムは、メディアデータの転送を高スループットに維持しながら、正確な同期処理を行い、各層の機能モジュールを同時に処理するために、UNIXのプロセスとMachのタスク・スレッドによる実装を行い、その同期精度の比較を行った。図2にクライアント側の実装を示す。UNIXシステムでは、各層の機能モ

Implementation and Performance Measurements of a Packet Video System for Multimedia Information Network

Shogo Shimizu, Naoya Seta, Hisao Kamihara, Yoshitaka Shibata

Toyo University

ジュールそれぞれにプロセスを割当て、メディアデータを格納するバッファを共有メモリで実現し、データ処理手順を直線化するために共有メモリに確保されたフラグを常に監視している。Mach システムでは、システム全体で一つのタスクを生成し、各層の機能モジュールそれぞれにスレッドを割当てることにより、スレッド群はタスクの資源として確保されたメディアデータ用のバッファを共有し、データ処理手順を直線化するためにスレッド用のシグナルを使用している。また、サーバー側でも同様な実装を行っている。

Mach による実装では、実行単位がスレッドであるため、各層の機能モジュールの制御が容易になり、しかも機能モジュール間でのデータコピーが必要ない。また、スレッドのコンテキスト切替えはプロセスの場合と比べ処理が軽く、マルチCPUで実行する場合、機能モジュールが負荷分散され物理的に並列処理される。

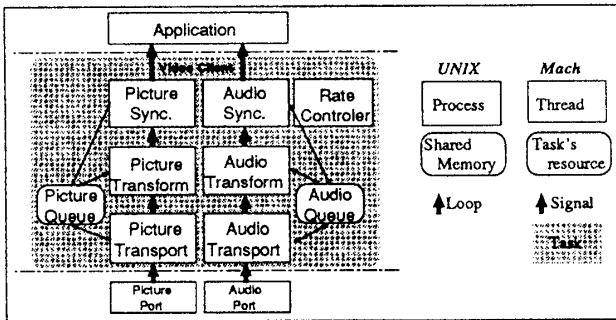


図 2: システムの実装方法

4 評価

プロトタイプとして2つのシステムを使用した。

機種	SPARC Station2	OMRON LUNA88k
性能	28[MIPS]	25[MIPS]
OS	UNIX	Mach2.5
Network	Ethernet(10Mbps)	
CPU数	1	1,2

ビデオデータとして次のようなデータを使用した。

FrameSize[pixel]	80x60, 120x90, 160x120, 240x180
FrameRate[fps]	10, 16, 24, 30
ColorDepth	8[bits/pixel]
Bandwidth	8000[Hz]
QuantizeBit	8[bits]
Modulation	PCM, μ -law

評価は同期精度を明確にする為に、サーバーでデータをディスクから読み、クライアントで同期処理を行うまでとし、動画の表示及び音響データの再生は実際には行なわなかった。

この結果、Strict Sync. の場合では図3より、UNIX、

Mach 両システムとも同期の時間誤差を1フレーム間隔以内に抑える事ができ、フレームのサイズやレートに関わらず同期の精度は高かった。Silence Sync. の場合では図4より、無音から有音になる点での同期は正確に行なえたが、有音状態が続いた場合の同期の時間誤差の増加は、Mach システムの方がUNIX システムに比べて小さかった。

5 まとめ

UNIX 及び Mach システムを比較した結果、相互作用する平行処理のアプリケーションを開発する際、プロセスよりもタスク・スレッドを使用の方が能率的であった。またプロトタイプでの評価を行なった結果、どちらの場合でも、ほぼ満足のいくレベルで動画像と音響データを同期を取りながら転送可能であることが確認できた。しかし非同期ポイントに関しては、Mach システムの方が比較的良い結果が得られた。

参考文献

- [1] 神原久夫, 河野太基, 柴田義孝: パケットビデオシステムのための同期メカニズム, 情報処理学会第46回全国大会, 1K-05, 1993
- [2] 柴田義孝: マルチメディア情報ネットワークにおける同期転送方式, 情報処理学会第45回全国大会, 1B-6, 1992

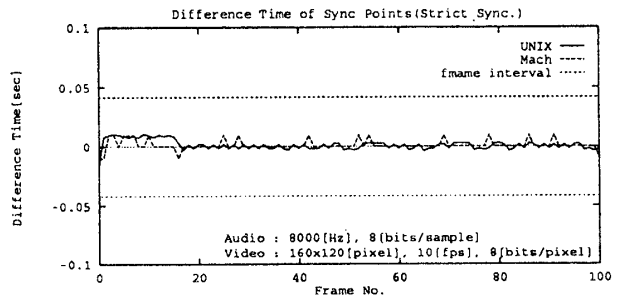


図 3: 同期の時間誤差 (Strict Synchronization)

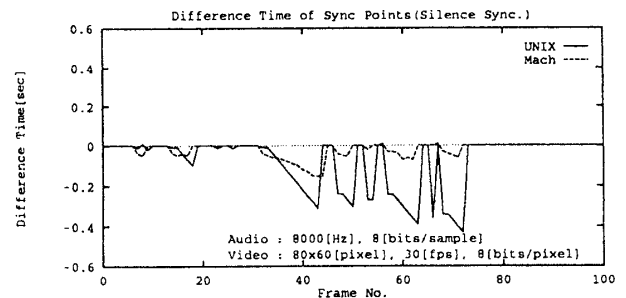


図 4: 同期の時間誤差 (Silence Synchronization)