

Keio-MMP における

6R-4 ビデオ会議システムのアーキテクチャの検討†

藤井敬三*†† 齋藤信男**

*(株)システムコア 開発技術本部

**慶應義塾大学 環境情報学部

1 はじめに

慶應義塾大学では、情報処理振興事業協会(IPA)から委託を受け、10社の企業と共同し、マルチメディア統合環境基盤ソフトウェアプロジェクト(Keio-MMPプロジェクト)を行なっている[1]。本稿では、Keio-MMPにおける分散アプリケーションプログラムとしてビデオ会議システムを例に、OS非依存部分に焦点を置き、ソフトウェアアーキテクチャについて述べ、作成したプロトタイプの評価結果について報告する。

2 基本アーキテクチャ

我々は、単一のサービスを行なう複数のサーバ(パフォーマンス)からシステム全体を構成する *Conductor/Performer* アーキテクチャを採用している[2]。分散環境においては各ワークステーション上に、ビデオ入出力を司るビデオパフォーマンス、オーディオ入出力を司るオーディオパフォーマンス、ネットワークを管理するネットワークパフォーマンス、各種サーバを統括管理し、分散アプリケーションとのインターフェイスを行なうコンダクタなどが走行する。各パフォーマンス間を結び、アプリケーションからの依頼により、コンダクタが生成する片方向通信路(ストリーム)を介して連続メディアデータが流れる。データはMDU(Media Data Unit)を処理単位として扱われる。パフォーマンス間のインターフェイスにより異なるが、例えばビデオでは1フレーム、オーディオではある時間単位(例:0.1sec)のデータがMDUに格納される。

2.1 Quality of Service

分散環境において限られた計算機資源(CPU, メモリ, I/Oなど)の元、連続メディアデータの持つ時間的制約を満足させるために、サービスの質(QOS, Quality of Service)を静的にあるいは、動的に変化させる必要がある。連続メディアデータにおけるQOSは、時間的解像度、空間的解像度に分類することができる[3]。動画像の場合、時間的解像度は単位時間当たりのフレーム(コマ)数、空間的解像度は色数、画素数などに相当する。

一方、人間の動画像に対する要求を考慮した場合[4]、QOSは動画像の性質、嗜好によりその満足度は大きく異なる。さらに、情報量の膨大な動画像に対してデータ圧縮を行なった場合、圧縮能力は、入力画像の性質に依存し、得られるデータ量は大きく影響を受ける。従って、ロボッ

トアイなどのように計測等が主目的ではない場合には、計算機資源のQOSと人間の要求するあいまいなQOSとの写像を考慮する必要がある。本稿では分散環境における共有計算機資源として、データ量(単位:bytes/sec)を各パフォーマンス間における共通パラメータとした。

2.2 Dynamic QOS Control

コンダクタは新しくアプリケーションを実行させるときに余剰計算機資源量と、そのサービスに必要な見積もった資源量から、サービス開始の可否の調停を行なう。余剰が十分でなければ、さらに少ないQOSにおける再試行を行なうか、あるいは既走行中のサービスのQOSを低下させ新規サービスの再試行を行なう。また資源量そのものは、優先度の高いジョブの起動、対話的処理の発生、一時的な負荷の増加などにより常に変動する可能性がある。従って、データの持つ論理性を損なわないため、人間に要求するQOSの範囲内で、動的に連続メディアデータ量とその処理に関わる計算量のQOSを動的に変化させる必要がある。

2.3 Multimedia Synchronization

連続メディアデータの静的動的なQOS制御を行なう場合、例えばビデオとオーディオデータでは、要求するQOSも異なり、処理の優先度も異なるため、それらをインターリーブしないで、個別に処理、移動、格納させることが必要である。異なるストリームから到着する各連続メディアデータの時間的制約を満足させ(ストリーム内同期)、なおかつストリーム間の時間的制約(ストリーム間同期、リップシンクなど)も満足させる必要がある。ビデオとオーディオではデータ量も異なり、また処理(圧縮)計算量も異なるため、処理遅延が発生し、またネットワークにおいても、さらにその遅延が拡大する(あるいは相殺される)可能性がある。従って、連続メディアのQOSとしてさらに、ストリーム内同期精度が加わる。我々はストリーム内同期のみに着目し、各ストリームが、共通に参照している時計(目標とする時刻)との差を保証することにより、結果的にストリーム間同期が保証されると考えた。ここでは許容範囲内であれば、同期が合っていると定義し、また、パフォーマンスの処理遅延が既知であれば、遅延を相殺するための任意オフセットを持つ場合でも、同期が合っているとした。

3 プロトタイプ概要

OS非依存部分のシステムアーキテクチャの検討を行なうため、Sun SPARCstation2, Parallax XVideo-24SVCを用いて、SunOS 4.1.2上にビデオ会議システムを試作した。当プロトタイプでは、ビデオとオーディオを独立したストリームとして扱い、各ストリームの遅延を測定するため、データ生成(送信)側でそれぞれタイムスタンプ(データ生成時刻として `gettimeofday()` より)を付

†"Video Conference System's Architecture in Keio-MMP"†

Keizo FUJII*†† and Nobuo SAITO**

* System Core Co.,Ltd., 1-22-3, Sangenchaya, Setagaya-ku, Tokyo, 154 Japan and **Keio University, 5322, Endo, Fujisawashi, Kanagawa, 252 Japan

†この研究は、情報処理振興事業協会(IPA)が実施している開放型基盤ソフトウェア研究開発評価事業「マルチメディア統合環境基盤ソフトウェア」プロジェクトのもとに行なわれた。

††開放型基盤ソフトウェア湘南藤沢キャンパス研究室の研究員としてIPAに登録されている。

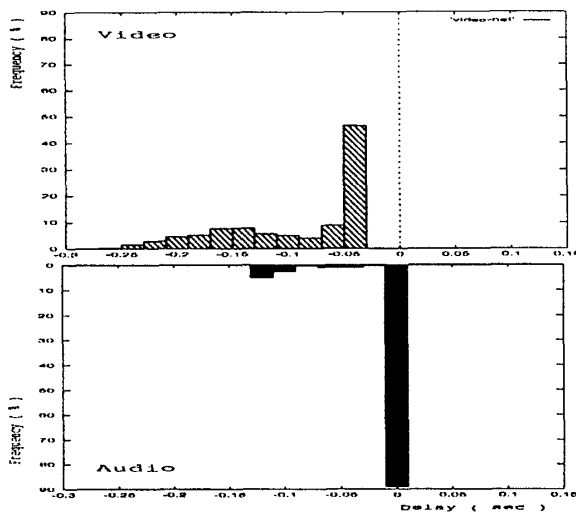


図 1: ネットワークによる遅延分散

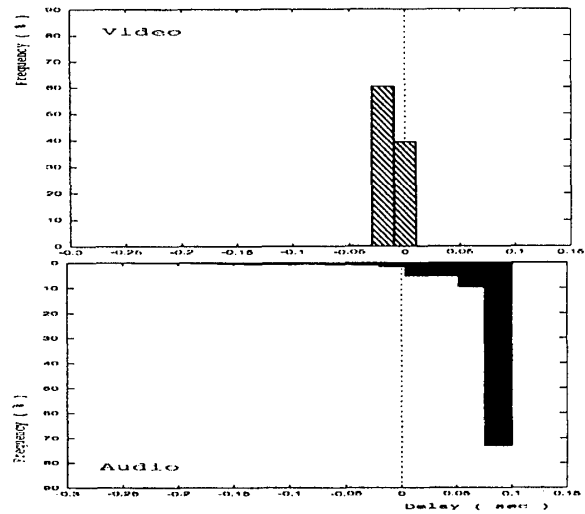


図 2: 同期処理後の遅延分散

加した。データ受信側ではマシン時刻と MDU のタイムスタンプの差により、ネットワーク部分の遅延を測定した。同期処理として、各ストリームにおいて、タイムスタンプに指定待ち時間を加算した時刻を処理（再生）開始時刻とし、その時刻まで処理を遅延させた。ビデオは XVideo のハードウェア JPEG を用いて圧縮し、QOS は、データ量は指定 bytes/sec 固定、時間的解像度として 2~10 フレーム/秒の範囲、空間的解像度として圧縮度合いを JPEG Q-factor により 25~1000 の範囲で変化させた。オーディオは、Sun Audio 8kHz, 8bits, μ -law を用い、QOS は固定とした。但し、入力レベルがある値以下のときは、無声部分と解釈し送出不ないようにした。

ビデオは MDU 発生周期可変、MDU サイズ可変 (1K~25Kbytes) とし、オーディオは MDU 発生周期固定 (約 0.05sec, 但し無音時には送出不) , MDU サイズ固定 (400bytes) とする。利用者が指定する QOS として、ビデオではデータ量 (単位:bytes/sec) と、時間的解像度、空間的解像度、視覚的要求のいずれを優先するかを指定させることとした。ビデオ入力 (圧縮含)、ビデオ出力 (伸張含)、オーディオ入力、オーディオ出力を、それぞれプロセスとして実装した。ネットワークプロトコルとして TCP/IP と UDP/IP を採用した。測定を行なったマシン間の時刻は xntp[5] により同期させマシン間のマシン時刻の差が ± 15 msec 以内であることを確認した。

なお、ビデオデータの伸張 (表示) には約 5msec, オーディオデータの再生には約 100msec の遅延が発生する。従って最終端で同期を保証するために、その差である約 100msec 相当分、オーディオを早めに (=ビデオを遅めに) 処理させた。

4 評価結果と考察

評価用 FDDI ネットワークを構築し、同期機構の有無による遅延時間を測定した。ネットワークプロトコルとして TCP/IP を用い、ビデオの QOS をデータ量 32Kbytes/sec, 視覚的要求優先とした場合のネットワークによる遅延分散 (同期処理前) を図 1, 同期処理後の遅延分散を図 2 に示す。横軸はビデオとオーディオデータそれぞれの遅延時間 (単位:sec, 負は遅れを意味する), 縦軸はそれぞれの遅延の発生頻度 (単位:%) を示している。これらの遅延には TCP/IP によるバッファリングの影響, JPEG に

よる圧縮サイズの差異による遅延変動などが含まれている。当評価では、ネットワークの遅延、遅延分散はビデオ側が大、再生 (表示) 処理遅延はオーディオ側が大であった。同期機構を導入することにより、最終端におけるビデオとオーディオの遅延差は、視覚的に違和感の少ない範囲 [6] に抑えることが可能であった。情報量の膨大なビデオデータを圧縮前状態で (もしくは伸張後状態で) 保持するためには膨大なメモリ量が必要になる。異なる能力を持つヘテロジニアスな計算機環境において、またマルチキャストを用いる場合などでは、同期させるポイントを何処に置くかを検討する必要がある。

5 おわりに

本稿では、分散環境における連続メディアを扱うアーキテクチャを検討するため、ビデオ会議システムを例として採用した。そして、データ量を制御可能とし、視覚的な要求を考慮したビデオデータの静的動的な QOS 制御方法と、複数連続メディアデータ間の同期のためにメカニズムを検討し、プロトタイプを作成、評価した。これらより、データ量 (bytes/sec) を制御し、その範囲内で時間的空間的解像度を制御させる方法が有用であることがわかった。

参考文献

- [1] 徳田, 萩野, 斎藤: 分散マルチメディア統合環境 Keio-MMP プロジェクトにおける連続メディア処理のためのソフトウェアアーキテクチャ, 第 49 回全国大会論文集, 7R-01 (1994).
- [2] 西尾, 他: “Keio-MMP における Conductor/Performer アーキテクチャの協調性能評価,” 第 49 回情処全大論文集 (7R-07) (1994).
- [3] H. Tokuda, Y. Tobe, S. T.-C. Chou and J. M. F. Moura: “Continuous Media Communication with Dynamic QOS Control Using ARTS with FDDI Network,” ACM SIGCOMM'92, pp. 88-98 (1992).
- [4] 藤井, 斎藤: 動的 QOS 制御機構を有するビデオ会議システムの評価, 信学技報, CPSY94-46 (1994).
- [5] D. Mills: “Network Time Protocol (v3),” RFC-1305 (1992).
- [6] Thomas Käpper, Dietmar Hehman and Ralf Steinmetz: “An Introduction to HeiMAT,” 3rd NOSSDAV, pp.362-373 (1992).