

2T-1 ファイルシステムレイヤによるリモートマルチメディア情報へのアクセス

久保 亮介* 嘉藤 将之† 豊田 博俊‡ 石橋 勇人* 安倍 広多* 山井 成良§ 松浦 敏雄*

*大阪市立大学大学院創造都市研究科 †岡山大学大学院自然科学研究科

‡大阪市立大学 (非常勤) §岡山大学総合情報基盤センター

1 はじめに

音声や画像などのマルチメディア情報をネットワークを介してアクセスし表示するストリーミングアプリケーションが広く利用されるようになってきた。しかし、既存のストリーミングアプリケーションの多くは独自のデータ形式や伝送プロトコルを採用しており、一般にこれらのデータ形式や伝送プロトコルの間には互換性がない。そのため、ネットワークを経由してメディアデータを共有するには、データ形式と伝送プロトコルを統一するか、もしくは、複数の形式に対応したクライアントを用意しなければならなかった。

これに対して、NFS を用いるとサーバにあるファイルをあたかもローカルなファイルのようにアクセスできるので、サーバ側で MPEG などの標準形式でメディアデータを格納しておけば、クライアント側ではローカルなファイルとして置かれたメディアデータを表示・再生できる任意のアプリケーションプログラムを用いることができる。しかし、NFS はファイルシステムの性質上、データの内容は正確に伝送されるけれども、実時間性などの QoS が考慮されていないため、LAN のように高品質な通信を行えるネットワーク環境でしか利用できない。

そこで、本稿では、NFS と同様にローカルなファイルシステムを拡張した、QoS 制御可能な、リモートマルチメディア情報へのアクセス手法を提案する。本手法を用いることで、ローカルなファイルに対する任意のアプリケーションプログラムを利用した場合でもファイルアクセスに対して QoS 制御を行うことができる。

2 試作システムの概要

試作システムでは NFS と同様にリモート上のファイルシステムをマウントすることで、ローカルディスク上にあるファイルへのアクセスと同様にリモート上にあるファイルへアクセスすることができる。

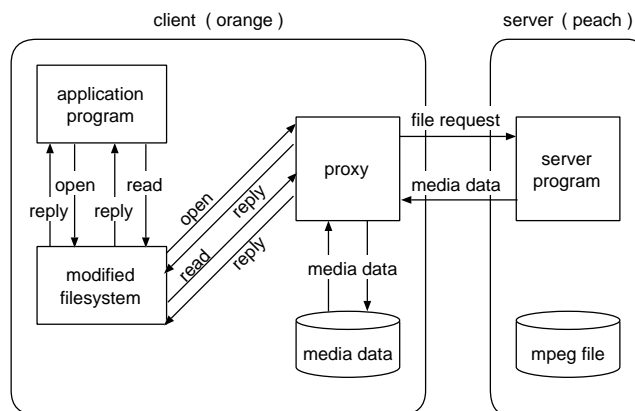


図 1: 試作システムの構成

試作システムの構成を図 1 に示す。リモート上のファイルへのアクセスの際、アプリケーションプログラムが発行する open や read の要求は modified filesystem に対して送信される。ローカルディスク上にあるファイルへのアクセスとリモート上にあるファイルへのアクセスの差異はファイルシステムによって吸収されるため、アプリケーションプログラムからすればその違いを意識することなくアクセスすることができる。modified filesystem はアプリケーションプログラムから受信した open や read の要求を proxy に送信し、proxy からメディアデータを受信してアプリケーションプログラムに供給する。また、QoS 制御に関しては proxy - server 間で行われる。proxy が server から受信したメディアデータはネットワーク状況に応じて QoS 制御されたデータであり、低品質なデータが蓄積されている可能性もある。

3 提案手法の設計・実装

ここでは modified filesystem について詳述する。proxy - server 間の具体的な QoS 制御手法に関しては本稿では割愛する (文献 [1])。

3.1 modified filesystem の開発環境

独自のファイルシステムを構築する際の環境として、Linux のユーザ空間でファイルシステムを構築することができる UVFS [2] を用いた。UVFS では本来カーネル空間で行う処理をユーザ空間で行うことができるため、

Remote Multimedia Data Access through Filesystem Layer
R.Kubo*, M.Kato†, H.Toyoda‡, H.Ishibashi*, K.Abe*,
N.Yamazi§ and T.Matsuura*

*Graduate School for Creative Cities, Osaka City University

†Graduate School of Natural Science and Technology, Okayama University

‡Osaka City University

§Information Technology Center, Okayama University

オペレーティングシステムの破壊を恐れずにファイルシステムを構築するための実験を行うことが可能であり、デバッグも容易であった。

UVFS には、UVFS 上で動作するサンプルのファイルシステムとしてメモリ空間上に構築される pmfs がある。本システムでは、pmfs の一部を書き直すという手法でマルチメディアデータの転送に必要な機能を実現した。

3.2 modified filesystem

アプリケーションプログラムが open / read 等の I/O リクエストを発したとき、pmfs では LOOKUP / READ 等のルーチンが動作する。LOOKUP は要求のあったファイルを探して見つければファイル情報を取得するという機能を持ち、READ はメディアデータを読み込むという機能を持つ。この 2 つのルーチンをリモート上のファイルシステムに対するアクセスに書き換えることで実装を行った。

以下、マルチメディアデータを保持するサーバ (peach) 上の "/SV/" ディレクトリをクライアント (orange) 上の "/MFS/peach/" にマウントする例を説明する。アプリケーションプログラムが "/MFS/peach/abc/xxx.mpg" という mpeg ファイルにアクセスすると、実際は peach サーバの "/SV/abc/xxx.mpg" というファイルにアクセスすることになる。

3.2.1 LOOKUP

LOOKUP はアプリケーションから要求されたファイルパスをルートポイント ("/MFS/") から 1 階層ずつ探索し、ファイル情報を取得しつつ、要求されたファイルにたどり着くまで繰り返す。

具体的な LOOKUP の動作として上記ファイルの例では、まず 1 回目の LOOKUP で peach サーバに接続できるか確認し、接続できたらその情報を保持する。次に 2 回目の LOOKUP で peach サーバの "/SV/" ディレクトリに "abc" という名前のファイルを探し、見つければそのファイルの情報を保持する。さらに 3 回目の LOOKUP で 2 回目に見つけた "/SV/abc/" ディレクトリに "xxx.mpg" という名前のファイルを探し、見つければそのファイル情報を保持する。このように要求されたファイルにたどり着くと LOOKUP は終了し、READ へ移行する。

こうすることで、アプリケーションプログラム側には "/MFS/peach/abc/xxx.mpg" というファイルにアクセスしているように見えているが、実際にはリモート上にある peach サーバの "/SV/abc/xxx.mpg" というファイルにアクセスすることができる。

3.2.2 READ

READ は保持しているファイル情報を参照し、サーバ

側のファイルがオープンされていないとオープン要求を送信し、オープンされていれば server から proxy に転送されたメディアデータを固定サイズ (4096 Byte) ずつ読み込む。server はファイルオープン時に proxy にメディアデータの転送を開始する。このとき、READ が proxy からメディアデータを読み込む処理と、server から proxy へのメディアデータの転送は互いに独立しており非同期である。

具体的な READ の動作は、まず 1 回目の READ ではサーバ側のファイルがオープンされていないので、proxy にオープン要求を送信する。proxy 経由でオープンの要求を受信した server はファイルを開く。オープン成功の reply を proxy 経由で modified filesystem に送信する。それとともに server は、proxy にメディアデータの転送を開始する。modified filesystem はオープン成功の reply を受信すると、ファイル情報をオープンされている状態に更新する。2 回目以降の READ ではサーバ側のファイルはオープンされているので、server から proxy に転送されたメディアデータを固定サイズずつ読み込む。

server から proxy へのメディアデータの転送が遅く、proxy に先読みされたメディアデータが READ の要求に間に合いそうにないときは、proxy - server 間で QoS 制御がなされ低品質なデータが転送される。

4 おわりに

本稿では、ファイルシステムレイヤからリモートマルチメディア情報にアクセスする手法を提案・実装した。また、ユーザ側のアプリケーションプログラムとして mpeg_play [3] を用いてリモート上にある mpeg ファイルを転送しつつ再生できることを確認した。

今後の課題としては、様々なネットワーク環境に対する適応性を調べるため、帯域等を制御し動作試験を行い本システムの有効性を評価することがあげられる。

謝辞

本研究の一部は平成 15 ~ 17 年度科学研究費補助金 (基盤研究 (C)(2), 課題番号 15500048) の補助を受けている。

参考文献

[1] 嘉藤将之, 山井成良, 岡山聖彦, 久保亮介, 松浦敏雄: “ファイルアクセス API を用いた連続メディア情報のリモートアクセス手法”, 情報研報 (DPS), Vol.2004, No.89, pp.85-90 (2004-09).

[2] UVFS, <http://www.sciencething.org/geekthings/> .

[3] The Berkeley MPEG Player, <http://bmerc.berkeley.edu/frame/research/mpeg/mpeg-play.html> .