2004－DPS－119　(13)
2004－EIP－ 25　(13)
2004／9／3

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

嘉藤将之[1]　山井成良[2]　岡山聖彦[3]　久保亮介[4]　松浦敏雄[4]

[1]岡山大学 大学院自然科学研究科
[2]岡山大学 総合情報基盤センター
[3]岡山大学 工学部
[4]大阪市立大学 大学院創造都市研究科

概　要

動画像・音声などの連続メディア情報のリモートアクセス手法として，NFS を用いる方式が提案されている．しかし，この方式は QoS 制御機能がないため，広域ネットワークなど帯域が十分でない環境では適用できないという問題があった．そこで，本稿ではファイルアクセス API の拡張による連続メディア情報のリモートアクセス手法を提案する．本手法ではファイルオープン時における先読み機能，ファイル読込み時における実時間保証機能，代替データのキャッシュ無効機能などをファイルアクセス API に組み込むことにより上記の問題に対処する．また，提案方式の実装を行ない，本方式が狭帯域ネットワーク経由でアクセスした場合でも有効であることを示す．

A Remote Access Method of Continuous Media Data Using File Access APIs

Masayuki Kato[1], Nariyoshi Yamai[2], Kiyohiko Okayama[3],
Ryosuke Kubo[4], and Toshio Matsuura[4]

[1]Graduate School of Natural Science and Technology, Okayama University
[2]Information Technology Center, Okayama University
[3]Faculty of Engineering, Okayama University
[4]Graduate School of Creative Cities, Osaka City University

Abstract

NFS-based services have been proposed for transferring continuous media data. However, on a network environment like WAN that does not have enough bandwidth, NFS-based services are not suitable since they do not provide QoS function. In this paper, we propose a remote access method of continuous media data by modifying file access APIs. This method provides three functions: (1) prefetching on file open, (2) realtimeness support on file access, and (3) cache invalidation for incomplete data. This paper also discusses a design and implementation of the proposed method and shows that our method is effective even on narrow band network environment.

1 はじめに

最近，計算機の高機能化やネットワークの高速化に伴い，音声や動画像などの連続メディア情報をネットワークを介してアクセスして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．このようなアプリケーションでは，表示・再生のために一定時間内に次々とメディアデータを供給しなければならず，従来のメディアを扱うアプリケーションよりも厳しい要件が課せられる．従って，ストリーミングアプリケーションの多くは，限定されたネットワーク環境のみで利用可能であったり，アプリケーションが使用するデータ形式や伝送プロトコルが独自のものに限定されたりしていた．そのため，専用のアプリケーションプログラム以外では，メディアデータにアクセスできないという問題があった．

一方，連続メディアを扱うアプリケーションプログラムの多くは，たとえばMPEG[1]のような標準的な形式のファイルをローカルディスクから読み込んで表示・再生する機能を有している．この点に着目した連続メディア情報のリモートアクセス手法として，NFS[2][3]を直接用いた方法（以下，NFS方式）[4]やNFSプロトコルをプロクシで変換する方法[5](以下，NFSプロクシ方式)がある．これらの方法では，クライアントはファイルとしてメディアデータにアクセスするため，サーバが標準的な形式でファイルを提供すればクライアントではその形式を扱える任意のプログラムを使用することが可能になる．

ところが，連続メディア情報を取り扱う際には実時間性など普通のファイルとは異なるQoS(Quality of Service)が要求されるにも関わらず，NFS方式やNFSプロクシ方式では正確性のみが重要視され，実時間性等は考慮されていない．このため，これらの方法はLANやイントラネットなど限定されたネットワーク環境にしか適用できなかった．

そこで，本稿ではファイルアクセスAPIへのQoS制御機能の組込みによる連続メディア情報のリモートアクセス手法を提案する．本手法では，ライブラリの動的リンク機構を利用しているアプリケーションプログラムであれば原理的には任意のものを利用でき，NFS方式やNFSプロクシ方式の利点を活かしながらQoS制御を行うことが可能になる．

2 従来手法の問題

既存のストリーミングアプリケーションの多くは，限られたディスク容量で大量のデータを格納し，また限られたネットワーク帯域でQoSを保証しながら伝送するため，アプリケーションプログラム毎に独自に工夫された表現形式や伝送プロトコルを採用している．MPEGやRTP(Real Time Protocol)など一部のものを除き，一般にこれらの間には互換性がないため，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，サーバ側で使用するアプリケーションプログラムと同じ種類のものをクライアント側でも使用したり，サーバ側で何種類ものアプリケーションプログラムに対応したりする必要があった．

これに対して，アプリケーションプログラムに依存しない連続メディア情報のリモートアクセス手法として，NFS方式が提案されている．NFS方式ではサーバ側のデータがクライアント側ではファイルとしてアクセスできるため，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その形式のファイルを表示・再生できる任意のアプリケーションプログラムをそのまま用いることができる．しかし，NFS方式ではファイルシステムの性質上，データの正確性のみが重要視され実時間性などのQoSが考慮されないため，LANのように高品質な通信を行えるようなネットワーク環境でしか利用できない．NFSプロクシ方式では，イントラネットのように十分な帯域を有するものの伝送遅延やジッタを無視できないようなネットワークでもNFS方式を利用できるように，クライアント側にプロクシを配置してプロトコル変換を行い，サーバに連続的にメディアデータを送信させる手法が提案されている．しかし，この手法も帯域が十分でないネットワーク環境では利用できない．

このように，従来の連続メディア情報のリモートアクセス手法は特定のアプリケーションプログラムに依存したり，あるいはQoS制御機能がなく利用可能なネットワーク環境が限定されたりするなどの点で問題があり，いずれも不十分であるといえる．

3 ファイルアクセスAPIを用いたリモートアクセス

前章で述べたように，NFS方式やNFSプロクシ方式はサーバ上のメディアデータをクライアントはファ

イルとしてアクセスできるため，特定のアプリケーションプログラムに依存しないという特徴を有する．そこで，本研究ではこの特徴を活かしながら QoS 制御機能を実現するため，ファイルアクセス API を拡張する手法を提案する．

3.1 ファイルアクセス API の利用

NFS 方式において QoS 制御機能を実現するには，NFS プロクシ方式のようにクライアント側にプロクシを設置し，その中に同機能を組み込む方法が考えられる．しかし，QoS 制御機能を導入するとデータの正確性が犠牲になるため，ネットワーク帯域が不十分な期間に一度低品質（不正確）なデータがプロクシから OS に渡されるとこのデータがキャッシュされ，ネットワーク帯域が回復した後に再びアクセスしても低品質なデータしか得られないという問題が生じる．

そこで，本稿では open()，read()，あるいは fopen()，fread() などのファイルアクセス API を拡張する手法を提案する．この手法ではクライアント上で既存のアプリケーションプログラムをそのまま利用できるかどうかが新たな問題となる．しかし，多くのプログラムでは動的リンク機能を用いてこれらの API を実行時にリンクするため，そのようなプログラムは再コンパイルなどを必要とせずにそのまま利用することが期待できる．

3.2 ファイルアクセス API 拡張の方針

3.2.1 データの先読み

通常，広域ネットワークの特性によるによる映像や音声の乱れは，クライアントのアプリケーションプログラム内にあるバッファのサイズを大きくすることで，ある程度は回避することができる．しかし，既存のアプリケーションプログラムに対してこのような変更を加えることは一般には困難である．

そこで，提案手法では API 側で大きなバッファを設け，open() や lseek() などの API が呼び出されたとき，あるいは read() などの API が呼び出されてバッファが枯渇しそうになったときに先読みを行うようにする．これによりアプリケーションプログラムや OS に変更を加えることなく，広域ネットワークの特性によるによる映像や音声の乱れを抑えることが可能になる．

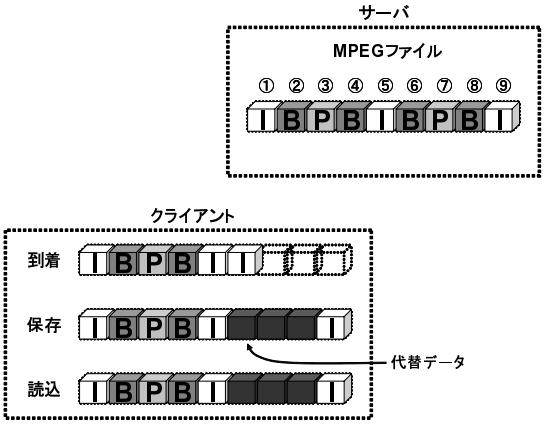

図 1: QoS を考慮したデータの先読み

3.2.2 実時間性の保証

データの先読みにより，バッファリングを行なったとしても，ネットワークの利用可能帯域が長期的に十分でない場合は，バッファの枯渇が防げない場合がある．このような場合，従来のファイルシステムの概念では，データを誤りなく読み書きすることが重要であるが，連続メディア情報を表示・再生する場合はデータの正確性を多少犠牲にしても，実時間性を保証することが望ましい．そこで，提案手法では，データの先読みが要求期限までに間に合わない場合は重要性の低いデータを欠落させることにより，実時間性を保証するようにする[1]．

ここで，欠落が生じた場合，アプリケーションプログラムに欠落部分を切り詰めたデータを提供してしまうとデータのファイル内でのオフセットやファイル全体のサイズが見かけ上変わってしまうという問題が生じる．すなわち，ファイルアクセス API を用いてファイルにアクセスする場合，データのファイル内オフセットが変わってしまうと，たとえば任意の時点からの再生などに不具合が生じることが予想される．そこで，バッファの枯渇が生じた場合には，パディングなどを行うことによりオフセットの変更が生じないようにする．

ファイル形式として MPEG を用いる場合の例を図 1 に示す．MPEG ファイルは，I ピクチャ，P ピクチャ，B ピクチャの 3 種類の画像コーディングパターンで構成される．この中で，特に重要なものが，他のピクチャ生成の際にも基準となる I ピクチャである．ネットワーク帯域が十分である場合には，1〜4 番データのように全種類のピクチャが先読みされバッファに格

[1] 欠落ではなく低品質の代替データを提供する方法もあり得る．

納される．ここで，4 番目のデータ転送が完了した時点でネットワークの帯域不足によりバッファの枯渇が発生する状態になったと想定すると，これ以降はサーバは I ピクチャ(5 番，9 番のデータ) のみを伝送するようにする．他のピクチャ（6〜8 番のデータ）は欠落するため，サイズが同じ代替データを格納し，9 番のデータのオフセットが変わらないようにする．代替データとしてはたとえば先頭に sequence error start code を表す 0x000001b4 を書き込み，残りを全て 0 で埋めたものが考えられる．

以上のような方法の採用により，提案手法ではオフセットやファイルサイズを変更することなく，実時間性を保証しながらメディアデータをファイルとして提供することが可能になる．

3.2.3 代替データのキャッシュ無効化

多くのファイルシステムでは，同一データに対するアクセス高速化のため，キャッシュが設けられている．したがって，NFS プロクシ方式に単純に QoS 制御機能を組み込むと，3.1 節の冒頭で述べたようにネットワーク状態が悪い時期にアクセスしたために代替データが格納された箇所を状態が改善されてから再びアクセスしてもキャッシュ機能が有効に働くため代替データしか得られない可能性がある．

このような問題を避けるため，提案手法では API 自身がキャッシュ制御を行い，代替データへのアクセスが予想される場合にはその部分だけキャッシュを無効化して高品質データの再取得を試みる．なお，キャッシュされた高品質データに対してアクセスがあった場合には，キャッシュが有効に機能し，サーバからのデータ再取得は行わない．

4 提案手法の設計・実装

4.1 試作システムの構成

提案手法ではライブラリ中のファイルアクセス API を置き換えるため，ファイルアクセス API を動的にリンクする全てのプログラムが影響を受けることになる．従って，もし全ての機能をファイルアクセス API 自身に組み込むと，リンク後のプログラムが肥大化し，主記憶を浪費することになる．そこで試作システムでは，図 2 のようにクライアント内に一種のプロクシを導入して前章で述べた様々な機能を実装し，ファイルアクセス API 自身はプロクシとの通信機能のみ

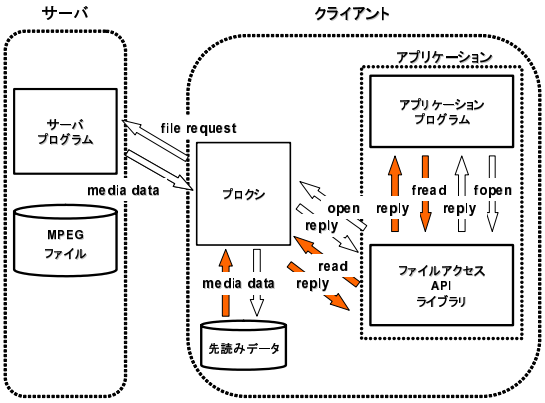

図 2: 試作システム構成とメッセージの流れ

を組み込むように設計した．なお，サーバ・プロクシ間の通信には，フロー制御，伝送誤り発生時の再送処理，データの順序保証などを行うため，これらの機能を全て有する TCP を用いたが，UDP や RTP などのプロトコルを用いることも可能である．

機能拡張の対象となるファイルアクセス API としては，open(), read() などの低レベル API と fopen(), fread() などの高レベル API の 2 種類がある．高レベル API は内部で低レベル API を使用するため，低レベル API を機能拡張すれば高レベル API を使用しているアプリケーションプログラムにも対応することができる．このため，本来であれば低レベル API を機能拡張すべきであるが，試作システムでは動作試験に使用した mpeg_play[6] が高レベル API を使用していたことから，実装の容易さを優先して高レベル API を機能拡張することにした．具体的には，fopen(), fread(), fseek() などの高レベル API のソースプログラム中に含まれる低レベル API の呼び出し部分において，特殊なパス名を持つファイルにアクセスした場合にのみ上記の通信機能を作動するように改造した．

この改造により，試作システムは以下のように動作する．

アプリケーションプログラムが特殊なパス名を持つファイルを引数として fopen() を呼び出すと，その内部処理において open() を呼び出す段階で fopen() はプロクシに対してサーバ上の指定されたファイルから先読みを行うようにプロクシに指示する．プロクシは一定量のデータがバッファに格納された時点で fopen() に応答を返し，更にバッファが満杯になるまで先読みを続ける．一方，fopen() はプロクシからの応答を受けてアプリケーションプログラムに結果を返す．引き続いてアプリケーションが fread() を呼び

出すと，その内部処理において read() を呼び出す段階で fread() はプロクシに対して指定されたファイルデータをプロクシに要求する．プロクシは先読みしたデータあるいは先読みが間に合わなかった場合には代替データを fread() に返し，fread() はこれをアプリケーションプログラムに渡す．

4.2 プロクシの内部構成と動作

次にプロクシの内部構造を述べる．図 3 にプロクシの内部構造と，そのメッセージの流れを示す．

プロクシの内部には，主にファイルのアクセス状況を管理し，ファイルアクセス API との通信を担当する制御スレッドと，主にサーバから送られるデータを受信し，キャッシュの管理や代替データの提供を担当するデータ受信スレッドの 2 種類が存在する．また，サーバにもプロクシの各スレッドと通信を行う 2 種類のスレッドが存在する．

プロクシの制御スレッドでは，アプリケーションが次に読み込むべきデータのオフセットと先読み済みのデータのオフセットを管理し，データ受信が間に合わない場合は，サーバの制御スレッドに QoS 制御メッセージを送信する．サーバの制御スレッドはこのメッセージを受信すると，データ送信スレッドを制御し，これ以降は I ピクチャのみを送出させる．但し，その場合でもクライアント側で再生の同期を取るために必要な情報（pack start code から picture start code までのデータ）は常に送るようにする．

サーバのデータ送信スレッドは指定されたファイルのデータをオフセット付きで送出する．プロクシのデータ受信スレッドはこれを受信すると指定されたオフセットの位置にそのデータを格納する．このとき同時に受信済みデータのオフセットを調べ，データの欠落を検出する．データの欠落を検出した場合，欠落した部分には代替データを用意して保存する．さらに，欠落したデータのファイル内オフセットとデータ長を保存し，再度同一ファイルに対してアクセスが発生した場合には，前回再生した品質よりも高品質で再生できるように欠落部分のデータを要求する．

4.3 動作試験

提案手法の有効性を検証するため，試作システムの動作試験を行った．この試験ではサーバに MPEG 1 ファイル（サイズ 5.7MB，再生時間 30 秒）を用意し，クライアントのアプリケーションプログラムとして mpeg_play を用いた．サーバとプロクシは C 言語で実装した，サーバおよびクライアントに使用した計算機の仕様を表 1 に示す．なお，mpeg_play は画像の同期をとるための機構が用意されていないため，データ作成時の速度で再生できない．そのため，同期がとれるように改良した状態で動作試験を行った．

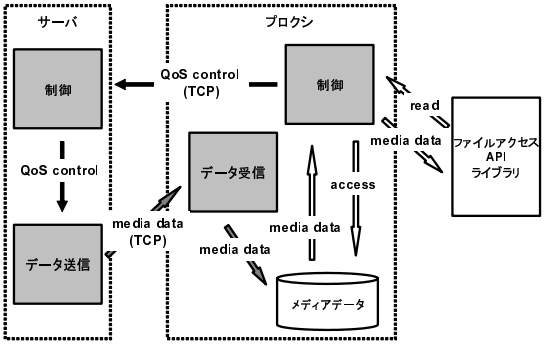

図 3: プロクシの内部構造とメッセージの流れ

表 1: サーバおよびクライアントの仕様

	サーバ	クライアント
CPU	Pentium4 1.8GHz	Celeron 900MHz
主記憶	512MB	256MB
OS	FreeBSD4.5R	FreeBSD4.8R

動作試験では，利用可能なネットワーク帯域が不十分な状況下での有効性を検証するため，サーバ側で dummynet[7] の帯域制限機能を利用し，帯域を 5.0Mbps, 1.5Mbps，1.0Mbps の 3 通りに設定した場合について各 10 回再生を試み，正常に再生できるかどうかを確認した．また比較の対象として NFS 方式を用いた場合についても同様の試験を行った．

動作試験の結果を以下に述べる．

まず，帯域を 5.0Mbps に制限した場合は，提案手法，NFS 方式ともに毎回映像は途切れることなく正常に再生された．これは試験に使用した MPEG ファイルの符号化ビットレートが 1.5Mbps であることから，十分な帯域であると考えられ，妥当な結果であると言える．

帯域を 1.5Mbps に制限した場合は，NFS 方式は正常に再生されたが，提案方式では毎回において再生が開始してすぐにコマ送り状態で再生され，それ以降は正常に再生された．コマ送りの状態では P, B ピクチャが欠落して I ピクチャのみが再生された結果である．再生開始直後でコマ送りが発生した理由としては，アプリケーションプログラム自身がバッファリン

グを行うために最大速度で大量のデータを読み込もうとした結果であると思われる．したがって，プロクシが fopen() に応答を返す時点でのバッファ内のデータ量を増やすことにより，この現象は改善されると推測できる．

帯域を 1.0Mbps に制限した場合は，NFS 方式では再生開始直後に再生が停止してしまい，それ以降は復旧できなかった．一方，提案手法では 10 回の試行でそれぞれ発生するタイミングは異なるものの，30 秒の再生の間に 10 回弱のコマ送りが発生しながら最後まで再生することができた．このことから提案手法は帯域が不十分な状況でも実時間性を損なうことなく I ピクチャのみをうまく再生できると言える．

次に，代替データのキャッシュ無効化機能を確認する実験を行った．この実験では，まず帯域を 1.0Mbps に制限した状態で再生を行って多くの代替データが生成される状況にした後，帯域を 5.0Mbps に緩和した状態で再び再生を行った．その結果，2 回目の再生では代替データのキャッシュが無効化され，1 回目の再生ではコマ送り状態になっていた部分も含めて正常に再生された．

次に，他のアプリケーションプログラムでも代替データを含むファイルを正常に再生できるかどうかを確認する実験を行った．使用したプログラムは FreeBSD 上で動作する gxine[8] および Windows 上で動作する QuickTime Player[9] の 2 種類である．この実験で用いたアプリケーションプログラムでは，使用しているファイルアクセス API の種類が異なる，あるいは試作システムと OS が異なるなどの理由により，試作システムと組み合わせて使うことが困難であったため，最初に行った動作実験において帯域を 1.0Mbps に制限した状態でプロクシに格納されたキャッシュデータをファイルとして準備し，これを各プログラムが正常に再生できるかどうかを確認した．その結果，代替データ部分ではコマ送りが発生し，その他の部分では正常に再生されることが確認された．

以上の結果により，提案手法は帯域が不十分な環境においてもコマ送り状態で実時間性を保ちながら再生を続けることが可能であり，また多くのアプリケーションプログラムへの対応が期待できるため，有効かつ実用的であるといえる．

5 まとめ

本稿では，ファイルアクセス API の拡張により，リモート上の連続メディア情報のアクセス手法を提案した．また，試作システムの動作試験を通してその有効性を確認した．

今後の課題としては，利用可能なアプリケーションプログラムの範囲を拡張するため，代替データを工夫して mpeg_play のような同期機構を持たないアプリケーションプログラムでもそのまま利用できるようにすることがあげられる．また，試作システムでは高レベル API を拡張したが，提案手法を低レベル API の拡張により実装することも挙げられる．これに加えて，QoS 制御機能の改良も今後行いたい．すなわち，試作システムにおける P，B ピクチャの欠落だけでは不十分な場合，I ピクチャの解像度を下げることにより必要なネットワーク帯域を更に減少させ，提案手法を適用できるネットワーク環境を増やすことも重要な課題である．

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] 藤原洋: 最新 MPEG 教科書，アスキー，1994.

[2] Sun Microsystems, Inc.: “NFS:network file system protocol specification”, RFC1094, IETF, 1989.

[3] B. Callaghan, B. Pawlowski, and P. Staubach: “NFS version3 protocol specification”, RFC1813, IETF, 1995.

[4] 大村猛，廣田照人，中西正典，岡秀幸: “ビデオサーバソフトウェア— Video Network Server—その設計と実装評価”，電子情報通信学会論文誌 (D-II)，Vol.J79-D-II, No.4, pp.626–633, 1996.

[5] 山井成良，浪平大輔，安倍広多，下條真司，松浦敏雄，村山孝三: 広域ネットワークにおける NFS を用いた連続メディアデータアクセス方式，情報処理学会論文誌，Vol.42, No.2, pp.178-187, Feb.2001.

[6] The Berkeley MPEG Player, http://bmrc.berkeley.edu/frame/research/mpeg/mpeg_play.html

[7] dummynet, http://info.iet.unipi.it/~luigi/ip_dummynet/

[8] gxine, http://xinehq.de/index.php/home

[9] QuickTime Player, http://www.apple.co.jp/quicktime/