システムソフトウェアと
オペレーティング・システム 87−19
（２００１．６．２９）

資源の予約管理機構に基づく リアルタイムオペレーティングシステムの設計

浅見 和男† 森本 洋行† 毛利 公一†† 吉澤 康文††

†東京農工大学大学院工学研究科
††東京農工大学工学部

Video on Demand に代表されるマルチメディアアプリケーションを実行する時，オペレーティングシステム (以下，OS) は，計算機資源の割当てをリアルタイムに実行可能であることを保証し，高速なデータ処理を行う必要がある．しかし，既存の OS では，資源の割当てに容量のみが指定可能で，時間の概念は無かった．また，巨大なコンテンツの効率的なアクセス手法は提供されず，ストリーム配信におけるオーバヘッドは大きなものとなる．現在，これらの問題を解決するために，オペレーティングシステム MOViE を開発している．本論文では，特に，MOViE の構成と，資源管理，コンテンツ管理，タスク間通信のオーバヘッド軽減の各手法について述べる．また，カーネルとの通信のコストを試算し，そのコストを軽減可能であることを示した．

Design of the Realtime Operating System based on Resource Reservation Management

Kazuo Asami† Hiroyuki Morimoto†
Koichi Mouri†† Yasufumi Yoshizawa††

†Graduate School of Engineering, Tokyo University of Agriculture and Technology
††Faculty of Engineering, Tokyo University of Agriculture and Technology

When multimedia application software such as Video on Demand are executed, an operating system (OS) needs to guarantee realtime assignment of the computer resources to the task and to perform high-speed data processing. However, in the existing OS, only capacity could be specified to be assignment of computer resources and the time could not been specified. The efficient access method of huge contents data is not offered, so the overheads of multimedia tasks could not be reduced. In order to solve these problems, we are developing operating system MOViE. In this paper, especially we describes the composition of MOViE, resources management, contents management, and technique of reducing overheads of IPC. The communication cost between a user task and a kernel was estimated, and it was shown that it is possible to reduce that cost.

1 はじめに

ADSL や IMT-2000 によって，パーソナルコンピューティング環境へのマルチメディアコンテンツの配信サービスが本格化しつつある．このようなサービスは，ソフトリアルタイム処理が可能であり，マルチメディアコンテンツのようなサイズが大きいデータを効率的に扱うことも可能とし，かつ広域ネットワーク環境やモバイル環境といった種々のネットワーク環境に適応可能で，スケーラビリティが高いマルチメディアサーバシステムを必要とする．

しかし，従来のマルチメディアサーバシステムの処理は，単純にコンテンツデータを 2 次記憶装置から読み出し，単純なフローコントロールを用いてネットワークへ送信するだけである．また，クライアントにおいても，コンテンツデータの到着が遅延しても再生可能とするために，必要以上に大きいバッファを用意して受信を行う必要があった．そのため，次のような課題がある．

(1) 携帯端末のようなクライアントが，マルチメディアコンテンツを受信するためには，サーバがクライアントの状態に従って適切な Quality of Service(以下，QoS) 制御を行い，クライアントが必要とする計算機資源の消費を抑える必要がある．

(2) 早送りなどの特殊な操作が発生した際に，動的なコンテンツデータの解析をおこなったり，それぞれの処理専用のコンテンツデータファイルを用意するなどの負担を軽減する必要がある．

(3) MPEG や AVI などのコンテンツデータは，それぞれのデータ構造にしたがって構成されており，ファイルを構成する個々の要素は可変長となっている．このような場合，単位時間辺りのデータ処理量を目安としてコンテンツデータを処理するのではなく，単位時間辺りの要素数を目安とする必要がある．

(4) リアルタイム処理とスケーラビリティの向上のため，種々のオーバヘッドを軽減する必要がある．

以上の課題を解決するために，MOViE では，次に示す機能を提供する．

(1) 資源予約機構
(2) 資源枯渇時の交渉に基づく資源調整機構
(3) 1 レベルストアによるコンテンツ管理機能
(4) カーネル拡張モジュールによるコンテンツ管理機能
(5) カーネルとユーザ間でのゼロコピー通信

(1) によってリアルタイムタスクが必要とする資源を確保するとともに，CPU 以外の資源の割当てにもリアルタイム性を保証する．(2) によってタスクの過剰な資源確保をふせぐ．(3) によって，ポインタを基本とするコンテンツデータへのアクセスが可能となるため，早送りなどの特別な操作のためのデータをポインタの集合として表現可能となり，特殊再生データを別途用意する必要がなくなる．また，(4) では，特定のデータ構造に基づいて 2 次記憶から先読みするなどの機能をユーザが指定可能とする．これによって，コンテンツデータが可変長の場合でもデータのページフォールトを防止できる．さらに，(5) によって，OS 内部および OS とアプリケーション間の通信におけるオーバヘッドを軽減し，スケーラビリティを向上させることが可能となる．

以下，本論文では，2章で連続メディア処理の特徴と現状の OS での問題点を述べる．3章で MOViE の特徴を提示し，4章では Linux のオーバヘッドを解析し，MOViE の機能によって改善できることを提示する．

2 連続メディアの処理方式

2.1 マルチメディアアプリケーション

マルチメディアは文字や音声，動画像などの様々なメディアを組み合わせ，統一的に扱う新たな情報の表現方法である．以下にマルチメディアを用いたアプリケーションの例を示す．

(1) Video on Demand

ユーザの要求を受け，ネットワークを介して動画像と音声をユーザに対し配信するサービスである．

動画像と音声データは共に小さなデータ構造の集合であり，集合単位ごとに周期的に資源を利用しながら処理が行われる．1つ1つの処理が正しく行われることよりも，全体として時系列を守ることが重視される．また，通常は周期的なデータ処理を行うが，ユーザからの突発的な要求によって非連続的なデータ参照を行う特徴がある．

(2) テレビ会議システム

テレビ会議システムはカメラで取得した画像を音声とともに遠隔地へ伝送し，複数人で会話を可能にするマルチメディアサービスである．

テレビ会議システムは動画像を配信するサービスであるが，2次記憶装置を用いない非蓄積型であり，データの生成や圧縮などをリアルタイムに行う特徴がある．

(3) マルチメディアデータベース

既存の文字だけでなく，多様なメディアとその相互関連を構築可能なデータベースである．

映画の場面や台詞を検索するなど，マルチメディアデータの利便性を向上させる機能としても利用可能である．メディア間の相互関連を持つ構造は，データ構造や関連を記述可能なファイルシステムを実現することにより，その構築が容易になる．

現在，特にその普及が期待されているサービスに，動画像や音声などの連続メディアを用いたマルチメディアサービスがあげられる．ネットワークの発達により，今後は様々なマルチメディアサービスにおいて，この連続メディアの利用が増加すると予想される．

2.2 連続メディア処理の特徴

VoDやテレビ会議システムなどのサービスは，利用者の満足度を維持するために，安定した動画像の配信と処理が要求される．もしリアルタイム処理のデッドラインを満たせない場合は重要度の低い評価項目の品質を低下させるなどのQoS制御を行わなければならない．例えば動画再生処理において，処理のデッドラインを超過する場合には，データの解像度を一時的に低下させ，処理時間を削減してリアルタイム性を優先する制御が行われる．

テキストやプログラムなどの既存のデータの処理は，その数値的な正確さを評価の対象としているが，リアルタイム性という評価を受けることは稀である．一方，連続メディアはリアルタイム性を持つ処理を要求されるという特徴がある．この規定された時間に対して，いかに正確な処理を行うかによって，主に連続メディアサービスはその品質が決定される．連続メディア処理は突発的なユーザからの処理の変更要求やデータのストリーム配信を行うため，ネットワークを含めた処理のリアルタイム性を要求される．分散マルチメディアサービスではクライアントのリアルタイム処理はサーバのデータ供給のリアルタイム性に依存するため，特にサーバのリアルタイム処理は重要である．

連続メディアデータのサイズは非常に大きく，動画像ではギガバイトオーダのデータになる場合がある．しかし，連続メディア処理はこの長大ファイルの全域を参照して，処理を行うわけではない．mpegファイルであれば，数Kbyteから数十Kbyteのピクチャと呼ばれるデータが多数集合することによって1つの長大なコンテンツを形成している．mpegに限らず連続メディアは小さなデータの繰り返しによって形成され，その構成単位を対象として周期的な処理が実行される．

以上をまとめると，連続メディア処理には次のような特徴がある．

(1) QoSの制御

(2) リアルタイム処理

(3) 構造を持つ長大データの処理

2.3 既存のOSでの問題点

連続メディアを主とするマルチメディアサービスを既存のOS上で提供するに当たって，以下の問題点が挙げられる．

(1) 資源管理

QoSの制御には計算機資源のリアルタイムな割り当てが不可欠である．既存のOSで

は計算機の資源管理の方針として，仮想化によって実際以上の資源を利用可能として実行タスク数を増加させるものが多い．しかし，資源の要求時割り当ては予測できないオーバーヘッドを伴うため，リアルタイム性を乱す場合がある．また，既存の OS ではタスクからの資源要求に対して無制限に資源を与えてしまうため，リアルタイムタスクのための資源が枯渇する可能性もある．

(2) リアルタイム処理のサポート

既存の OS では CPU の公平な分配をタスクスケジューリングの方針としているものが多い．これは計算機の利用目的を対話処理などのリアルタイム性を要しない処理と想定し，設計を行ったことに起因するものである．

また，これまでリアルタイム OS の研究が進められていた分野は，主に組込み機器や機器制御などのハードリアルタイム OS であった．連続メディア処理では厳密なリアルタイムを必要としない．また，ハードリアルタイム OS は多くのタスクを実行できないため，連続メディア処理には適さない．

(3) 大規模データへのアクセス方式

連続メディアデータのサイズはテキストなどの既存のデータに比べ，非常に大きいという特徴がある．主に 32bit ポインタを用いる既存の OS では，連続してデータを操作する場合には，そのサイズを 4Gbyte 以下に抑える必要があり，問題となっている．

また，UNIX ではファイルのアクセス方法として，汎用的なファイルアクセス手段が提供されている．これはテキストデータを処理することを想定したものであり，データ構造を定義することはできず，連続メディアを処理する際にはデータのサーチなどの処理によって大きなオーバーヘッドが生じる．

計算機のあらゆる処理はユーザタスクとカーネルとの通信によって行われる．既存の OS ではデータの保護の観点から，ユーザとカーネル間の通信はデータのコピーによって行われる．しかし，連続メディアはデータ量が大きく，このコピーのオーバーヘッドが問題となる．

3 MOViE の特徴と構成

2 章で述べた問題点を解決し，OS が様々な連続メディア処理に対応するため，MOViE では次に示す機能を提供する．

(1) 時間を意識した資源予約機構の提供

(2) 交渉に基づく資源調整機構の提供

(3) ソフトリアルタイムスケジューリング機構

(4) 効率的なデータ検索および配信を考慮した連続メディア管理機構

(5) モジュールによるカーネル機能の拡張

(6) カーネルとユーザ間のゼロコピー通信

以下に MOViE の構成と特徴について述べる．

3.1 MOViE の構成

MOViE は図 1に示す各機構によって構成される．計算機資源は資源抽象化層によって抽象化される．資源予約，調整機構によって連続メディアのリアルタイム処理と QoS 制御に必要な計算機資源の配分を行い，リアルタイムスケジューラによって実際にリアルタイムタスクのスケジューリングを行う．また，巨大な連続メディア配信のオーバヘッドの削減のために，カーネル拡張モジュールによるコンテンツ管理と連続メディア管理機構によってデータへのアクセスを行い，ゼロコピー通信機構によってユーザとカーネル間通信の高速化を実現する．

3.2 資源予約および調整機構

既存の OS において，計算機資源はその容量のみが管理の対象とされ，その資源が割り当てられるまでの時間は保証されていない．メモリを確保する時に 2 次記憶装置に実メモリ上のデータをページアウトする必要が生じた場合には，大幅な

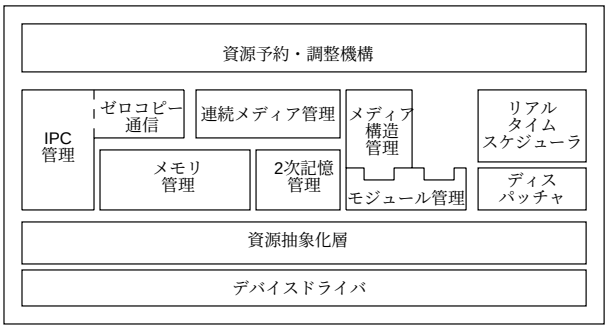

図 1　MOViE の構成

応答遅延が発生する．連続メディア処理を行う場合には，この種の予測できない遅延が問題となる．

連続メディア処理のリアルタイム性を保証するために MOViE では，CPU 以外の資源を含め，資源の利用に時間の概念を導入し，予約を行うことでリアルタイムタスクに対して一定時間内での応答を保証する．メモリを確保する場合，要求にレスポンスタイムを設定することによって，指定時間内で割り当て可能なメモリをユーザに対して提供する．これによってメモリ確保が完了するまでタスクがブロックされ，デッドラインを超過することを防ぐことができる．

また，CPU 以外の資源についても資源の予約に基づく配分を行う．ネットワーク資源であれば，単位時間あたりの転送データ量という形で資源を確保することによって，リアルタイム性を持ったタスクの通信帯域が非リアルタイムタスクの帯域によって圧迫されることを防くことができる．

また，資源が枯渇した時の挙動についても定義する必要がある．各タスクが確保および予約した資源と利用率を常に監視し，不必要に大量の資源を確保しているタスクから資源を奪う機構を提供する必要がある．このために MOViE では，資源交渉機能を備える．ユーザのタスクは，必要な資源が確保できない場合に，MOViE の交渉機能を利用しながら，他の低優先度のタスクから資源を奪うことを可能にする．

そのために，MOViE では，資源の調整を次の処理によって実行する．

(1) 各タスクの駆動周期とデッドラインを取得する

(2) CPU 割当て時間内に使用できる計算機資源の量を取得する．

(a) 各資源へのアクセス速度から，タスクの実行時間内で使用可能な資源量の予測値を計算する．

(b) 実際に確保している資源の利用頻度を取得する．

(3) 今までの使用頻度を考慮して，不必要に多くの資源を確保しているタスクを抽出し，資源を剥奪する．

駆動周期とタイムクオンタムの小さなタスクは他のタスクよりも即時性を求められていると考え，より小さなタイムクオンタムで動作し，小さな資源要求を繰り返し行うタスクに対して高い資源の優先度が与えられる．一方，大きなタイムクオンタムで，一度に大量の資源を要求するタスクの資源の優先度は低くなる．

MOViE では OS が提供する資源予約，交渉機構は計算機資源の操作に時間概念を導入することと，資源枯渇時の動作を規定することを目的とする．QoS 管理などに必要な資源量表現の変換は，拡張や変更を容易にするために外部機構を用いて実現する．

3.3　リアルタイムスケジューリング機構

リアルタイム性を持つ連続メディアを処理するタスクに対しては，周期的に CPU を割り当てる必要がある．このため，資源予約機構と協調してリアルタイムスケジューリングを行う．

以下に，MOViE のリアルタイムスケジューリング機構の機能を示す．

(1) レートモノトニックスケジューリング

連続メディアの周期性とリアルタイム処理を考慮したスケジューリング方式として，MOViE では連続メディア処理の周期性を考慮したレートモノトニックアルゴリズムをベースとするスケジューラを採用する．また，連続メディア処理はソフトリアルタイム性を持つため，MOViE ではタスクの厳密なリアルタイム性を追求せず，デッドライン

の超過を通知する機構によって回復処理の実行を可能とする．

(2) I/O スケジューリング

リアルタイムスケジューリングを行うときの大きな問題として優先度逆転現象がある．この現象は主に入出力デバイスを複数のタスクが利用するときに発生する現象である．現在，I/O アクセスのタイミングをスケジューリングすることによって解決する研究を行っている．

(3) スケジューラの拡張

タスクのスケジューリングアルゴリズムは処理する連続メディアの種類やその処理内容によって最適な方式は異なる．他のスケジューラを容易に MOViE へ適応するためにスケジューリングアルゴリズムを容易に変更できるインターフェイスを持たせる．

リアルタイムタスクのデッドライン超過を検知した場合には，その情報をタスクだけでなく，資源管理機構に対しても通知する．デッドラインミスの回復処理の呼び出しを行い，資源管理機構内では優先度の再計算を行うことによって，連続メディアを処理するタスクのリアルタイム性を維持する．

3.4 連続メディアデータ管理機構

既存の OS ではデータサイズが問題となり連続メディアデータを扱うことは困難であった．また，長大データを考慮したファイルのアクセス方法は提供されていなかった．MOViE では動画像に代表される長大サイズの連続メディアデータの管理手法として，64bit 単一アドレス空間上で 1 レベルストアを用いるデータ管理機構を提供する．

1 レベルストアを用いることにより，連続メディアのデータ構造を直接アドレス空間上に投影する事が可能である．よって，データ構造を改変せずに各連続メディアの構成単位にアクセスするための手法を提供することが可能であり，単一アドレス空間によってデータは一意なアドレスで表せる．例えば，mpeg データは図 2のようにリンクを用いたデータ構造で表現することにより，構成単位であるピクチャへのダイレクトアクセスが可能となる．

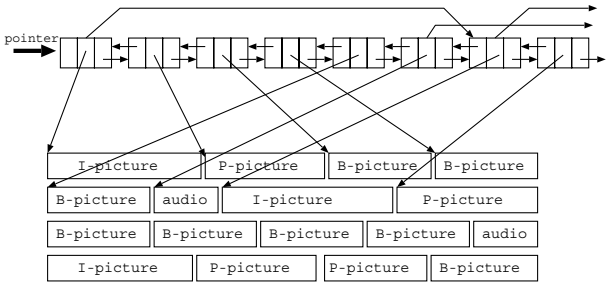

図 2　mpeg 構造のリンクによる表現

このリンクを適宜辿ることによって，再生，早送り，巻戻し，音声のみの再生などの処理を，ファイルからの読み込みを明示することなく，ポインタによる参照で行うことが可能である．特に早送りにおいて，キーフレーム切り替え方式 [2] を用いる場合でも，別ファイルとしてキーフレームファイルを生成する必要がなく，無駄なディスク資源の使用を防ぐことが可能である．

また，利用者がリンク情報を保持することによって，任意の場面からの再生や頭出しなどの様々なデータの参照操作や，リンクの繋ぎ変えによる動画編集を行うことが可能である．この連続メディア管理機構を用いることによって，サービスコンテンツを管理するデータベースとの連携を行うことが容易になる．

1 レベルストアは仮想記憶を利用した記憶管理の手法である．仮想記憶ではメモリアクセス時に実メモリを割り当てるため，あらかじめメモリアクセスのコストを見積もることは難しい．MOViE では連続メディアの構造に基づく先読みをカーネル内のモジュールによって実行することにより，メモリアクセスのリアルタイム性を確保する．

3.5 カーネル拡張機構

取り扱うデータや処理内容によって，ユーザの OS に対するサービス要求は異なるものとなる．最大公約数的な機能を提供する OS ではリアルタイム処理などの特殊な処理をサポートするには限界がある．OS の拡張や機能の変更を可能にすることでアプリケーションプログラムの処理に適し

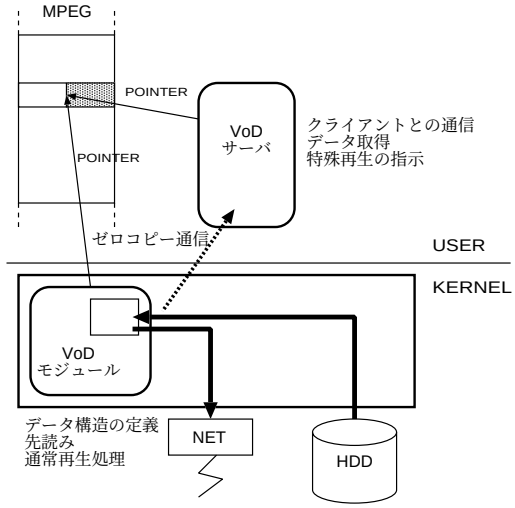

図 3　連続メディア処理を支援するモジュール

た OS サービスを提供することが可能になる．

MOViE ではカーネルを拡張させる機能をモジュールとして開発し，そのモジュールをカーネルにリンクさせることによって拡張性を持たせる．サーバは処理を停止させることを極力避けるべきであり，モジュール管理機構は OS 全体を停止させることなく動的にロード可能な機構にする．このモジュール管理機構を用いることによって，連続メディアの種類や，構造にあわせた OS 機能の拡張が可能となる．

この機構を用い，連続メディアを処理するモジュールを拡張する事によって，多様な連続メディアデータの特徴に適した制御を行うことが可能となる．図 3に示すように，連続メディアのデータ構造をモジュール内で定義し，このデータ構造を元に先読みを実行した場合，無駄なデータアクセスをすることなく，ヒット率を向上させることが可能である．また，後述のゼロコピー通信と組み合わせることにより，高速なデータ渡しを行うことができる．

3.6　ゼロコピー通信機構

既存の OS ではメモリの保護の問題からカーネル内のデータはユーザへコピーによって渡される．しかし，連続メディア処理で発生するデータを頻繁にコピーする場合，そのオーバヘッドが問題になる．MOViE ではこのオーバヘッドの軽減のため，メモリの保護属性の切り替えを用いたタスク間の通信を行う機構を提供する．

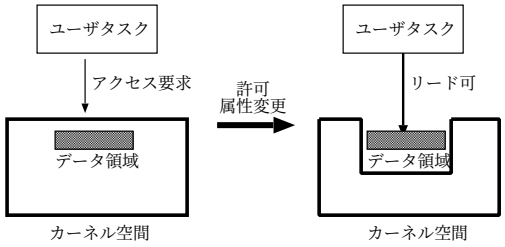

図 4　属性切り替えを用いたコピーの削減

カーネル内の一部のデータ領域の保護属性を，図 4 のようにユーザからの要求によって一時的にユーザタスクからのリードを可能にする．これによりコピーなしのデータ通信が可能になる．データ量が大きい連続メディアを通信内容とした場合には，メモリコピーに比べ空間切り替えのコストが小さくなり，メモリコピーのオーバヘッドを軽減したデータ渡しが可能である．

通常，連続メディアサービスの処理は取得したデータを参照し，送信の判断を行った後，送信の指示を出すことによって処理が行われる．そのため，データのリードが主な処理であり，データの保護の問題とも両立が可能である．

4　通信コスト削減の試算

MOViE のゼロコピー通信を用いてユーザとカーネル間の通信を行った場合のコストの試算を行った．また，メモリコピーによる通信との比較を行った．Intel x86 アーキテクチャCPU を用いて，1 ページ 4kbyte のデータをページの属性を変更して参照した場合と，実際にメモリコピーを行った場合の所要クロックの試算を表 1と表 2に示す．

メモリ属性の変更は PTE の操作や TLB フラッシュによって発生するコストが大きいが，1 ページ以上のデータを対象とする場合には，コピーのコストの方が上回ると考えられる．この試算を適応した場合，512byte 以上のデータを転送する場合はページテーブルを書換えてゼロコピー

表 1　メモリ属性変更のコスト

処理	クロック数
必要な PTE を検索	27〜60
PTE の内容を書換え	1
TLB のフラッシュ	25〜50
TLB ミスのペナルティ	(9〜13) × (32+64)
合計	917〜1359

表 2　メモリコピーのコスト

処理	クロック数
メモリコピー	(1〜2) × 4096
合計	4096〜8192

通信を行った方が高速なデータ通信が可能となる (図 5).

5　おわりに

本論文では，ソフトリアルタイム処理が可能であり，マルチメディアコンテンツのようなサイズが大きいデータを効率的に扱うことも可能とし，かつ広域ネットワーク環境やモバイル環境といった種々のネットワーク環境に適応可能で，スケーラビリティが高いマルチメディアサーバシステムを構築するための OS MOViE の設計について述べた．また，Intel x86 プロセッサによるユーザとカーネル間のメモリコピーの処理時間について，その概算と考察を示した．

MOViE は，時間の概念を導入した資源管理と交渉機構とリアルタイムスケジューラによってソフトリアルタイム処理を実現する．また，1 レベルストアを用いた連続メディア管理機構とモジュールによるデータ構造の定義によって，長大コンテンツの管理と無駄な先読みを抑え，オーバヘッドの軽減を可能としている．さらに，ゼロコピー通信機構によっても，連続メディアサーバを実行する上でのオーバーヘッドを軽減可能としている．

今後，本論文で述べた設計に従って，1 レベルストアによる連続メディア管理機構などの開発を進める予定である．

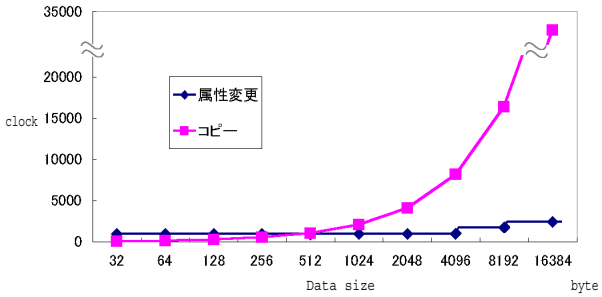

図 5　メモリコピーコストの試算

参考文献

[1] Thomas Plagemann, Vera Goebel, and Pal Halvorson, “Operating System Support for Multimedia System, ” Computer Communications Journal, 1999.

[2] 蓬田 義一, 吉澤 康文, “キーフレームストリームによる特殊再生機能を持つ VOD システムの実現, ” 情報処理学会第 56 回全国大会論文集，pp. 347–348，1998.