連続メディア配信システム：S³におけるメモリ管理機構の開発

浅見 和男† 帆波 幸二† 高野 了成† 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‡

1．はじめに

動画や音声などの連続メディアデータを配信するサーバは，ディスクとネットワークのI/Oを周期的に行う．I/O処理は一般的にCPUに比べ低速であり，多数のクライアントに対するサービス提供のボトルネックになる傾向がある．本稿では上記のボトルネックを解消するために連続メディア配信システムS³におけるI/O削減を目的とする新しいメモリ管理方式を提案する．

2．目標

S³のメモリ管理機構は連続メディア配信サーバの性能向上のために以下の機能を実現することを目標としている．

(1) ディスクI/O回数の削減

ディスクから読み込んだ連続メディアデータはクライアントに非依存な再利用可能なデータである．この連続メディアデータをキャッシュすることにより msec オーダであるディスク I/O 処理の発生回数を削減し，かつCPUのオーバヘッドも削減する．

(2) 連続メディアの参照特性に基づくページ管理

既存のLinuxにおけるファイルデータのキャッシュであるページキャッシュはLRUアルゴリズムを用いているが，LRU はシーケンシャルアクセスが主体である連続メディアへのアクセスには不適当である．S³では連続メディアへのアクセスパターンを考慮したキャッシュのリプレースメントを行う．

3．S³の概要

S³はストリーミングデータをキャッシュするStream Manager とディスクアクセス時間の見積もりに基づくI/Oのリクエストスケジューリングを行う Disk I/O Manager[1]から構成される．以下，本稿では Stream Manager についてその構成について述べる．S³全体のソフトウエア構造は図1に示す通りである．

4．Stream Managerの構成

ストリームデータのI/Oはクライアントと1対1で対応する必要はなく，データがメモリ上に存在していれば再利用が可能である．また，連続メディアデータは複数のクライアントから要求され，読み込みのみであるためデータを共有可能である．

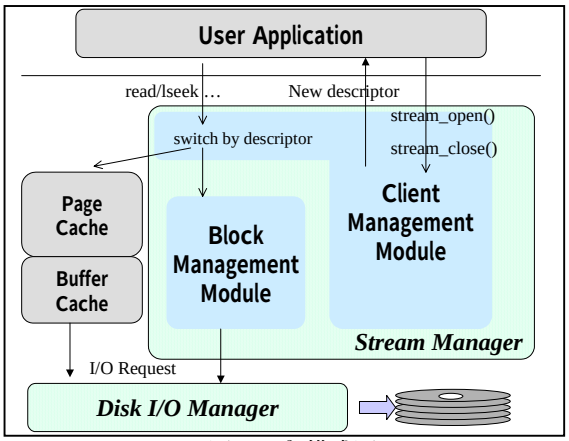

図1 S³ 構成図

S³では複数のクライアントが同時に配信サービスを受けるシステムをターゲットとしているため，スループットの向上が目標となる．

4．1 Stream

連続メディアサーバは動画像や音声など複数のストリーミングデータをクライアントへ送出する．しかしMPEG4ファイルでは映像，音声など複数のストリーミングデータは分離して格納されている．この性質の異なるストリーミングデータに対してStream Manager はファイル中の連続領域を Stream として定義する．既存のI/O操作ではできないファイルよりも細かな単位である Stream に対して適切な先読みサイズやアクセスパターンなどを設定することができる．また，Stream を定義することでファイルの先頭や終端に付加されているメタデータなどの管理データをストリーミングデータから除外することが可能となる．

4．2 Block

表1 Block の状態

状態	意味	解放順位
Free	メモリ未割り当て	なし
Reserved	Disk I/O完了待ち	2
Pavement	先読み完了	3
Hot	User がアクセス中	4
Reclaim	再利用待ち	1

連続メディアはシーケンシャルアクセスであり，先読みは有効なスループット向上手段である．ゆえに，先読みが確実にヒットする効率的なキャッシングを実現する必要がある．S³では Stream に対して Block と呼ばれる単位で I/O を発行する．この際必要に応じてメモリの解放をする Block リプレースメントを行う．Block は I/O 処理により表1に示す状

Development of Memory Manager in S³ System
Kazuo Asami†, Koji Honami†, Ryosei Takano†, Yasufumi Yoshizawa‡
†Graduate School of Engeneering, Tokyo University of Agriculture and Technology
‡Faculty of Engeneering, Tokyo University of Agriculture and Technology

態を持つ．これらの Block 状態は後述する Block リプレースメントの優先順位としても利用される．

4．3 Client Management Module

Client Management Module は連続メディア配信サーバに対し，Stream へのアクセス手段となるシステムコールを提供する．stream_open はファイルに格納されているストリームの位置やビットレートなどの情報を S^3 にあたえ，利用開始を宣言する．stream_open が返すディスクリプタを用いて read, lseek を発行し，S^3 の Block Management Module 経由でデータを取得する．また，stream_close によりアクセスの終了宣言を行う．これらの仕様は図 2 に示す通りである．

```
int stream_open(int fd,
                //ファイルを open したディスクリプタ
                long long offset,
                //ストリームの開始位置[byte]
                long long length,
                //ストリーム長 [byte]
                int bitrate);
                //ビットレート[byte/sec]
int stream_close(int sdesc);
                //stream_open が返すディスクリプタ
```

図 2　システムコール仕様

4．4　Block Management Module

(1)Block 状態管理

クライアントごとに Block を管理するために Spanning_group と呼ばれるデータ構造を定義した．Spanning_group は stream_open システムコールの処理で生成される．(図 3 参照)

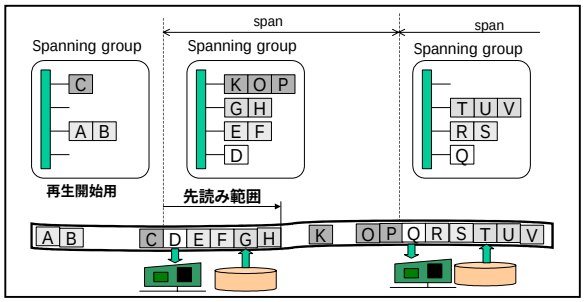

図 3　Spanning_group

Spanning_group は 1 つのクライアントに対応し，クライアントが将来利用する可能性がある Block を保持する．Spanning_group は Hot(図 3 D)，Pavement（図 3 EF），Reserved(図 3 GH)，Reclaim(図 3 KOP)の 4 つの状態の block を管理する．同一の Stream を再生している Spanning_group 同士はリストとして管理される．

先読み範囲内において Block にメモリが割り当てられていないときはメモリを割り当て，Reserved とし，I/O 要求を発行する．Spanning_group が保持する Block の状態はディスク I/O の完了などによって変化する．データの送信後 Hot が使用済みになると，後方の Spanning_group の Reclaim に移し，後続のクライアントの再利用に備える．また，常に Stream の始点は Spanning_group であり，再生開始時のディスク I/O を削減することが可能となる．

Spanning_group を定義することにより，クライアントの通常再生におけるアクセス位置の進行が Block の移動として表現でき，Block の存在の有無，ディスク I/O が必要なクライアント状態等の管理が容易になる．

(2)Block リプレースメント

Free Block が不足した時は使用中の Block の解放を行う必要がある．LRU アルゴリズムではリプレースメントの対象となった Block がデッドライン直前に再び必要となる場合が生じる．そこで本方式では以下に示す Stream のアクセス傾向に基づく新しいリプレースメント方式を提案する．

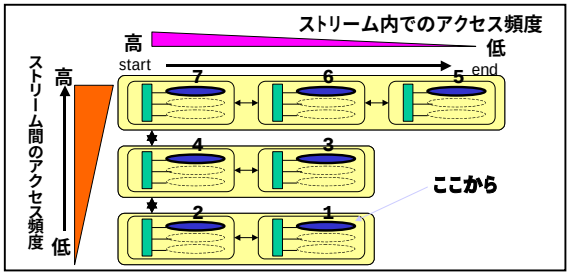

図 4　リプレースメントの優先度

I．Spanning_group の数が最も少ない Stream 中最も Stream 終端に近い Spanning_group を選択する（図 4 の 1）

II．Spanning_group 内では表 1 に示す解放順位に基づき Block を選択する

複数の Stream が再生されている時には，クライアントの嗜好によりアクセス頻度に大きな偏りが生じると思われる．Spanning_group の数はその頻度を示すことになる．また，再生は先頭から行われ，途中で再生を中止するクライアントもあるため，先頭ほどアクセス頻度は高くなる．Spanning_group 内では Block の状態はデッドラインを示すため，デッドラインに最も遠い Reclaim から解放する．また，Spanning_group 同士が接近しているときは Block を保持することで後続の I/O を全て省略可能である．これは Hot 間の距離である span が先読み範囲以下になり，Block が全て Pavement になることで実現できる．

5．まとめ

本稿では連続メディア配信サーバのディスク I/O 削減を目標とした S^3 におけるメモリ管理方式について述べた．本方式では Spanning_group を管理構造体として用いることで複数のクライアントの管理を容易にし，ストリーム間，ストリーム内でのアクセス頻度と Block の優先度を考慮した block リプレースメントの有効性について述べた．今後は具体的な評価を進めたい．

参考文献

[1]帆波幸二他,”連続メディア配信システム:S^3 におけるディスクスケジューリング方式”,情報処理学会第 65 回全国大会,2003