ストリーミングメディアの参照特性に基づく入出力削減方式

高野了成[†] 浅見和男[†,☆]
帆波幸二[†,☆☆] 吉澤康文[††]

ストリーミングサーバの実現において，参照特性に基づいてコンテンツをメモリ上に配置することで，入出力回数を削減できる．本論文では，クライアントごとに周期的に行われるシーケンシャルアクセス特性とコンテンツに対する参照頻度特性に着目したメモリ管理手法としてスパニンググループキャッシング方式を提案する．本方式では，ホットスポットが近いリクエストをグループ化し，優先的にメモリに常駐させる新しいアルゴリズムによるディスクアクセスの削減を実現する．方式の有効性を確認するために LRU との比較をシミュレーションにより評価した．その結果，本方式はキャッシュカバー率が 4%の場合，高負荷時で 2.7 倍，低負荷時で 2.8 倍の入出力回数の削減が期待できると予測された．さらに本方式はコンテンツの動的な参照要求に対して自動的に対応可能であり，コンテンツ常駐化対象の選択にも効果がある．

An I/O Reducing Strategy Based on Streaming Media Workloads

Ryousei Takano,[†] Kazuo Asami,[†,☆] Koji Honami[†,☆☆]
and Yasufumi Yoshizawa[††]

To reduce physical I/O requests by using buffer cache is required in streaming servers. However, a commonly used cache algorithm such as LRU is not effective in streaming media workloads. In this paper, we propose a new I/O reducing strategy called the Spanning Group Caching (SGC) for streaming servers. It is based on the characteristics of stremaing media workloads such as the following: (1) the access patern is periodical and sequential, (2) the popularity distribution is usually highly skew. A simulation study of comparisons with the SGC and LRU shows about 2.7 times I/O reduction under high workloads and about 2.8 times under low workloads. As a result of SGC, what parts of the streaming media are automatically chosen to reside in the cache adapting with the changes in the popularity distribution.

1. はじめに

ストリーミングサーバでは大容量のファイルを多数のクライアントへ配信する必要があり，高速かつリアルタイム性の高い入出力機構が要求される．そこでディスク上のファイルをメモリ上にキャッシュすることで，入出力回数の削減が期待できる．利用可能なメモリ容量の増大，MPEG4 に代表される動画像圧縮技術の進歩が進展しており，映画の再生時間が 2 時間前後と一定であることを前提に考えると，サーバ上により多くのコンテンツをキャッシュすることが可能になった．この大規模なキャッシュを利用し，ファイルの先読みや常駐化などの手法が用いられているが，汎用的なキャッシュ管理アルゴリズムではこのキャッシュを効率的に利用することができないため，ストリーミングサーバに特化した高度な技術開発が必要とされている．

従来のオペレーティングシステム（以下，OS と記す）の多くは LRU（Least Recently Used）や CLOCK などのアルゴリズム[4] をキャッシュ管理に採用している．これらは大容量ファイルに対するシーケンシャルアクセスが主体となるストリーミングメディアには向いていない．その要因として，LRU は局所参照性だけを期待したアルゴリズムであり，複数クライアントからの連続メディアに対するアクセス時の参照特性といった大局的な情報を利用していないことがあげられる．

そこでクライアントごとに周期的に行われるシーケ

† 東京農工大学大学院工学研究科
Graduate School of Technology, Tokyo University of Agriculture and Technology
†† 東京農工大学工学部
Faculty of Engineering, Tokyo University of Agriculture and Technology
☆ 現在，日本電気株式会社
Presently with NEC Corporation
☆☆ 現在，株式会社日立製作所
Presently with Hitachi, Ltd.

ンシャルアクセス特性とコンテンツに対する参照頻度が一様ではないこと（人気コンテンツに対する再生要求が圧倒的に多い事実）に着目し，これらの状態情報を OS 内部のキャッシュ管理に利用する手法を本論文で提案する．本手法による入出力削減方式をスパニンググループキャッシング方式と名付けた．

以下，本論文では 2 章でストリーミングサーバにおけるキャッシュ管理機構に対する要求分析について述べ，3 章では新しく提案するスパニンググループキャッシング方式の設計について述べる．4 章ではシミュレーションによる本提案の効果予測結果について述べる．5 章では関連研究について述べる．

2. 要求分析

本章では想定するストリーミングシステムとストリーミングメディアの特徴，そして既存のキャッシュ管理アルゴリズムの問題点について述べ，ストリーミングサーバ向けキャッシュ管理機構に対する要求分析を行う．

2.1 ストリーミングシステム

想定するストリーミングシステムの全体像を**図 1** に示す．クライアント，サーバ間では，ストリーム配信するセッションごとに，次に述べる複数のコネクションを利用して通信を行う．

クライアントは RTSP（Real Time Streaming Protocol）[5] を利用して，ストリームの再生，停止などの操作をサーバへ要求し，ストリーミングメディアデータは RTP（Realtime Transport Protocol）[6] を利用して配信される．同時に遅延やジッタなどの統計情報が RTCP（Real Time Control Protocol）[6] を利用してサーバに通知される．

標準的なストリーミングメディアである MPEG4 ファイルは映像，音声トラックとそれぞれに対するメタデータを格納したヒントトラックから構成される．映像，音声トラックには再生に利用されるメディアデータ本体と RTP パケットを生成するために必要となる時間情報などを格納した RTP パケットヒントが含まれる．

ストリーミングサーバは最初にヒントトラックを読み込み，メディアデータと RTP パケットヒントの対応を取得する．続いてクライアントからのリクエストを受信し，セッションを確立すると，映像，音声トラックに対してシーケンシャルにファイルアクセスを行い，ストリーム配信を開始する．

2.2 ストリーミングサーバにおけるキャッシュ管理

ストリーミングサーバにおいて性能上のボトルネックになるのはファイル入出力である．図 1 に示すようにサーバはファイルをメモリ上にキャッシュすることで，入出力回数を削減することが可能である．

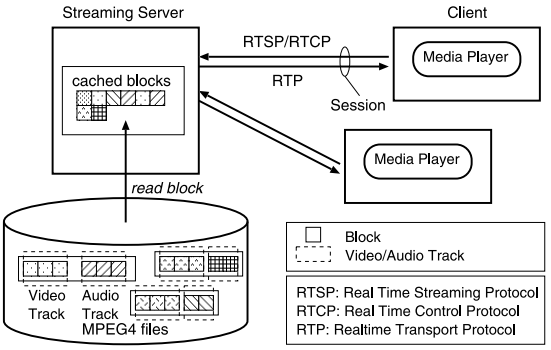

図 1 ストリーミングシステム
Fig. 1 Overview of a streaming system.

たとえば Linux 2.4.20 においてファイルから 4 KB の read を行った場合，キャッシュヒット時には実行時間が平均 9.7 μs，ダイナミックステップ数が 567 であるのに対して，キャッシュミス時には実行時間が 1.4 ms，ダイナミックステップ数が 11,700 という測定結果がある．このようにキャッシュを活用することによって入出力と CPU 資源の節約が可能である．

したがってストリーミングサーバの性能を向上させるために，ストリーミングメディアの参照特性を活かしたキャッシュヒット率の高いキャッシュ管理を実現することが有効である．なお本論文では入出力の単位をブロックと定義し，キャッシュはブロック単位で行うこととする．

2.3 ストリーミングメディアの特徴

ストリーミングサーバの実現に関連するストリーミングメディアの特徴として，(1) セッションごとに周期的に発生するシーケンシャルアクセス特性，(2) コンテンツに対する参照頻度の不均一性，(3) リアルタイム性があげられる．

(1) シーケンシャルアクセス特性
- 数百 MB から数 GB とサーバのキャッシュに収まりきらないファイルに対してシーケンシャルにアクセスが発生するため局所参照性が期待できない．
- ファイルサイズは伸縮せず，セッション内では入出力要求が一定周期で発生するので，将来のアクセスが予測可能である．

(2) 参照頻度の不均一性
- 人気コンテンツに対する再生要求が圧倒的に多く，コンテンツに対する参照頻度が一様では

ない．
- 1セッションは数十分から数時間と長時間継続するので，参照頻度の高いファイルに対して複数のアクセスが同時並行して発生する．

(3) リアルタイム性
- 遅延やジッタがクライアントでの再生品質に影響を及ぼすため，リアルタイム性が重要である．
- リアルタイム処理には処理時間の見積りが重要であるが，たとえばディスクアクセス時間において大きな割合を占めるシーク時間はシーク移動距離によって数ミリ秒から十数ミリ秒の幅がある．キャッシュ管理はディスクスケジューラと協調して先読みを行うなど，処理時間のゆらぎに対応する必要がある．

2.4 既存のキャッシュ管理アルゴリズムの問題点

従来のOSの多くが採用しているLRUアルゴリズムをストリーミングサーバにおけるキャッシュ管理に適用した場合に次のような問題点が考えられる．

(1) 公平なキャッシュ割当て
配信中のストリームごとに均一量のブロック（全キャッシュ容量/ストリーム数）がキャッシュされるため，リクエスト到着間隔がキャッシュされたブロックの再生時間より長い場合，キャッシュはまったく再利用されない．つまりストリーム数の増加に従いキャッシュ効率が悪化する．
そこでストリームのリクエスト到着間隔，コンテンツの参照頻度といったストリームデータ参照特性に応じたキャッシュの傾斜配分が必要になる．たとえば参照頻度の偏りが大きい場合は，人気の高いコンテンツをメモリ上に常駐させる手法が有効である．この際，常駐化対象を人気の変化に合わせてどのように選択するか，キャッシュ容量の何割を常駐化に割り当てるかが問題となる．

(2) デッドラインを考慮しないキャッシュ再利用
シーケンシャルアクセス下ではLRUの挙動はFIFO（First In First Out）に近似する．これは後続するストリームによって先に再利用されるキャッシュから順に解放されることを意味する．そこでデッドラインが遠いキャッシュから，つまりFILO（First In Last Out）で解放する方が再利用される可能性が大きい．

3. スパニンググループキャッシング方式

本章ではストリーミングサーバの具体例としてS^3システムについて述べ，本システムが提供しているストリーミングサーバ向けキャッシュ管理であるスパニンググループキャッシング方式の設計について述べる．

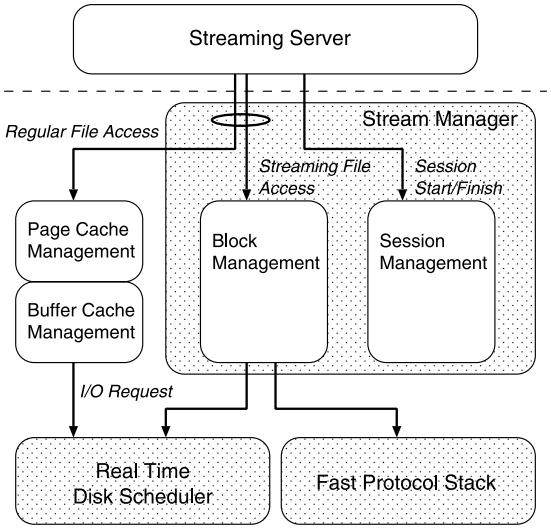

図2 S^3システムの全体構成
Fig. 2 Modules in the S^3 system.

3.1 S^3システムの概要

S^3システム[1]は高性能なストリーミングサーバを実現することを目的としており，**図2**に示すようにストリームマネージャ[2]，リアルタイムディスクスケジューラ[3]，高速プロトコルスタックを提供する．

S^3システムではOSがストリーミングメディアに対する参照特性を把握するためにセッション情報を利用する．この理由はセッション中は特定のファイルに対して一定周期でシーケンシャルなアクセスが発生するため，入出力要求が予測可能であること，およびコンテンツあたりのセッション数が動的な人気度の代理指標になるからである．

ストリームマネージャはOSが提供する通常のファイルキャッシュ管理を代替し，ストリーミングサーバに特化したキャッシュ管理を提供する．ストリームマネージャはセッション管理部とブロック管理部から構成され，前者はストリームとコンテンツの対応を管理し，後者はブロックの状態管理，ディスクスケジューラへの入出力要求を行う．

3.2 基本設計

スパニンググループキャッシング方式（以下，SGCと記す）はストリームマネージャのようなキャッシュ管理部で使用することを想定したストリーミングサーバ向けキャッシュ管理手法である．SGCの基本的なアイデアは，コンテンツの参照頻度とリクエスト到着間隔という参照特性に応じて，各ストリームに対する

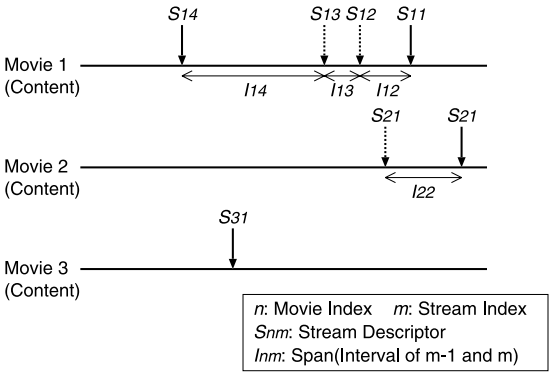

図 3 ストリームとコンテンツの関係
Fig. 3 Stream and content.

キャッシュ割当てを傾斜配分することにある．そこで参照特性に応じたキャッシュの割当てと，無駄なキャッシュ再利用を回避する仕組みを提供する．

またストリーミングメディアの性質からファイルアクセスはシーケンシャルに行われることが予測されるので，キャッシュの利用状況を参照しながら先読みを行う．一方，ネットワーク入出力はビットレートに合わせて周期的に行う．このようにコンテンツに対するディスク，ネットワーク入出力は非同期に処理されるので，これをサポートするキャッシュの状態管理を提供する．

以降の節では SGC の基本となるストリーム管理，ブロック状態管理について述べ，SGC を実現するためのデータ構造，各アルゴリズムについて述べる．

3.2.1 ストリーム管理

ストリームとコンテンツの関係を**図 3** に示す．横軸は時系列を表し，右側のストリームほど再生が先行していることを意味する．そしてリクエスト到着間隔をスパンと呼ぶ．たとえば $Movie\ 1$ に対するストリーム S_{11} と S_{12} に対するスパンは I_{12} となる．

基本的には先行するストリームによってディスク入出力が発生し，ブロックがキャッシュされるため，追従するストリームはそのキャッシュを利用することで入出力が削減できる．このように連続するストリーム間ですべてのブロックがキャッシュされ，ディスク入出力が発生しない状態をブリッジと呼ぶ．SGC ではより多くのブリッジを構築し，維持することによって入出力削減の向上を目指す．

多くのブリッジを構築するにはスパンが短いストリーム間のブロックを優先してキャッシュに残すことが有効である．たとえば図 3 ではスパン I_{12}，I_{13}，I_{22} が短いので，これらのスパン間でブリッジを構築する．この際，破線矢印で示したストリーム S_{12}，S_{13}，S_{21}

表 1 ブロックの状態
Table 1 Block states.

状態	意味	スティール優先順位
Free	メモリ未割当て	なし
Reserved	ディスク入出力完了待ち	2
Pavement	ディスク入出力完了	3
Hot	ネットワーク配信中	なし
Reclaim	再利用待ち	1

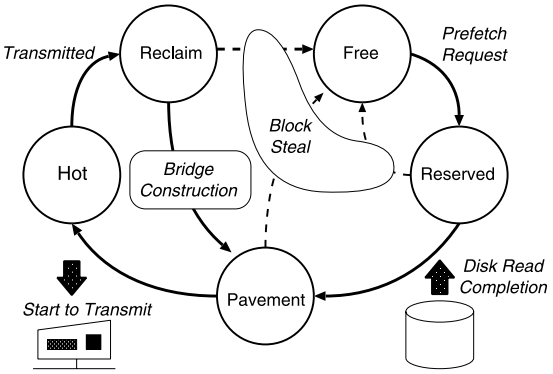

図 4 ブロック状態遷移
Fig. 4 Block state transition.

では配信時にディスク入出力は発生しない．

3.2.2 ブロック状態管理

キャッシュの基本単位であるブロックは**表 1** に示す 5 つの状態（Free，Reserved，Pavement，Hot，Reclaim）を持ち，利用状態によって**図 4** に示す状態遷移を行う．Free ブロックはキャッシュとして有効なデータを持たないブロックである．先読みが必要になれば，Free ブロックを確保して Reserved ブロックとし，ディスクスケジューラに入出力要求を出す．Reserved ブロックはディスク入出力中であることを意味し，まだデータの読み込みが完了していない．

ディスクスケジューラからのディスク入出力完了の通知を受け取ると，Pavement ブロックに遷移する．Pavement ブロックはデータがキャッシュされており，近い将来における参照が予測されることを意味する．Hot ブロックはクライアントに対してパケットをネットワーク配信している状態である．Hot 状態が完了したブロックはすぐに Free ブロックにするのでなく，キャッシュを再利用するために Reclaim ブロックとする．

Free ブロックの供給をスティールと呼ぶ．ブロックにはスティール優先順位があり，優先順位の高い Reclaim，Reserved，Pavement 状態の順にスティールされる．ページングにおけるページ再配置とは異なりブロックは読み込み専用であり，ディスクへの書き戻

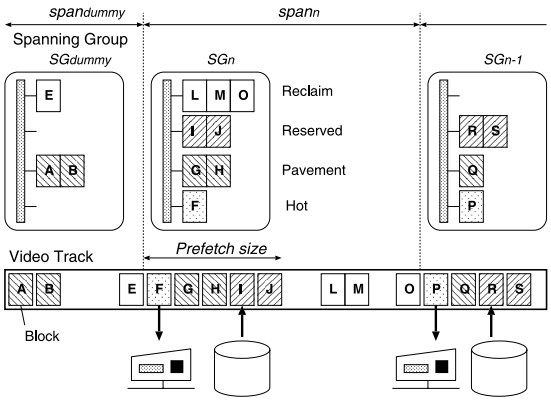

図 5 スパニンググループ
Fig. 5 Concept of the Spanning Group.

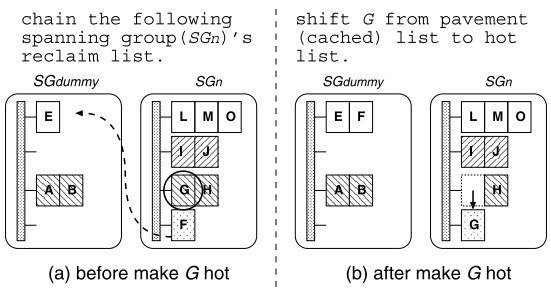

図 6 スパニンググループとブロック管理
Fig. 6 Spanning Group and block management.

しの必要はない．つまりスティールはブロックの状態を変更するだけであり，スティールによる時間的遅延は重大ではない．このため Free ブロックがなくなった時点でスティールを実行する．

SGC によるキャッシュ管理ではどの Reclaim ブロックを Pavement ブロックにするか，Free ブロックが足りない場合にどのブロックをスティールするかの，2 つのアルゴリズムが重要になる．前者をブリッジ構築アルゴリズム，後者をブロックスティールアルゴリズムと呼ぶ．これらの詳細は後述する．

3.3 スパニンググループ構造

SGC ではストリーミングメディアの参照特性を基にしたキャッシュ管理を実現するためにストリーム間のスパンに着目したスパニンググループと名付けたデータ構造を利用する．

スパニンググループはセッションと 1 対 1 に対応するデータ構造であり，各セッションが将来参照すると予測されるブロックを保持する．そして同一コンテンツに対するスパニンググループを関連付けて管理することで，後続するセッションにおける入出力削減を目的とする．

スパニンググループ構造は**図 5** に示すようにスパン，4 つのブロックリスト，先読み範囲を持つ．スパニンググループはセッションと対応し，再生の時間軸順（$SG_1, SG_2, \ldots SG_n$）のリストとして保持する．例外として，リスト先頭に存在するスパニンググループ SG_{dummy} は実際のセッションとは対応せず，次に発生するセッションにブロックを引き継ぐために存在するダミーのスパニンググループである．

ダミー以外のスパニンググループは必ず Hot ブロック，つまりネットワーク送信中のブロックを持つ．スパニンググループはその Hot ブロックから先行する Hot ブロックの直前までの範囲のブロックを管理する．この範囲をスパンと定義する．たとえば $span_n$ 内に存在するブロックはスパニンググループ SG_n が保持する．スパンが短いことはリクエスト到着間隔が短いことを意味する．

スパン内のブロックは各状態ごとに，かつデッドライン順に整列されたリストとして保持する．図 5 では Hot ブロックが $\{F, P\}$，Pavement ブロックが $\{A, B, G, H, Q\}$，Reserved ブロックが $\{I, J, R, S\}$，Reclaim ブロックが $\{E, L, M, O\}$ である．そしてディスク転送が必要なブロックは各 Reserved リストの先頭 $\{I, R\}$ であり，ネットワーク転送が必要なブロックは各 Hot ブロック $\{F, P\}$ であることを示している．

各スパニンググループは先読み範囲を持ち，この範囲を Reserved ブロックにすることで先読みを試みる．そして先読みが完了次第，Pavement ブロックにする．同様にブリッジ構築アルゴリズムに従って Reclaim ブロックを Pavement ブロックにする．

ネットワーク配信の完了時点で，現在の Hot ブロックを Reclaim ブロックにし，続いて再生されるブロックを Hot ブロックにする．さらに Hot ブロックの位置に合わせてスパニンググループの位置も変化するため，Hot ブロックから Reclaim ブロックへの遷移では，現在のスパニンググループではなく，後続するスパニンググループの Reclaim リストにブロックをつなぐ．

たとえば**図 6** に示すようにスパニンググループ SG_n における Hot ブロックが F から G に遷移する場合，まず F を後続するスパニンググループ SG_{dummy} の Reclaim リストにつなぎ（図 6 (a)），次に G を Pavement リストから Hot リストにつなぎ替える（図 6 (b)）．

3.4 アルゴリズム

3.4.1 ブリッジ構築アルゴリズム

ブリッジ構築アルゴリズムは無駄なキャッシュ再利用を回避するために，次に示す方法でスパン内の全

ブロックを Pavement ブロックにし，ブリッジを構築する．

- 先読み範囲内に入った Reclaim ブロックを Pavement ブロックにする
- Hot ブロックが後続のスパニンググループの先読み範囲内の場合は Reclaim リストではなく直接 Pavement リストにつなぐ

またキャッシュサイズは有限であるので，先読みサイズを適切に設定することで，キャッシュを参照頻度に応じて傾斜配分する．すべてのコンテンツのサイズとビットレートが均一であると仮定すると，先読み範囲は次式により決定する．SG_{MAX} は参照頻度が最大のコンテンツに対するスパニンググループ数であり，SG_i はスティール対象となるコンテンツ i に対するスパニンググループ数である．

$$Prefetch\,Size = \frac{Content\,Size}{SG_{MAX}} \times \frac{SG_i}{SG_{MAX}},$$
$$SG_{MAX} = \max(SG_i) \qquad (1 \le i \le N)$$

上記の式は参照頻度に比例して先読み範囲を広げ，参照頻度の最も高いコンテンツにおいては全領域をキャッシュするように先読みすることを意味する．たとえば $ContentSize = 1\,\mathrm{GB}$，$SG_{MAX} = 20$ とすると，上式より $SG_i = 20$ で先読みサイズは 50 MB，$SG_i = 10$ で 25 MB が得られる．

3.4.2 ブロックスティールアルゴリズム

Free ブロックが不足した場合は使用中のブロックをスティールし，再割当てする必要がある．スティール対象はブリッジ構築に関係するブロックを極力スティールしないように選択することが求められ，その基準としてコンテンツの参照頻度を示すスパニンググループ数，リクエスト到着間隔を示すスパン，デッドラインを示すブロック状態を利用する．つまりスパニンググループ数が少ないコンテンツ中で，一番スパンが長いスパニンググループにおける Reclaim リストの最後尾ブロックが最もスティール優先順位が高くなる．

まとめるとブロックスティールアルゴリズムは，次に示すアルゴリズムに従ってスティール対象を決定する．

(1) 参照頻度の低いコンテンツから優先的にスティールする．
(2) スパンが長いスパニンググループから優先的にスティールする．
(3) ブロックのスティール優先順位に従い Reclaim，Reserved，Pavement の順にスティールする．
(4) 同一ブロック状態ではデッドラインの遠いブロックを優先的にスティールする．
(5) Pavement 状態ではブリッジではないブロックを優先的にスティールする．

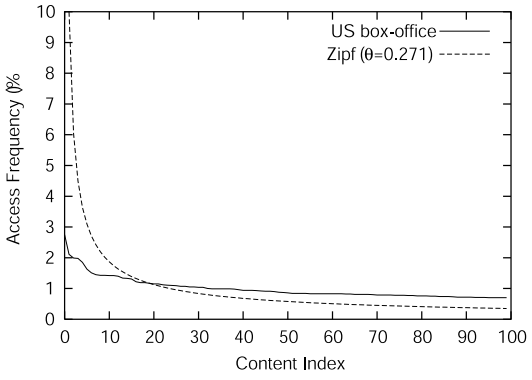

図 7 リクエスト分布
Fig. 7 Popularity distribution.

4. シミュレーションによる効果予測

提案方式の有効性を検証するために，ストリーミングサーバに対する 2 種類の負荷モデルを定義し，シミュレータによる効果予測を行った．本章ではシミュレーション環境と負荷モデルの定義，そして得られた結果と考察について述べる．

4.1 負荷モデル

シミュレータはポアソン分布に従ったリクエスト到着間隔でセッション開始のリクエストを発生させる．各セッションにおいて再生するコンテンツは負荷モデルごとの参照頻度に従い選択されるが，その前後関係に依存性はないものとする．また各セッションはコンテンツの最初から最後まで通常再生するものとし，途中で再生を停止したり，早送り，巻戻しなどの特殊再生を行ったりしないものとする．

コンテンツの参照頻度モデルとして全米映画興行収入トップ 100 [16] より求めた一般分布に従う US モデルと Zipf 近似分布に従う Zipf モデルの 2 種類を使用する．**図 7** に示すように US モデルはトップ 10 付近は指数分布に近いが 10 位以下は一様分布に近い．Zipf 近似分布では，コンテンツを参照頻度順にインデックス付けした場合，i 番目のコンテンツを参照する確率が $1/i^{1-\theta}$ となる．なお θ の範囲は $0 < \theta \le 1$ である．これは人気順位とアクセス数の積が一定になることを意味し，θ が 0 に近付くほど参照が偏ったスキューな分布になり，1 に近付くほど一様分布に近くなる．Dan ら [7] はレンタルビデオの貸出し数は $\theta = 0.271$ の Zipf 近似分布になると報告している．レンタルビデオにおける貸出し傾向はストリーミングサーバに類似すると考えられるため，Zipf モデルでも $\theta = 0.271$ の Zipf 近似分布に従うものとする．

表 2 ストリーミングメディア
Table 2 Streaming media attributes.

フォーマット	MPEG4 シンプルプロファイル
再生時間	120 分
画像サイズ	320×240
ビットレート	142.2 KB/秒（固定）
ファイルサイズ	1 GB

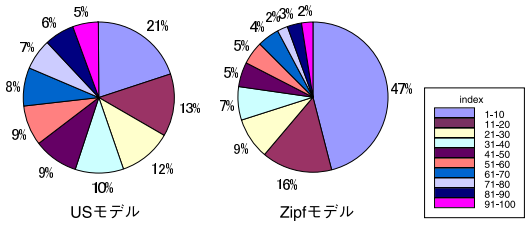

図 8 参照累積分布
Fig. 8 Cumulative distribution of requests.

ストリーミングメディアは**表 2** に示すような MPEG4 シンプルプロファイルのコンテンツを 100 本とする．そして図 7 の分布に従った参照累積分布を**図 8** に示す．円グラフの各領域は参照頻度が高い順に 10 等分されており，Zipf モデルでは上位 10%のコンテンツへの参照が 47%を占めていることが分かる．

4.2 パラメータ

ストリーミングサーバのシミュレーションに対するパラメータを**表 3** のように定義した．キャッシュカバー率（C）は全コンテンツの合計サイズに対してメモリ上にキャッシュ可能な容量の割合である．たとえば 1 GB のコンテンツ 100 本に対してメモリサイズが 4 GB である場合，キャッシュカバー率は 4%となる．ブロックサイズ（B）は IDE ディスクが一度に入出力要求を発行できる最大サイズである 128 KB 固定とした．コンテンツ常駐化率（R）は利用可能なメモリ領域の何パーセントを人気の高いコンテンツに静的に割り当て，常駐化させるかを示している．

そして平均リクエスト到着間隔 $\lambda^{-1} = 9$ 秒の場合を高負荷時，$\lambda^{-1} = 180$ 秒の場合を低負荷時と定義する．高負荷時は平均 800 ストリーム，低負荷時は平均 40 ストリームを並列に処理することになる．

なお高負荷時の条件は提案方式を評価するための極端な数値ではあるが，次に示すとおりシステム構成上は実現可能であると考える．表 2 に示した MPEG4 圧縮されたコンテンツを 800 ストリーム同時にユニキャスト通信により配信するには，約 114 MB/s のネットワークバンド幅が必要であり，ギガビットクラスのネットワークが必要となる．現在のコモディティ回線上でこのようなストリーム配信を行うことは現実的ではないが，LAN や専用回線で接続された WAN では実現可能である．また必要なディスクバンド幅は入出力削減率に依存するが，MAXTOR 6Y200PO 4 台と 3Ware Escalade 7500-4 による RAID0 構成で 120 MB/s の実効バンド幅を測定しており，想定している規模のストリーミングサーバでは複数の装置を多重化することで実現可能である．

表 3 シミュレーションパラメータ
Table 3 Parameters used in simulation.

C	キャッシュカバー率（%）
B	ブロックサイズ（128 KB 固定）
R	コンテンツ常駐化率（%）
λ^{-1}	リクエスト到着間隔の平均値
θ	Zipf 近似分布のパラメータ

表 4 キャッシュヒット率
Table 4 Cache hit ratio.

	US モデル		Zipf モデル	
	LRU	SGC	LRU	SGC
低負荷時	4.5 (1.0)	20.2 (4.5)	12.8 (1.0)	36.4 (2.8)
高負荷時	6.3 (1.0)	28.9 (4.6)	16.7 (1.0)	44.5 (2.7)

単位は%，括弧内は LRU に対する比．

4.3 結果と考察

比較対象として LRU を用い，次の項目に関してシミュレーションによる効果予測を行った．

- キャッシュヒット率の比較
- キャッシュカバー率（C）による影響
- コンテンツ常駐化（R）との併用

4.3.1 提案方式における入出力削減効果

SGC と LRU における入出力削減効果を比較するために，キャッシュヒット率と入出力削減量の累積分布を調べた．以下，断りのない限り C は 4%とする．

各条件でのキャッシュヒット率と LRU のキャッシュヒット率を 1.0 とした場合の SGC の相対比を**表 4** に示す．低負荷時における US モデルにおけるキャッシュヒット率は SGC が 20.2%に対して LRU が 4.5%と約 4.5 倍，Zipf モデルで SGC が 36.4%に対して LRU が 12.8%と約 2.8 倍高い．一方，高負荷時における US モデルにおけるキャッシュヒット率は SGC が 28.9%に対して LRU が 6.3%と約 4.6 倍，Zipf モデルで SGC が 44.5%に対して LRU が 16.7%と約 2.7 倍高い．

両者の差はブロックスティールアルゴリズムの違いによる影響が大きい．LRU では単純に最も最近に参照されたブロックをキャッシュに残すので，近い将来に参照される可能性があるブロックをスティールしてしまい再利用できない．一方，SGC ではデッドラインが遠いブロックを優先してスティールするので，無

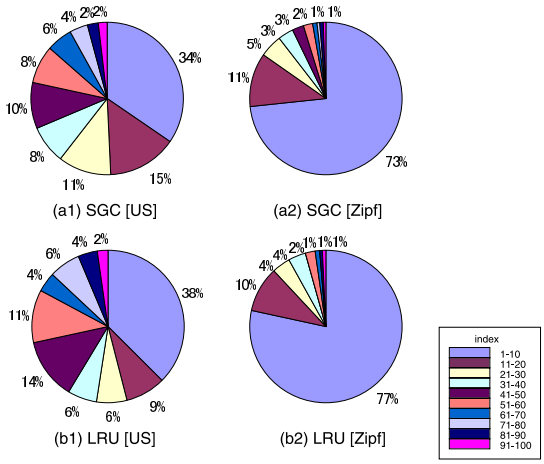

図 9 入出力削減量の累積分布
Fig. 9 Cumulative distribution of reduced disk I/O.

駄なスティールが削減できる．

次にコンテンツの参照頻度とキャッシュヒット率の関係を調べた．低負荷時における全入出力削減量に対して図 7 同様に分割したグループごとに占める入出力削減量の累積分布を**図 9** に示す．図 8 と図 9 の参照累積分布と比較すると，参照頻度が高いコンテンツほど入出力削減量も大きいことが分かる．さらに図 9 (a1), (b1) を比較すると，SGC に対して LRU の方が偏りがやや大きい．これは LRU において参照頻度だけではなくリクエスト到着間隔の影響を受けていることが要因である．一方，SGC では各ストリームの参照頻度に応じてキャッシュを傾斜配分し，かつデッドラインが遠いブロックを優先してスティールするので，参照頻度に応じた入出力削減が可能になったと考えられる．なお高負荷時においても同じ傾向を示す．

4.3.2 入出力削減効果の詳細

前項で述べた SGC における効果の内訳を示すために，スパニンググループ数を基にしたキャッシュの傾斜配分の効果とブリッジ構築によるキャッシュヒット率への影響を調べた．

(1) スパニンググループ数を基にしたキャッシュの傾斜配分

コンテンツの人気度に応じてキャッシュを傾斜配分するためには，コンテンツに対する参照頻度と参照間隔を把握する必要がある．そこで SGC では人気度の代理指標としてスパニンググループ数を利用しているが，この妥当性を評価した．

低負荷時における入出力削減量とスパニンググループ数の最大値，平均値の関係を**図 10**，**図 11** に示す．平均スパニンググループ数 0 は一度も参照されなかったコンテンツであり，1 は参照はされたが，そのセッション中に同一コンテンツに対する参照が発生しなかったことを意味する．

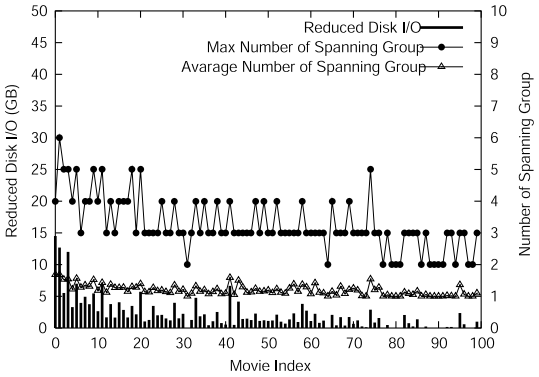

図 10 入出力削減量とスパニンググループ数 [US]
Fig. 10 Reduced disk I/O and number of spanning group [US].

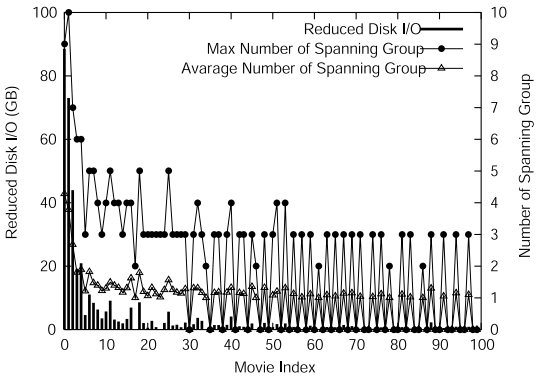

図 11 入出力削減量とスパニンググループ数 [Zipf]
Fig. 11 Reduced disk I/O and number of spanning group [Zipf].

図 10，図 11 よりスパニンググループ数が多い箇所における入出力削減量が多いことが分かる．特に参照頻度が同程度でもスパニンググループ数が多い，つまりリクエスト到着間隔が短く，並列したストリームが存在するコンテンツにおける入出力削減量が多い．これよりコンテンツあたりのスパニンググループ数がコンテンツに対する動的な人気度の代理指標として機能することが分かる．

(2) ブリッジ構築による効果

ブリッジ構築有効時と無効時それぞれのキャッシュヒット率を比較した．なおブリッジ無効は，ブリッジ構築アルゴリズムにおいて先読み範囲のブロックを Pavement 状態に遷移させないことで評価した．

LRU と各条件の SGC におけるキャッシュヒット率，さらに LRU のキャッシュヒット率を 1.0 とした場合

表 5 ブリッジの有無によるキャッシュヒット率
Table 5 Cache hit ratio with and without bridge construction.

	LRU	SGC	
		ブリッジ無効	ブリッジ有効
低負荷時	12.8 (1.0)	36.3 (2.8)	36.4 (2.8)
高負荷時	16.7 (1.0)	29.2 (1.7)	44.5 (2.7)

単位は%，括弧内は LRU に対する比．

の SGC の相対比を**表 5** に示す．ブリッジ構築によってキャッシュヒット率は高負荷時に 15.3%向上している．一方，低負荷時は 0.1%の向上とほとんど効果はみられない．したがって高負荷時にはブリッジ構築により無駄なスティールが削減されており，ブリッジが入出力削減に貢献していることが分かる．高負荷時ほど入出力削減が要求されるので，この傾向はブリッジ構築の有効性を示していると考えられる．

4.3.3 キャッシュカバー率の変化に対する挙動

キャッシュカバー率 C を変化させ，キャッシュヒット率への影響を評価した．US モデルでの結果を**図 12**，Zipf モデルでの結果を**図 13** に示す．

一般的な傾向として C が低いほど LRU に対する SGC のキャッシュヒット率が高く，C が高くなるとその差は小さくなる．たとえば $C = 1\%$での LRU に対する SGC のキャッシュヒット率を調べると，US モデルの場合，低負荷時で 7.8 倍，高負荷時で 12.0 倍，Zipf モデルの場合，低負荷時で 6.0 倍，高負荷時で 3.2 倍とどちらも表 4（$C = 4\%$）で示した改善よりも大きい．

Zipf モデルより US モデルの方が SGC の効果が高いのは，一様分布の方がリクエスト到着間隔が大きくなる傾向にあるので，LRU では C が下がると 2.4 節 (1) に示した問題によってキャッシュヒット率が悪化するが，SGC ではブロックスティールアルゴリズムの工夫によってこの問題を改善しているためであると考える．

4.3.4 コンテンツ常駐化との併用

キャッシュヒット率を上げるために参照頻度の高いコンテンツをメモリに常駐化することはよく使われる手法である．しかし，どのコンテンツを選択するかを前もって見積もることは困難であることが多い．そこで SGC とコンテンツ常駐化を併用する手法を評価した．

コンテンツ常駐化率 R は使用可能なメモリの何パーセントを常駐化に利用するかを示すパラメータである．常駐化はコンテンツ単位で行うこととした．コンテンツサイズは 1 GB なので，R が 25%の場合は，最も参照頻度の高いコンテンツを常駐化させ，残りの 3 GB を SGC で使用することになる．

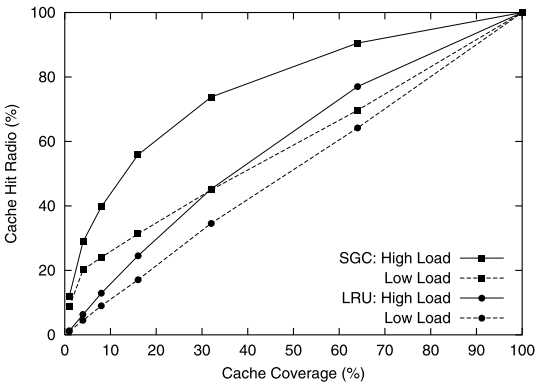

図 12 キャッシュカバー率とキャッシュヒット率 [US]
Fig. 12 Cache coverage vs. cache hit ratio [US].

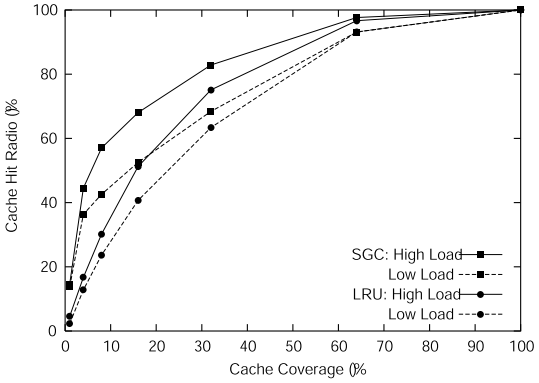

図 13 キャッシュカバー率とキャッシュヒット率 [Zipf]
Fig. 13 Cache coverage vs. cache hit ratio [Zipf].

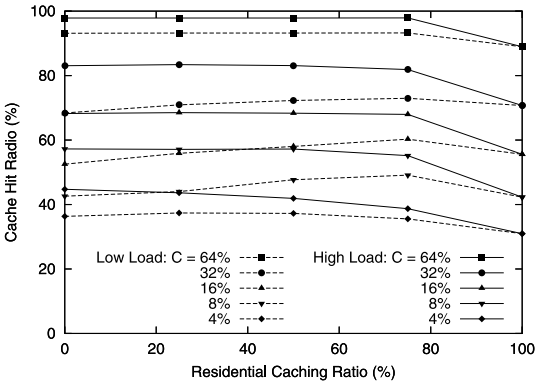

図 14 常駐化と併用時におけるキャッシュヒット率
Fig. 14 Combination with a residential caching.

負荷モデルが Zipf の低負荷時と高負荷時においてキャッシュカバー率 C と R を変えてシミュレーションした結果を**図 14** に示す．低負荷時でかつキャッシュカバー率が低い場合はコンテンツ常駐化と併用することで最大 10%キャッシュヒット率が向上するが，それ

以外の場合は，同じか低下がみられるため，コンテンツ常駐化と併用する効果はない．これは明示的に常駐化するコンテンツを指定しなくても，SGC によって参照頻度の高いコンテンツが暗黙的に常駐化された状態になるためである．また，ストリーミングメディアは 1 つのコンテンツのサイズが大きいため，常駐化によって無駄になるブロックも多い．さらに SGC では 4.3.1 項で示したように，参照頻度の変化に対して動的に対応できるため，常駐化するコンテンツを手動で取捨選択することが不要であるというサーバ運用上の利点がある．

5. 関連研究

ストリーミングサーバや VoD（Video on Demand）サーバによるマルチメディアデータ配信に関して数多くの既存研究が存在するが，特に近年ではプロキシを用いた配信機構の最適化に関する多くの研究がなされている[14),15)]．SGC はストリーミングサーバ内のキャッシュ管理として利用することを想定しているが，配信機構のバッファ管理とも技術的な共通点は多い．

配信機構のバッファ管理としては，バッファを用いて複数リクエストの時間差を解消する batching や patching と呼ばれる手法が研究されている．batching は同一コンテンツに対して時間的に近い複数ストリームをマルチキャスト通信を利用することで単一ストリームにまとめる手法である．しかしサーバがクライアントからの要求を受け取っても，同一コンテンツに対する要求を待つ必要があり，再生開始まで遅延が発生する．そこで patching はすでに配信が開始されたマルチキャストストリームへ合流するために必要な部分だけをユニキャスト通信で送信することで，遅延発生を防いでいる．SGC がストリーミングメディアの参照特性を基にキャッシュを利用してディスク入出力の削減を目指しているのに対し，これらの手法はネットワーク入出力の削減を目的としている．

SOCCER[15)] ではインタネット上に分散したプロキシ間で協調しながら，ストリーミングメディアのセグメントをキャッシュするために，patching を用いてバッファリングの最適化を行っている．ストリーミングサーバに対しても SGC に加えて patching を適用することで，ネットワーク入出力の削減やスパニンググループの管理コストの軽減が期待でき，スケーラビリティの向上が可能になると考える．

また従来よりデータベースシステムなど巨大なファイルを利用するシステムにおいて LRU-K[9)] や 2Q[10)] など LRU を拡張したアルゴリズムの研究が行われてきた．LRU は最も最近のページ参照情報だけを利用するが，これらのアルゴリズムはさらに以前の参照履歴も利用することで参照特性をページ再配置に反映するように改良されている．さらに参照間隔や参照の周期性に着目したアルゴリズムとしては LIRS（Low Interreference Recency Set）[11)] や ICP（Interval Caching Policy）[7),8)] がある．

LIRS は最も最近の参照だけではなく，その 1 つ前の参照との間隔である IRR（Inter-Reference Recency）を指標とするアルゴリズムである．LIRS ではキャッシュを IRR が短い LIR（Low IRR）ブロックと長い HIR（High IRR）ブロックに分類して管理する．そして続く参照も IRR の間隔で起こると仮定し，LIR ブロックを優先的にキャッシュに常駐化する．

LIRS は LRU をベースに IRR をヒント情報として利用するため，各ストリームの周期性を正確に把握することはできず，参照ごとに LIR と HIR の切替えを制御する必要があるが，全ブロックを保持するスタックと常駐化する HIR ブロックを保持するリストにより実現できるので，実装のオーバヘッドが小さいという利点がある．一方，SGC ではストリームとコンテンツの対応をスパニンググループとして保持し，参照の周期性を正確に把握しているので，より無駄の少ないキャッシュ再利用を実現できるが，実装のオーバヘッドに関しては検討の余地が大きい．

ICP は同一コンテンツに対する連続したアクセスに着目したアルゴリズムであり，リクエスト到着間隔が短いストリームを優先的にキャッシュする．ICP がストリームの要求間隔であるインターバル単位でキャッシュ対象を選択しているのに対して，SGC におけるスパンはインターバルと近い考え方であるが，デッドラインが遠いブロックからスティールするというブロック単位のキャッシュ管理を行っており，キャッシュカバー率が低い場合にはキャッシュヒット率の向上に効果があると考える．さらに SGC は先読みを実現するため，ディスクとネットワークに対する入出力を非同期で行う点が異なる．

SEQ[12)] はページ参照履歴からシーケンシャルなアクセスパターンを検出し，仮想記憶管理におけるページ再配置に適用している．つまりページフォールトの発生が連続する部分に対しては MRU（Most Recently Used）を，非連続な部分には LRU を適用する．また Cao ら[13)] はアプリケーションが明示的にヒントを与えることにより，ファイルキャッシュ，先読み，ディスクスケジューリングを制御する手法を提案している．しかし SGC はストリーミングメディアに対する適用

を前提としており，スパニンググループ構造の構築にはシーケンシャルなアクセスパターンの検出や，ユーザレベルのヒントを必要としない．

6. おわりに

本論文ではストリーミングメディアの参照特性に基づく入出力削減方式であるスパニンググループキャッシング方式を提案し，シミュレーションによる有効性の検証を行った．本方式はコンテンツに対する参照頻度に応じてキャッシュを傾斜配分し，かつコンテンツごとにリクエスト到着間隔の近いリクエストをグループ化することで，デッドラインが近いキャッシュを優先的にメモリに残し，入出力削減を実現する．シミュレーション結果より本方式はキャッシュカバー率が 4%の場合，LRU と比較すると高負荷時で 2.7 倍，低負荷時で 2.8 倍の入出力回数を削減できると予測される．

さらに本方式はコンテンツの動的な参照要求に対して自動的に対応可能であり，コンテンツの常駐化対象の選択にも効果がある．したがって本方式がストリーミングサーバにおける入出力削減方式として有効であると考える．

なお現在本機構を実装中であり，今後の課題として実環境下での評価，および既存研究との比較評価があげられる．

参考文献

1) 高野了成，浅見和男，帆波幸二，吉澤康文：ストリーミングデータの参照特性に基づく入出力削減方式の提案，情報処理学会研究会報告，2003-OS-92, pp.61–68 (2003).
2) 浅見和男，帆波幸二，高野了成，吉澤康文：連続メディア配信システム：S^3 におけるメモリ管理機構の開発，情報処理学会第 65 回全国大会，6U-1 (2003).
3) 帆波幸二，浅見和男，高野了成，吉澤康文：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，6U-2 (2003).
4) Tanenbaum, A.S.: Modern Operating Systems, Second Edition, Prentice Hall (2001).
5) Schulzrinne, H., Rao, A. and Lanphier, R.: RFC 2326: Real Time Streaming Protocol (RTSP) (1998).
6) Schulzrinne, H., Casner, S., Fredrick, R. and Jacobson, V.: RFC 1889: RTP: A Transport Protocol for Real-Time Applications (1996).
7) Dan, A. and Sitaram, D.: Buffer Management Policy for an On-Demand Video Server, *IBM Research Report RC 19347* (1994).
8) Dan, A. and Sitaram, D.: A Generalized Interval Caching Policy for Mixed Interactive and Long Video Environments, *Proc. IS&T SPIE Multimedia Computing and Networking Conference*, pp.344–351 (1996).
9) O'Neil, E.J., O'Neil, P.E. and Weikum, G.: The LRU-K Page Replacement Algorithm for Database Disk Buffering, *Proc.1993 ACM SIGMOD Conference*, pp.297–306 (1993).
10) Johnson, T. and Shaha, D.: 2Q: A Low Overhead High Performance Buffer Management Replacement Algorithm, *Proc. 20th International Conference on VLDB*, pp.439–450 (1994).
11) Jiang, S. and Zhang, X.: LIRS: An Efficient Low Inter-reference Recency Set Replacement Policy to Improve Buffer Cache Performance, *Proc. 2002 ACM SIGMETRICS Conference*, pp.31–42 (2002).
12) Glass, G. and Cao, P.: Adaptive Page Replacement Based on Memory Reference Behavior, *Proc. 1997 ACM SIGMETRICS Conference*, pp.115–126 (1997).
13) Cao, P., Felten, E.W., Karlin, A.R. and Li, K.: Implementation and Performance of Integrated Application-Controlled File Caching, Prefetching, and Disk Scheduling, *ACM Trans. Comput. Syst.*, No.14, Vol.4, pp.311–343 (1996).
14) Hua, K.A., Cai, Y. and Sheu, S.: Patching: A multicast technique for true video-on-demand systems, *Proc. ACM Multimedia Conference* (1998).
15) Hofmann, M., Ng, T., Guo, K. and Zhang, P.H.: Caching techniques for streaming multimedia over the Internet, Technical Report GL011345-990409-04TM, Bell Lab. (1999).
16)「全米興行収入ランキング」歴代 Best100. http://yarusou.com/r0mrekidai.htm

(平成 15 年 3 月 31 日受付)
(平成 16 年 2 月 2 日採録)

高野 了成（学生会員）

1997 年東京農工大学工学部電子情報工学科卒業．1999 年同大学大学院工学研究科電子情報工学専攻博士前期課程修了．同年同大学院電子情報工学専攻博士後期課程進学，2003 年 4 月（株）アックスに入社，現在に至る．オペレーティングシステムに興味を持つ．

浅見 和男（正会員）

2001 年東京農工大学工学部電子情報工学科卒業．同年同大学大学院工学研究科電子情報工学専攻博士前期課程進学，2003 年修了．同年日本電気（株）に入社，現在に至る．連続メディア配信処理に興味を持つ．

帆波 幸二（正会員）

2001 年東京農工大学工学部電子情報工学科卒業．同年同大学大学院工学研究科電子情報工学専攻博士前期課程進学，2003 年修了．同年（株）日立製作所に入入社，現在に至る．リアルタイムスケジューリングに興味を持つ．

吉澤 康文（フェロー）

1967 年東京工業大学理工学部応用物理学科卒業．同年（株）日立製作所に入社，中央研究所に勤務．1973 年同社システム開発研究所に勤務．大型計算機用オペレーティングシステムの開発と性能評価，オペレーティングシステムのテスト・デバッギングシステムの開発，ハイエンドサーバ，超並列計算機，実時間システムの研究開発に従事．1995 年東京農工大学工学部情報コミュニケーション工学科教授，現在に至る．工学博士．メディア情報処理，モバイルコンピューティング等の研究に従事．情報処理学会論文賞受賞（1972 年）．著書に『計算機システム性能解析の実際』（共著，オーム社),『オペレーティングシステムの実際』,『IT 革命時代のオペレーティングシステム』（いずれも単著，昭晃堂）がある．ACM，IEEE-CS 各会員．