連続メディア配信システム:S³におけるディスクスケジューリング方式

帆波 幸二† 浅見 和男† 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

連続メディア配信サーバには，大容量のメディアデータを多数のクライアントに配信するため，高速なI/O機能が必要である．また，メディアに課せられた時間制約を満たすためにリアルタイム処理の機能も必要となる．そこで，我々はLinuxをベースに連続メディアの配信に特化させたオペレーティングシステム:S³を開発している．

本報告ではS³の概要と，実装したリアルタイムディスクスケジューラについて述べる．本スケジューラは，最悪の入出力時間を用いた悲観的なスケジューリングを廃止するために，アクセス時間の予測を用いる．予測はシステム起動時に取得したディスクの性能データをもとに，I/Oリクエストに対して動的に見積もり時間を付加することで実現される．

以下，本稿では，2章でS³の概要，3章でディスクスケジューラ，4章でスケジューラの評価について述べ，5章でまとめと考察について述べる．

2. S³の概要

S³は通常のLinuxに，連続メディア配信支援のためのメモリ管理，ディスク管理，ネットワーク管理の3つの管理モジュールを付加した形で構成される．メモリ管理では，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に基づいた新しいメモリ管理手法を提供し，ディスク管理では，見積もりに基づいたリアルタイムディスクスケジューリングを行う．ネットワーク管理はゼロコピープロトコルスタックによるパケット生成時のコピー削減と，周期駆動による割込みの削減を実現する．これらの機能により，配信スループットの向上とI/Oレートの保証を目指す．

3. S3ディスクスケジューラ

S³におけるディスクスケジューラの最大の特徴はディスクアクセスの処理時間を見積もり，スケジューリングに利用することである．リアルタイムシステムにおいてはデッドラインミスを防ぐため，ワーストケースの処理時間を想定してスケジュールを行うのが一般的である．しかし，磁気ディスクの場合，1回のアクセス時間は，数百マイクロ秒から数十ミリ秒までとばらつきが大きく，常にワーストケースを想定すると，スケジュール可能なリクエストの数が大きく減少してしまう．本スケジューラは，リクエストの前後関係からアクセス時間の見積もりを行うことでこの問題を緩和し，リアルタイム性を維持しつつディスクの利用率を向上させるアプローチをとる．

3.1. S³ディスクスケジューラの構成

S³ディスクスケジューラはリクエストインタフェース，ディスクスケジューラ本体，性能測定モジュールから構成される(図1参照)．

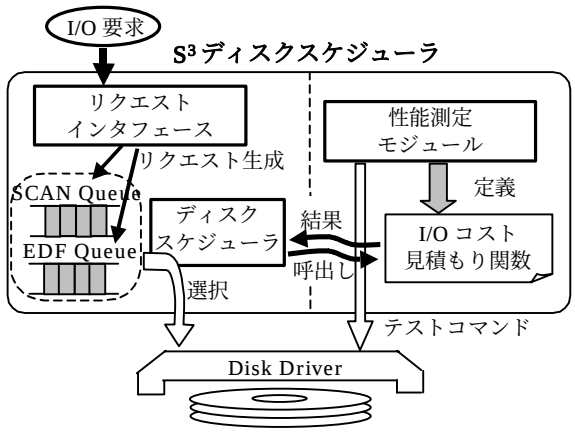

図1 S³ディスクスケジューラの構成

(1)リクエストインタフェース

Linuxのバッファキャッシュからの I/O 要求と，S³メモリ管理からのデッドラインを持つI/O要求を受け付け，リクエストキューを生成する．デッドラインを持つリクエストは，デッドラインの早い順に並べた EDF(Earliest Deadline First)キューに，そうでないリクエストはセクタ順に並べた SCAN キューにそれぞれつながれる．

(2)ディスクスケジューラ

I/O の完了時にたびに実行され，リクエストキューの中から次に実行するリクエストを選択する役目を持つ．スケジューリングアルゴリズムについては次節で述べる．

(3)性能測定ルーチン

システム起動時に実行され，磁気ディスクの性能情報を取得し，アクセス時間の見積もり関数の定義を行う．見積もり関数は，シーク時間，回転待ち時間，転送時間の3つについて定義される．

Disk Scheduling Using Access Time Estimates in S³ System
Koji Honami†, Kazuo Asami†, Ryousei Takano† and Yasufumi Yoshizawa‡
†Graduate School of Engineering, Tokyo University of Agriculture and Technology
‡Faculty of Engineering, Tokyo University of Agriculture and Technology

3.2. スケジューリングアルゴリズム

本スケジューラは，EDF キューの先頭から一つずつリクエストを取り出し，C-LOOK[1]の並び順になるようにスケジュールキューを生成するアルゴリズムを採用している．デッドラインミスを起こさない範囲でスケジュールキューの長さを最大にすることで，見積りによって得られたスケジューリングにおける余裕をシーク最適化に割り当てることを実現している．アルゴリズムのフローを図 2 に示す．

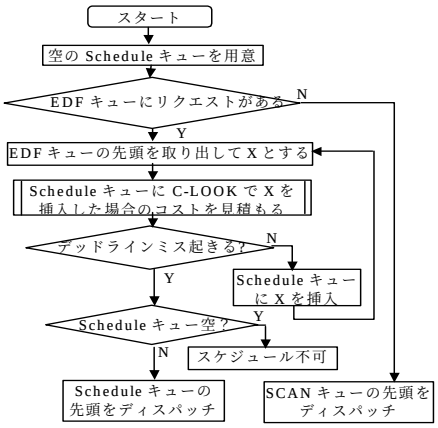

図 2 スケジューリングアルゴリズム

3.3. アクセス時間の見積もり

ここでは，ディスクアクセス時間の内でもっとも大きな割合を占めるシーク時間の見積もり方法について述べる．シーク時間は，シークの移動距離に対する関数で示すことができる．そこで，システム起動時にシークの移動距離とシーク時間の対応を測定し，表を作成しておくことで，コストの見積もりを行っている．

実際に磁気ディスク(表 1 参照)に SEEK コマンドを発行して測定した結果を図 3 に示す．また，アクチュエータの加速度が一定であると仮定して導き出される見積もり関数を図 3，4 に示す．

表 1 ディスクのスペック

Fujitsu MPE3084AE	
Formated Capacity	8.45GB
Disks / Heads	1 / 2
Bytes/Sector	512
Logical blocks	16,514,064
Drive Interface	Ultra EIDE (ATA/66)
Average Seek Time	9.5ms(typ)
Record Density	352,728bpi

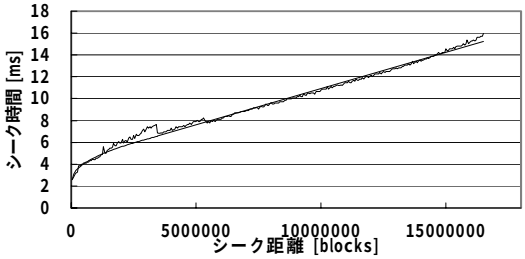

図 3 シーク距離とシーク時間の関係

$$seektime(x)[ms] = \begin{cases} 0 & if \quad x = 0 \\ 2.6 + 3.5 \times 10^{-3}\sqrt{x} & if \quad x \le 3.4 \times 10^{6} \\ 6.5 + 7.3 \times 10^{-7}(x - 3.4 \times 10^{6}) & if \quad x > 3.4 \times 10^{6} \end{cases}$$

図 4 シーク時間の見積もり関数

4. スケジューラの評価

本スケジューラの有効性を示すために，ワーストケースを想定した場合との比較に加え，Linux で採用されている C-LOOK，リアルタイムスケジューリングアルゴリズムである SCAN-EDF[2]との比較も行った．実験は，仮想クライアントによりサーバに負荷をかけ，クライアント数とデッドラインミスの関係を測定した．その結果を図 5 に示す．

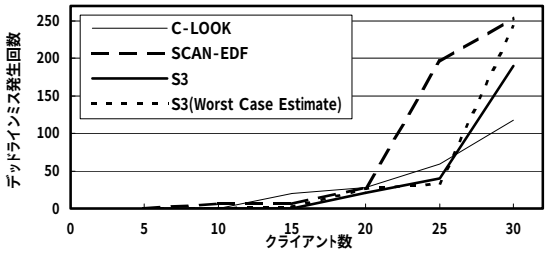

図 5 アルゴリズムの比較

この結果から，本スケジューラは，デッドラインミスを起こさずにサポートできるクライアント数が SCAN-EDF 同様に C-LOOK よりも多くすることが可能で，なおかつ，SCAN-EDF よりもデッドラインミスの発生回数を小さく抑えていることがわかる．このことから，本スケジューラがリアルタイムスケジューラとして有効であることが確認できる．また，ワーストケースを想定した場合よりも 18%スケジュール可能性を向上させるという結果が得られた．

本スケジューラの欠点として，デッドラインミスが発生し始めると，ドミノ効果でデッドライン発生件数が増えてしまうという点が挙げられる．これは，負荷が大きくなるとスケジューラの挙動が EDF に近づくことに起因する．この問題はアドミッション制御を行うことで回避できると考える．

5. おわりに

S^3 のディスクスケジューラについて述べた．アクセス時間の見積もりを用いることにより，デッドラインミスの発生を防ぎ，ワーストケースを想定した場合よりスケジュール可能性を 18%向上させることを示した．これらの結果からりリアルタイムディスクスケジューリングで見積もりを用いることの有効性を示すことができた．

参考文献

[1] Abraham Silberschatz, Peter B. Galvin, "Operating system concepts," Addison Wesley, 4th ed., 1994.
[2] A. L. N. Reddy and J. Wyllie, "Disk Scheduling in Multimedia I/O System," in Proceeding of ACM Multimedia'93, (Anaheim, CA), pp. 225-234, August 1993.