マルチメディアアプリケーションに適した仮想記憶管理手法

五條 多紀子 † 瀧本 栄二 † 芝 公仁 †† 毛利 公一 ††† 大久保 英嗣 †††
† 立命館大学大学院理工学研究科 †† 龍谷大学理工学部 ††† 立命館大学理工学部

1 はじめに

連続メディアデータを扱うマルチメディアアプリケーションは, 時間の正当性に依存した処理を行う. これを実現する 1 つの方法に, メモリ管理において, ページフォルトの発生を削減し, マルチメディアアプリケーションの処理遅延を削減する方法がある. ページフォルトを発生させないためには, マルチメディアアプリケーションが用いるページをすべて物理メモリ上に固定する方法が考えられる. しかし, この方法を用いると, 物理メモリを占有し, 他のアプリケーションが実行できなくなるという問題が生じる. したがって, マルチメディアアプリケーションの実行に必要なページを選択し, 物理メモリ上に固定する必要がある.

連続メディアデータを扱うマルチメディアアプリケーションは, デコード処理と再生処理を繰り返すため, 特定のページに集中してアクセスする. したがって, アプリケーションのページアクセス頻度を計測し, その変化率から, 適応的に物理メモリ上に固定するページを選択することで, 常にアクセスされるページのみを固定することが可能となる. これにより, 他のアプリケーションの実行を妨げることなく, マルチメディアアプリケーションの処理の効率化が可能となる.

2 マルチメディアアプリケーションに適したページ管理手法

2.1 マルチメディアアプリケーションのメモリアクセスパターン

マルチメディアアプリケーションは, コード部分とデータ部分から構成される. 連続メディアデータを扱うマルチメディアアプリケーションは, デコード処理と再生処理を繰り返すため, コード部分は, アプリケーションの開始から終了まで, 常に同じページをアクセスする. データ部分は, リングバッファのように一旦確保したバッファを再利用するアプリケーションの場合, コード部分と同様, 常に同じページをアクセスする. 一方, バッファ領域を拡張するアプリケーションの場合, 時間によって異なるページをアクセスする.

2.2 ページ管理手法の概要

本手法では, アプリケーションのデコード処理と再生処理の実行ごとにページアクセス頻度を計測し, 過去数回のページアクセス頻度の履歴から, 各ページのアクセス頻度の変化率を算出する. ページアクセス頻度が増加または一定傾向のページは, ページアウトの対象外とし, 物理メモリ上に固定する. 減少傾向のページは, 物理メモリ上に固定するが, システム全体のメモリの使用状況に応じて, ページアウトの候補とする. これにより, アプリケーションの動的なメモリアクセスに対して, 適応的に物理メモリ上に固定するページを決定し, アプリケーションが常にアクセスするページのみを物理メモリ上に固定することが可能となる. したがって, アプリケーションの実行中におけるページフォルトの削減が可能となり, マルチメディアアプリケーションの円滑な処理が可能となる.

2.3 ページ管理機構の実装

2.3.1 ページの状態

本手法では, 物理メモリ上のページを以下の 4 段階に分けて管理する. それぞれのページは, 図 1 に示す状態を遷移する.

FIXED ページアウトの対象外である.

SECOND_FIXED ページアウトの対象外とし, 物理メモリ上に固定する. ただし, 新たにページが必要となり, NON_FIXED に該当するページがない場合は, ページアウト候補となる.

NON_FIXED ページアウト候補である.

PG_FREE 物理メモリの空きページである.

従来は, マルチメディアアプリケーションがアクセスするページは, すべて物理メモリ上に固定していた. そのため, 物理メモリ内の空きページがなくなり, NON_FIXED のページもなくなると, 新たにアプリケーションを実行することが不可能となり, システム全体のメモリの使用効率が低下するという問題があった. 本手法では, SECOND_FIXED というページ状態を追加することで, システム全体のメモリの使用状況に応じて適応的にページアウトするページを選択することが可能となる.

A Virtual Memory Management Scheme for Multimedia Applications

Takiko Gojo †, Eiji Takimoto †, Masahito Shiba ††, Koichi Mouri †††, and Eiji Okubo †††
†Graduate School of Science and Engineering, Ritsumeikan University
††Faculty of Science and Technology, Ryukoku University
†††Faculty of Science and Engineering, Ritsumeikan University

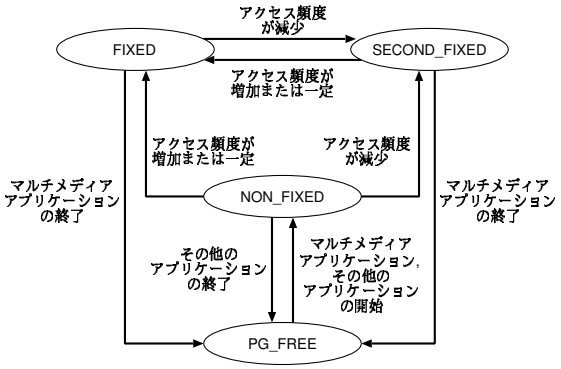

図 1 ページの状態遷移

2.3.2 ページの状態遷移の流れ

図 2 に, 本手法の処理の流れを示す. アプリケーションがデコード処理を開始すると, カーネルは, タイマ割込み毎にページアクセスチェックを行う. アクセスのあったページは, ページアクセス頻度をインクリメントする. 1 フレームのデコード処理と再生処理が終了するまで, ページアクセス頻度を計測し, ページアクセス頻度の変化率を求める.

アプリケーションの実行開始時にアクセスするページは, 一旦すべて NON_FIXED に設定する. 1 回目のデコード処理と再生処理におけるページのアクセス頻度と, 2 回目のアクセス頻度を比較し, 増加または一定傾向にあるページは, FIXED に遷移する. アクセス頻度が減少傾向にあるページは, SECOND_FIXED に遷移する. 以降, 同様に, カーネルは, アクセス頻度の変化率をもとに, 各ページをそれぞれ適切な状態に遷移させる.

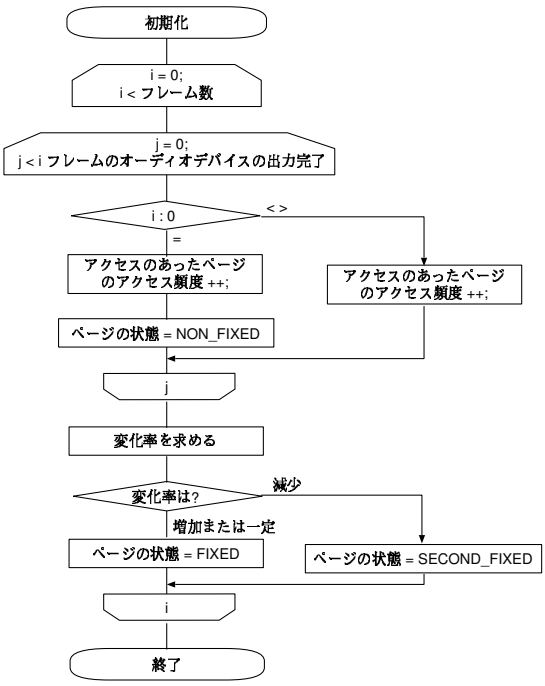

図 2 ページ状態遷移機構

3 動作例

CPU Intel Celeron 1.3GHz, メモリ 256MB, カーネル Linux 2.4.18 を搭載した PC/AT 互換機を用い, mpg123 を用いて, 再生時間 4 分 45 秒, ファイルサイズ 5.5MB の MP3 データを再生し, その際のページアクセスの様子を観測した. mpg123 は, 一旦確保したバッファを再利用する方式を用いるため, コード部分もデータ部分も, MP3 データの再生が終了するまで同じページのみアクセスする. したがって, 本手法を用いると, アクセスしたページはすべて FIXED に遷移し, 固定するページサイズは 184KB となり, メモリ 32MB の場合, 固定する割合は, 0.56% である. これより, 一旦確保したバッファを再利用するアプリケーションは, 再生するデータが大きい場合でも, 数百 KB のページが固定されていれば, ページフォルトを発生することなく円滑に実行することが可能であるといえる. したがって, 本手法を用いた場合, マルチメディアアプリケーションの実行に必要なページの固定を最小限に抑えることが可能となり, 他のアプリケーションへの影響は少ないといえる.

また, mpg123 は, オーディオデバイスの出力バッファが満杯になった場合, 出力バッファに空きが生じるまで待ち状態が長く続く. 今回の実験では, 約 1.5 秒間の待ち状態が, 10 秒ごとに発生している. LRU や, 類似のページ置換えアルゴリズムでは, 待ち状態になる前のページを今後アクセスされないページと判断し, 当該ページをページアウトする可能性がある. 本手法では, デコード処理と再生処理中のページアクセスチェックを行うため, マルチメディアアプリケーションが長い待ち状態に遷移した際も, それ以前にアクセスされたページがページアウトされることはない.

4 おわりに

本稿では, 連続メディアデータを扱うマルチメディアアプリケーションのメモリアクセス特性について述べ, それを考慮したページ管理手法について述べた. アプリケーションのページアクセス頻度と変化率から, 適応的に物理メモリ上に固定するページを選択することで, 常にアクセスされるページのみを固定することが可能となる. これにより, マルチメディアアプリケーションのページフォルトの削減が可能となり, マルチメディアアプリケーションの円滑な処理が可能となる. 今後は, 本手法のリアルタイム OS Easel [1] 上への実装と評価を行う予定である.

参考文献

[1] 谷出新: “リアルタイムオペレーティングシステム Easel におけるメモリ管理機構,” 情報処理学会研究報告 2001-OS-86, Vol 2001, No.21, pp. 91-98 (2001).