B-018

システムコール発行頻度に基づくバッファキャッシュ制御法における重要度更新の遅延方式

Delay Method of Updating File Importance on I/O Buffer Cache Mechanism Based on Frequency of System Call

寺岡 明彦 † 山内 利宏 † 谷口 秀夫 †
Akihiko Teraoka Toshihiro Yamauchi Hideo Taniguchi

1. はじめに

外部記憶装置との入出力処理の速度が低速であることから，入出力の回数を削減するために，バッファキャッシュが利用されている．バッファキャッシュは，ブロックと呼ばれる固定長の単位でデータを管理する．このため，キャッシュヒット率を向上させるためには，ブロック置き換え規則が重要である．ブロック置き換え規則の代表的な方式である LRU 方式では，1 つの大きなファイルへの入出力により，他のファイルがバッファキャッシュから解放されてしまい，キャッシュヒット率が低下するなどの問題が指摘されている [1].

そこで，著者らは，ファイル操作のシステムコール発行頻度に基づくバッファキャッシュ制御法（Frequency of File Usage 方式，FFU 方式）を提案した [1]．また，重要度更新の契機を設定する手法を提案した [2]．文献 [2] の設定手法では，特定のアクセスパターンで性能を向上できる．さらに本稿では，重要度更新処理を遅延させ，特定のアクセスパターンにおいて，さらに性能を向上させる方式を提案する．

2. ファイル操作のシステムコール発行頻度に基づくバッファキャッシュ制御法

2.1 基本方式

FFU 方式を図 1 に示し，説明する．FFU 方式は，ファイルを操作するシステムコールの発行を契機として，ファイルを操作した情報（以降，ファイル操作情報）を取得する．また，一定の周期でファイル操作情報を集計し，ファイルの重要度を算出する．この重要度に基づき，2 レベルに分割したバッファキャッシュを制御する．また，FFU 方式では，バッファキャッシュのために確保された領域を重要と判断されたファイルを構成するブロック（重要ブロック群）を格納するキューと，重要と判断されなかったファイルを構成するブロック（通常ブロック群）を格納するキューで管理する．

ファイル操作時の処理流れを以下に述べる．

(1) OPEN と CLOSE のシステムコールからファイル操作情報を取得し，ファイル情報表に格納する．
(2) 重要度更新処理を行うか否か判断する．重要度更新の契機であれば重要度更新処理を行い，契機でなければ処理を終了する．
(3) 重要度更新処理を行う場合，ファイル操作情報から重要度を算出し重要度表を作成する．次に，重要度更新処理後に，新たに重要と判断したファイルを構成するブロックを通常ブロック群から重要ブロック群へ移動する．また，重要でなくなったファイルを構成するブロックを重要ブロック群から通常ブロック群へ移動する．

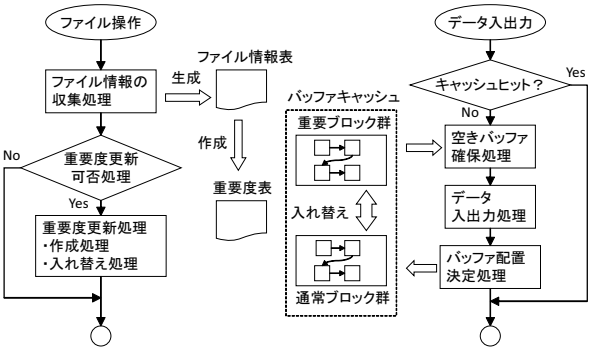

図 1 FFU 方式

また，データ入出力時の処理流れを以下に述べる．

(1) READ，または WRITE システムコールを契機として，要求されたデータがキャッシュヒットしたか否かを調べる．
(2) キャッシュミスした場合，バッファキャッシュから空きバッファを確保する．
(3) データ入出力処理を行う．
(4) データ入出力処理でアクセスしたブロックを，そのブロックが構成するファイルが重要であれば重要ブロック群に，重要でなければ通常ブロック群に格納する．

2.2 重要度更新の契機と処理の開始

ファイルが集中的に OPEN されているか否かという状態に着目した重要度更新の契機を設定する手法 [2] について，以下に説明する．

OPEN されるファイルの状態を，閾値 P を用いて集中状態と非集中状態に分類する．このため，OPEN 間隔を導入する．OPEN 間隔とは，着目したファイルについて OPEN から次の OPEN までに発生した OPEN の総数である．また，閾値 P は OPEN 間隔の中央値であり，「多くのファイルは，P 回ごとに OPEN される」ことを意味する．集中状態のファイルとは，OPEN 間隔が閾値 P 以下のファイルであり，非集中状態のファイルとは，OPEN 間隔が閾値 P より大きいファイルと定義する．このとき，ファイルが集中状態から非集中状態へ，または非集中状態から集中状態へ遷移することを状態変化と呼び，状態変化したファイルの数を状態変化数と呼ぶ．

ファイルを OPEN するごとに，ファイルの状態の更新処理を行い，ファイルの状態を更新する．状態変化

† 岡山大学大学院自然科学研究科，Graduate School of Natural Science and Technology, Okayama University

表 1 評価におけるパラメータ設定

項目	カーネル make 処理	Web サーバ処理
ファイル情報表の上限数	65,536 個	
重み	0.5	
ファイルサイズ制限	2 MB	
保護ファイル数	548 個	1,757 個
閾値 P	83	260
閾値 R	2,075	520

数が閾値 R を超えたときを重要度更新の契機とし，直ぐに処理を開始する．

2.3 重要度更新の契機と処理開始の時期

文献 [2] の手法は，重要度更新の契機が来ると直ぐに処理を開始する．しかし，次の場合は問題になる．集中的にアクセスされるファイル群が切り替わる前後で重要度更新の契機を迎えると，以前にアクセスされていたファイルの重要度が高く設定されたままであり，今後アクセスされるファイルの重要度は相対的に低いままである．このため，今後アクセスされるファイルについてキャッシュミスする確率が高まり，性能が向上しない．この場合の例として，カーネル make 処理のように，集中的にアクセスされるファイル群が明確に切り替わる処理がある．

3. 重要度更新の処理を遅延させる方式

3.1 基本方式

ここでは，重要度更新の処理を遅延させる方式を提案する．提案方式では，任意の正の整数 N を用いて，（$N \times$ 閾値 P）で遅延期間を算出し，重要度更新の契機から遅延期間分だけファイルが OPEN された後，重要度更新の処理を実行する．

提案方式において，閾値 P を基準に遅延期間を設定するのは，閾値 P は OPEN 間隔の中央値であり，頻繁に OPEN されるファイルの OPEN 間隔は閾値 P よりも小さいためである．このことから，遅延期間を閾値 P の定数倍に設定することで，十分なファイル操作情報を収集できると考えられる．

3.2 評価

3.2.1 評価方法と評価環境

評価対象 AP には，2.3 節で述べたアクセスパターンを持つ処理としてカーネル make 処理を，2.3 節で述べたアクセスパターンを持たない処理として Web サーバ処理を用いる．Web サーバ処理では，岡山大学の Web サーバへのリクエストパターンから抽出した 100,000 回のリクエストに対する応答時間について評価を行なった．

評価では，2 台の計算機（CPU: Celeron 430（2.4 GHz），メモリ: 64 MB，OS: FreeBSD 6.3-RELEASE，バッファキャッシュサイズ: 12 MB，VMIO: 無効，バッファの大きさ: 16 KB）を利用した．また，評価時のパラメータは，表 1 に示す値を設定した．

3.2.2 考察

図 2 にカーネル make 処理の実行時間を示す．図 2 より，遅延期間が 83 のとき実行時間が約 18.8 秒（約 2.0%）減少している．これは，提案方式の導入により，2.3 節で述べた問題が解消されたためであると推察できる．また，遅延期間を 249 と 415 に設定したときには，

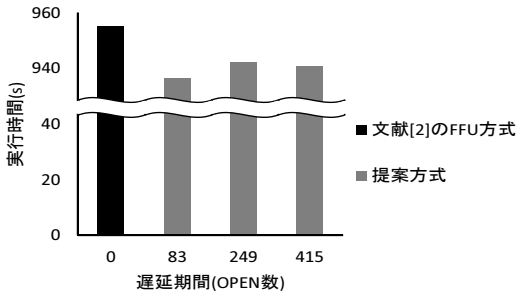

図 2 カーネル make 処理

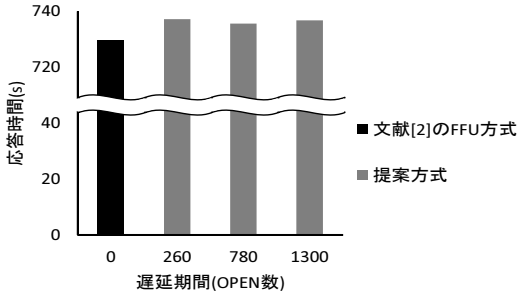

図 3 Web サーバ処理

83 に設定したときと比べ，実行時間の減少量は少なくなっている．これは，遅延期間を長く設定すると，集中的にアクセスされるファイル群が遅延期間よりも短い期間で切り替わる場合に適切なファイルの重要度が算出できないためであると推測する．

次に，図 3 に Web サーバ処理の応答時間を示す．図 3 より，応答時間が最大で約 7.7 秒（約 1.1%）増大している．これは，本評価で用いた Web サーバ処理は 2.3 節で述べたようなアクセスパターンを持たないためである．このため，Web サーバ処理に限らず, 2.3 節で述べたようなアクセスパターンを含まない処理には，効果がないと推察できる．

以上のことから，提案方式は，カーネル make 処理のような，アクセスするファイル群が処理とともに変わる場合に有効であると推察できる．なお，提案方式と既存の LRU 方式を比較した場合，カーネル make 処理と Web サーバ処理ともに提案方式の方が短い処理時間を示すため，FFU 方式の有効性は保たれている．

4. おわりに

重要度更新契機における処理開始を遅延させることにより，特定のアクセスパターンにおいて FFU 方式の性能を向上させる方式について述べた．評価より，提案方式の導入によってカーネル make 処理の実行時間が最大で約 18.8 秒減少することを示した．

残された課題として，適切な遅延期間の設定手法の検討がある．

参考文献

[1] 片上達也，田端利宏，谷口秀夫：ファイル操作のシステムコール発行頻度に基づくバッファキャッシュ制御法の提案，情報処理学会論文誌コンピューティングシステム (ACS)，Vol.3，No.1，pp.50–60 (2010).

[2] 山内利宏，横山和俊，乃村能成，谷口秀夫，芦塚正雄：ファイル操作のシステムコール発行頻度に基づくバッファキャッシュ制御法における重要度更新契機の設定法，情報処理学会論文誌，Vol.56，No.6，pp.1451–1462 (2015).