2G-06

SSDにおけるRAM Buffer管理アルゴリズムの動的チューニング手法

小川 愛理† 吉瀬 謙二†

†東京工業大学 大学院情報理工学研究科 ‡東京工業大学 情報理工学院

1 はじめに

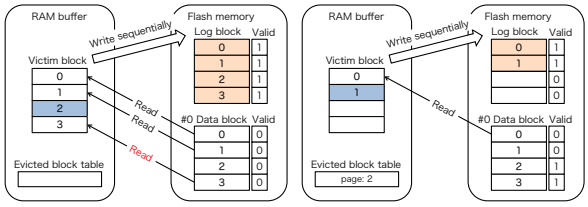
近年，ストレージシステムとして従来の Hard Disk Drive (HDD) に替わって Solid State Drive (SSD) が多く利用されている．SSD は HDD に比べて消費電力が少なく，読み書きが高速であるなど多くのメリットを持つ一方で，書き込み耐久性が低いというデメリットが存在する．

これを軽減するために，我々は SSD 内の書き込みキャッシュとして機能する，大容量の RAM Buffer を効率よく扱う手法 [1] を提案している．提案した RAM Buffer 管理アルゴリズムは，耐久性に悪影響を及ぼすブロック消去の回数を大幅に削減することが可能である．しかし，提案した RAM Buffer 管理アルゴリズムには性能を左右する閾値が存在し，実行するアプリケーションによって最適な閾値は異なる．閾値を求める最もシンプルな手法として，アプリケーション毎にすべての閾値で性能を計測し，その中で最も良い閾値を静的に調べる手法がある．この手法は最適な閾値を一意に選ぶ事ができる一方で，動かすアプリケーションが固定されていない場合には非常に効率が悪い．

そこで本稿では，上記の閾値を動的にチューニングする手法を提案する．この手法では，RAM Buffer を動かしながら自動で最適な閾値に収束するため，静的手法と異なりアプリケーション毎に性能を計測する手間を削減することができる．静的手法と提案手法との耐久性をシミュレーションにより評価し，提案手法が効果的である事を示す．

2 背景と先行研究

多くの SSD は性能と耐久性の向上のために大容量の RAM Buffer を内部に持つ．ホストシステムから SSD への書き込み，読み出し要求はまず RAM Buffer が受け取り，そこからデータが追い出されたり読み出しミスをした場合には，アドレスマッピング機構での論理物理アドレス変換を経て記憶領域であるフラッシュメモリへと要求が行く．フラッシュメモリは，4KB 程度のページと，ページを 128 個程度まとめたブロックで構成される．フラッシュメモリは，読み書きをページ単位で行うことができるが，データの上書きはできないという特性を持つ．そのため，上書きする際には一度ブロック単位でデータを消去し再度データを書き込む必要があり，これには膨大な時間がかかる．また，フラッシュメモリのブロック消去可能回数は限られているため，消去回数を減らすことは耐久性の向上につながる．



(a) オリジナルのページパディング手法　(b) 提案したページパディング手法

図 1: 提案した RAM Buffer 管理アルゴリズム

ブロック消去を削減する RAM Buffer 管理アルゴリズムとして，我々は効率的なページパディング手法を提案している [1]．ページパディングとは，RAM Buffer にないデータをあえて一度フラッシュメモリから読み出し，RAM Buffer に元々あったデータと合わせてもう一度フラッシュメモリに書き戻す手法である．図 1 (a) はオリジナルのページパディングの例である．この例では，1 つのブロックは 4 つのページから成る．図の中の 1 つの四角は 1 つのページを表し，番号はページ番号を示す．フラッシュメモリは上書きができないため，RAM Buffer からフラッシュメモリへの書き込みは一度 log block という名前の空のブロックに書き込まれる．例えば，RAM buffer の追い出しブロックがページ番号 2 のデータしか持っていなかった場合 (図 1 (a) の青いブロック)，フラッシュメモリからページ番号 0, 1, 3 のデータを読み出した上でページ番号 0 から 3 のデータをすべて log block に書き戻す．一方で，図 1 (b) は我々が提案したページパディング手法で，追い出しの際に RAM Buffer が持っている一番大きいページ番号までのデータしかページパディングを行わず，後続のデータが RAM Buffer に入ってくるのを待つという手法である．例えば，RAM buffer の追い出しブロックがページ番号 1 のデータしか持っていなかった場合 (図 1 (b) の青いブロック)，フラッシュメモリからページ番号 0 のデータのみを読み出し，ページ番号 0 と 1 のデータのみを log block に書き戻す．そして，追い出したデータの数を Evicted block table という表に記録する．この場合 0 と 1 の 2 つのページを追い出したので 2 という情報が表に記録される．Evicted block table に保存されているデータよりも大きい番号のページが RAM Buffer に来た場合，この手法はオリジナルのページパディングに比べてブロック消去の回数を削減することができる．

図 1 (a) のようなオリジナルのページパディングを用いるか，図 1 (b) の提案したページパディングを用

A Dynamic Tuning Method for A RAM Buffer Management Algorithm of SSDs

Eri OGAWA†and Kenji KISE‡

†Graduate School of Information Science and Engineering

‡School of Computing, Tokyo Institute of Technology

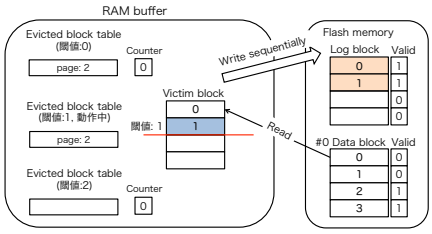


図 2: 動的チューニング手法の例

いるかは，追い出しの際に RAM Buffer が持つ一番大きいページの番号と閾値によって決定される．例えば閾値が 1 である場合，RAM Buffer が持つ一番大きいページの番号が 1 より大きい場合には図 1 (a) のようなオリジナル手法を，1 以下である場合には図 1 (b) のような提案したページパディング手法を用いる．閾値は提案手法の性能に大きな影響を与える上，アプリケーションによって最適な閾値は異なる．

3 提案手法

RAM Buffer を動作させながら閾値をチューニングするために，現在動作中の閾値の値の前後の値における性能を予測する手法を提案する．例えば閾値 1 で RAM Buffer が動作している場合，閾値が 0 と 2 の時の性能を予測し，一定時間後により良い閾値の値に変更する．これを繰り返す事で最終的に最適な閾値付近に収束する．

閾値の効果を正しく得るためには RAM Buffer 内のすべてのブロックを複製する必要があるが，これには多くの RAM リソースが必要になる．そのため，提案手法では Evicted block table のみを複製する事で簡易的な性能予測を行う．図 2 は，提案手法の例である．動作中の閾値は 1 であると仮定する．この時，閾値が 0 と 2 の時の性能予測用を含め 3 つの evicted block table が存在する．RAM Buffer が持つ一番大きいページの番号が 1 の時，閾値は 1 であるため提案したページパディング手法が適用される．そのため，ページ番号 0 と 1 のみがフラッシュメモリに書き込まれ，Evicted block table にはページ数 2 が保存される．もし閾値が 0 であった場合にも，提案したページパディング手法が適用されるため，表には同様にページ数 2 が保存されるはずである．しかし，もし閾値が 2 であった場合にはオリジナルのページパディングが適用されるため，Evicted block table には何も書き込まれない．前述の通り，Evicted block table に保存されている番号以上のページ番号を持つデータが RAM Buffer に来た場合にはブロック消去が削減できるため，そのようなデータが来た数を性能予測中の閾値毎にカウントする．例えば，図 2 の後にページ番号 2 のデータが来た場合には，閾値 0 のカウンタがインクリメントされる．一定時間後，このカウンタの数が大きい方の閾値に動作中の閾値が変化する．

3.1 評価

提案手法と静的手法との比較評価を行う．評価は FlashSim [2] というイベント駆動の SSD シミュレータを用い，ページサイズ 4 KB, ブロックサイズ 512 KB の 32GB の SSD をシミュレートした．評価には [3] と [4] から 7 つの I/O トレースを用いた．RAM Buffer は 32MB とし，既存研究 BPLRU+[1] をベースに閾値が 1.0 の時を BASELINE, 静的に最適な閾値を計測したものを STATIC, 提案手法の動的チューニング手法を DYNAMIC としてブロック消去の回数を計測した．

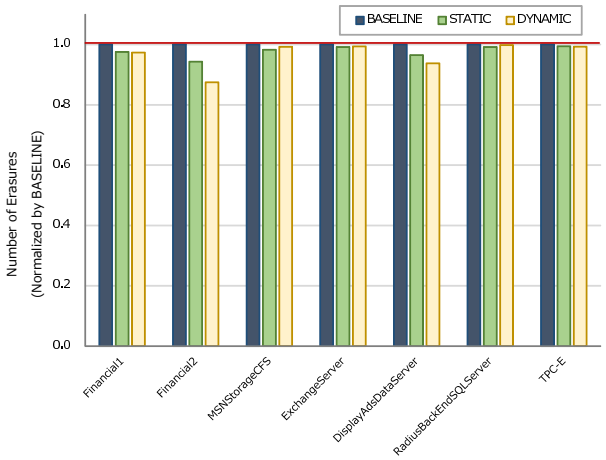


図 3: アプリケーション毎のブロック消去回数

図 3 に評価結果を示す．消去回数は BASELINE を 1 として正規化している．消去回数が少ないほど耐久性が高くなるのでグラフの値は低いほど良い．グラフより，幾つかのアプリケーションにおいて DYNAMIC が STATIC よりもブロック消去が少なくなっており，最大で 8% 程度消去回数を削減している．1 つのアプリケーションを動かす中で最適な閾値が変化していく場合には，DYNAMIC の方が性能が良くなると考えられる．MSNStorageCFS や RadiusBackEndSQLServer などでは DYNAMIC の方が消去回数が増えてしまっているものの，最大でも 1% 程度の増加に止まった．1 つのアプリケーションを動かす中で最適な閾値が変わらない場合には，最適な閾値に収束するのに時間がかかる分 DYNAMIC の方が多少性能が悪くなると考えられる．静的手法にはアプリケーション毎に最適な閾値をあらかじめ計測しておかなくてはならないという手間があることも考慮すれば，今回提案した動的チューニング手法は有用であると言える．

4 まとめ

SSD の耐久性を高めるページパディング手法のための，閾値の動的チューニング手法を提案した．その結果静的に閾値を決める手法に比べ，計測の手間を省きさらに最大で 8% のブロック消去を削減した．

今後の課題は，チューニング手法のさらなる改良や，ハードウェア増加量を含めた評価を行うことがあげられる．

参考文献

[1] E.Ogawa, K.Kise. ”An Effective Page Padding Method for RAM Buffer Algorithm to Enhance the SSD Endurance”. In proceeding of the 4th International Symposium on Computing and Networking (CANDAR’16). 2016.

[2] Y.Kim, B.Tauras, B.Urgaonkar. ”FlashSim: A Simulator For NAND Flash-based Solid-State-Drives”. In Proceeding of First International Conference on Advances in System Simulation (SIMUL’09). 2009.

[3] U.T.Repository. ”OLTP Application I/O”. http://traces.cs.umass.edu/index.php/Storage/Storage

[4] S.I.Repository. ”MSN Storage CFS”. http://iotta.snia.org/traces/