ログ構造化ファイルシステム mylfs の設計と初期評価

鷹津冬将[†] 建部修見[††,†††]

1. はじめに

近年，様々な分野の計算機で使用されるストレージデバイスがハードディスクから SSD など高速なものに移行しつつあるが，SSD は，ハードウェアの書き換え回数に限界があるなどデバイス特有の欠点がある．そのような様々な制約を解決するために，SFS[1)] の様な様々な研究があり現在も研究開発が活発に行われている．また，ハードディスクも旧来のものに比較するとより高速化したが，依然としてシークタイムが発生するなどの欠点が残っている．そこで本研究では，HDD，SSD などにおいて効率的なファイルシステムの実現に向けて，ストレージへの書込が逐次書込となるよう設計を行った mylfs のプロトタイプ実装を行い，様々なアクセスパターンによる評価を行う．

2. mylfs の設計

Log Structured File System [2)] は Mendel Rosenblum らによって設計されたファイルシステムである．Ext3 など現在広く使われているファイルシステムはファイルの属性と実際のデータが別けられて保存されていることや，書き換える際に同じブロックを書き換えようとすることなどにより，書き換え時にシークが大量に発生する事からランダムライトの性能が低下している．これらの問題を解消するファイルシステムの 1 つに Log-Structured File System がある．Log-Structured File System はすべてのデータを 1 つのログとし，書き換え時においても実際にブロックの書き換えを行わずログに追記する形でシークの回数を減らしている．

そこで mylfs は Log Structured File System をベースとした設計にし，メタ情報を含めたすべてのデータをストレージ上に唯一のログとし末尾に追記する設計とした．

† 筑波大学情報学群情報科学類
†† 筑波大学システム情報系
††† 独立行政法人科学技術振興機構 CREST

3. mylfs のプロトタイプ実装

本稿では，Log Structured File System としてストレージへの最大の書き込み性能を評価するために FUSE を用いて mylfs のプロトタイプを実装した．最大の性能の調査のためファイルの属性や，ディレクトリエントリの情報，InodeMap などの各種メタ情報はメモリ上に保持する様に実装し，ストレージにはファイルのデータのみを書き込むように実装した．また，ストレージ上でログの末尾となっているアドレスはメモリ上に保持し，ファイルにデータを書き込む際は常にログの末尾となっているアドレスにシークした後データを書き込み，書き込み後にメモリ上のログの末尾のアドレスを更新するように実装した．

4. 評価

mylfs のプロトタイプ実装と比較する対象として ext3, NILFS2, btrfs, XFS, fuseext2 の 5 種類を定め，様々なアクセスパターンで評価した．

4.1 dd による評価

dd は入力から出力へデータをコピーするプログラムである．これを用いて 2 GiB のデータを書き込み，書き込んだデータすべてを読み込むことで評価した．書き込むデータの生成は/dev/zero を用い，読み込んだデータは/dev/null にはき出すようにしている．読み書きが終了した後に，ページキャッシュを解放させ，HDD，SSD ともに各ファイルシステムについて一連の動作を 10 回繰り返し，単位時間あたりの読み書きの性能の平均を求めた．この結果を**図 1** に示す．

4.2 書き換え性能の評価

一般にファイルは同じファイルを何度も書き換えられる．そこで同じファイルを何度も書き換えるプログラムを作成し，その実行時間を計測した．プログラムは，160MiB のファイルを生成し，4KiB ずつ書き換える処理を行う．書き換える場所は以下の 2 通りである．

(1) ファイルの先頭から逐次的に書き換える
(2) ファイルの中からランダムに書き換える

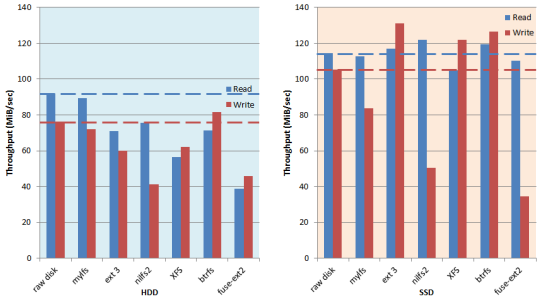

図 1 dd による評価結果

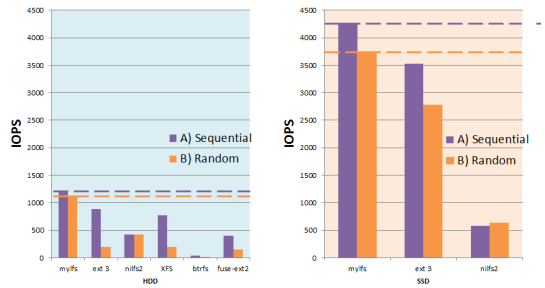

図 2 書き換え性能の評価結果

書き込み後は毎回 fsync() を呼び出す．最後にそれぞれの処理に要した時間を出力する．このプログラムを HDD，SSD ともに各ファイルシステムについて 10 回繰り返し，単位時間あたりの書換回数の平均を求めた．この結果を**図 2** に示す．SSD において，XFS，Btrfs，Fuse-ext2 は評価に 3600 秒以上経過しても終了しなかったため評価を打ち切った．

5. 関 連 研 究

今回 mylfs の設計のベースとした Log Structured File System として実装されているファイルシステムには NILFS2 がある．

NILFS2 のほかにも，Log Structured File System として実装されているものとして JFFS[4] や YAFFS[?]，UBIFS[?] がある．これらのファイルシステムは SSD などフラッシュメモリ式ストレージデバイス向けとして実装されている．これらのファイルシステムでは本研究と同様に Log Structured File System を使うことでウェアレベリングを行っている．

また，本稿では高速なストレージデバイスとして SSD をターゲットにしたが，SSD 以外の高速なストレージデバイスとして Storage Class Memory (SCM) があり，SCM 向けのファイルシステムとしては SCMFS[5] がある．

6. お わ り に

dd による逐次アクセスの性能評価では，raw device と比較した場合において，HDD では性能に対して書き込みが 94%，読み込みが 97%の性能を出した．また，SSD では書き込みが 79%，読み込みが 98%の性能を出した．また，fuse-ext2 と Ext3 の性能を比較すると fuse-ext2 は Ext3 に比べて 2 ～ 7 割の性能低下が確認された．これは FUSE によるオーバーヘッドが起因すると考えられる．

書き換え性能の評価は，mylfs が他のファイルシステムに比べて高い性能を示した．特にランダムに書き換える評価では広く使われている ext3 と比較すると HDD では 570%，SSD では 135%の性能を示した．

今後の課題として，メタ情報などのストレージへの書き込みと空き領域マネジメント機構の実装，分散ファイルシステムへの応用及び不揮発性メモリにおけるファイルシステムの設計などが考えられる．

謝辞 本研究の一部は，JST CREST「ポストペタスケールデータインテンシブサイエンスのためのシステムソフトウェア」および文科省次世代 IT 基盤構築のための研究開発「研究コミュニティ形成のための資源連携技術に関する研究」(データ共有技術に関する研究) による．

参 考 文 献

1) Changwoo Mina, Kangnyeon Kimb, Hyunjin Choc, Sang-Won Leed, Young Ik Eome.; SFS: Random Write Considered Harmful in Solid State Drives, Proceedings of the 10th USENIX Conference on File and Storage Technologies, pages 1-16, 2012.
2) Mendel Rosenblum and John K. Ousterhout; The Design and Implementation of a Log-Structured File System, Proceedings of the 13th Symposium on Operating System Principles, pages 1-15, October 1991.
3) 佐藤ほか, ログ構造化ファイルシステム NILFS の設計と実装, 情報処理学会 論文誌コンピューティングシステム (ACS), Vol.2, No.1, pp.110-122, 2009.
4) D. Woodhouse. Jffs: The journalling flash file system. In The Ottawa Linux Symposium, RedHat Inc, 2001.
5) Xiaojian Wu, Narasimha Reddy ; SCMFS : A File System for Storage Class Memory, Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, pages 39:1–39:11, 2011.