Gfarmにおけるクライアントキャッシュの効率化手法の提案

石黒　駿[†1]　大山恵弘[†1,†2]

1. はじめに

大規模なデータを扱うために，分散ファイルシステムが注目されている．分散ファイルシステムとは，複数のノードにデータを分散して配置し，それらのノードをまとめて一つのストレージとして提供するファイルシステムである．分散ファイルシステムの一つとして，Gfarm[1)] が挙げられる．Gfarm は広域ネットワークで利用可能な分散ファイルシステムであり，高容量，高信頼，高性能を実現している．Gfarm はメタデータサーバ及び I/O サーバから構成される．メタデータサーバは，Gfarm ファイルシステム上のファイルのメタデータを一括して管理する．I/O サーバは複数存在し，ファイルの内容を保存する．Gfarm のクライアントが Gfarm ファイルシステム上のファイルにアクセスする際は，まずメタデータサーバへアクセスし，どの I/O サーバに目的のファイルがあるかを知り，次に I/O サーバにアクセスしてデータをやりとりする．

クライアントは，ユーザレベルファイルシステムを実現するためのフレームワークである FUSE[2)] を利用して，Gfarm ファイルシステムをマウントできる．通常 Gfarm ファイルシステムにアクセスするためには，クライアントは専用の API を用いる必要があるが，マウントすることにより，システムコールを利用して Gfarm ファイルシステムにアクセスできる．FUSE にはカーネルモジュールが用いられているため，ユーザレベルファイルシステムはカーネルのキャッシュを利用できる．したがって，マウントされた Gfarm ファイルシステム上のファイル内容も，クライアントカーネルのページキャッシュにキャッシュされる．しかしながら，現在の実装では，Gfarm ファイルシステム上のファイル内容とページキャッシュの内容の一貫性を保つために，ファイルを open する度にページキャッシュは破棄される．Gfarm ファイルシステム上のファイルの更新は，クライアントには通知されず，ファイルオープン時にキャッシュが残っていると，古いデータにアクセスすることになってしまうためである．ゆえに，Gfarm ファイルシステム上のファイル内容とページキャッシュの内容が一致し本来ページキャッシュを破棄する必要がない場合にもキャッシュが破棄されるため，性能を損なっている．

そこで本研究では，ファイル open 時にページキャッシュの内容が最新であり破棄する必要がない場合に，ページキャッシュを保持するための手法を提案する．これによりページキャッシュの内容が最新でないときのみキャッシュを破棄するため，キャッシュを有効活用できる．

2. 方　針

FUSE は**図 1** に示すように，カーネルモジュールである FUSE モジュールとユーザレベルで動作する FUSE デーモンを連携させることにより，ユーザレベルファイルシステムを実現する．FUSE モジュールが，ユーザレベルファイルシステムに対するリクエストを受け取り，実際の処理を FUSE デーモンに要求する．FUSE デーモンは実際の処理を完了すると結果を FUSE モジュールへ返し，リクエストが完了する．FUSE デーモンは多くの場合，ネットワークで通信したり ext3 などのローカルファイルシステムにアクセスしたりする．

Gfarm ファイルシステムのマウントでは，FUSE デーモンとして gfarm2fs が利用される．gfarm2fs は，FUSE モジュールからリクエストを受け取ると，メタデータサーバや I/O サーバとやりとりして，結果を FUSE モジュールへ返す．提案手法の方針として，gfarm2fs が open リクエスト処理時にそのファイルのページキャッシュの内容が最新かどうかをチェックし，最新の場合はキャッシュを破棄しないように FUSE モジュールへ伝える．

3. 提案手法

まず，ページキャッシュの内容が最新であるかどうかを調べる方法であるが，これには Gfarm の inode 番号と世代番号を利用する．inode 番号は，ファイル固有の一意な値であり，世代番号はその inode が表すファイルが更新されるとインクリメントされる値である．提案手法では，まず open したファイルの inode 番号とその inode の世代番号をペアで覚えておく．次の

†1 電気通信大学
The University of Electro-Communications
†2 独立行政法人科学技術振興機構，CREST
JST，CREST

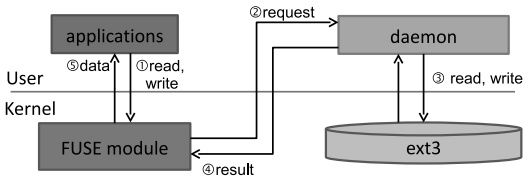

図 1　FUSE のアーキテクチャ

open 時に open 対象のファイルの inode 番号とその世代番号をメタデータサーバから取得し，その inode 番号を元に覚えておいた世代番号と取得した世代番号を比較する．もし，世代番号が一致していればそのファイルは更新されていないため，ページキャッシュを破棄せず，そうでない場合は破棄する．

次に gfarm2fs から FUSE モジュールにページキャッシュを破棄しないよう伝える方法である．FUSE では FUSE デーモンを起動する際のオプションで，ファイル open 時もキャッシュを破棄しないように指定することができるが，分散ファイルシステムのようなファイルシステムでは，ファイル内容の一貫性の問題からこのオプションは利用されない．FUSE デーモンを開発するためのライブラリは，ファイルの open 処理時に，このオプションの値を調べ，ページキャッシュを破棄するか否かを FUSE モジュールへ伝える．FUSE ライブラリは，ファイル情報を持つ構造体のメンバに値を設定することで，ページキャッシュを破棄するかどうかの情報を FUSE モジュールへ伝えている．そして，このメンバには FUSE デーモンからアクセスすることができる．そこで提案手法では，このメンバの値を FUSE ライブラリが設定した後に上書きすることにより，ページキャッシュの破棄操作を制御する．提案手法は，FUSE デーモンの変更のみで実現できる．

4. 評　　価

提案手法の有効性を調べるために，評価実験を行った．実験では，1GB のファイルを Gfarm ファイルシステム上に用意し，このファイルを 4KB ずつ read するプログラムの実行時間を計測した．キャッシュの効果を確かめるために，プログラムを 2 回連続で実行し，2 回目以降の実行時間を計測した．実験は，提案手法を用いない場合と用いる場合に対して行った．実験環境は，メタデータサーバ 1 台，I/O サーバ 1 台，クライアントのノード 1 台の計 3 ノードであり，クライアントのノードに Gfarm ファイルシステムをマウントした．すべてのノードの性能は，CPU が Intel Xeon 2.40GHz × 2，メモリが 48GB，HDD が 15,000rpm 600GB であり，ノード間は InfiniBand にて接続されている．また OS は Cent OS 5.5 64bit（kernel-2.6.18），Gfarm のバージョンは 2.4.2，FUSE のバージョンは 2.7.4，Gfarm2fs のバージョンは 1.2.3 である．実験の結果を**表 1** に示す．

表 1　Gfarm ファイルシステム上の 1GB のファイルを 4KB ずつ read するプログラムの実行時間

	提案手法なし	提案手法あり
実行時間（s）	2.96	0.330

実験の結果，ページキャッシュの内容が最新のファイルに対する read では，クライアントのページキャッシュが利用されるため，性能が向上することが確認できた．提案手法により，場合に応じてページキャッシュを破棄するかしないかを open 時に指定することができ，より効率的にページキャッシュを利用できる．

5. 関 連 研 究

Gfarm を対象にした FUSE のキャッシュ機構の研究として，石黒らの研究[3] がある．この研究では，同一の端末で複数のユーザが Gfarm ファイルシステムをそれぞれ別のディレクトリにマウントした場合に，本来同一ファイルのキャッシュが複数できてしまうものを，一つにまとめている．

6. 現状と今後

マウントした Gfarm ファイルシステムにおいて，ページキャッシュを open 時に破棄するかしないかを制御する手法を提案した．これにより，ページキャッシュの内容が最新である場合に，そのファイルの read 性能が向上することを確認した．今後は，他の分散ファイルシステムのキャッシュ機構について調査する．また，gfarm2fs のレイヤでのキャッシュ機構を設計・実装し，評価する予定である．

謝　　辞

本研究を行うにあたって，有益な助言を頂いた筑波大学建部研究室の方々に深く感謝する．また本研究は，科学技術振興機構戦略的創造研究推進事業（JST CREST）の研究課題「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」の支援を受けている．

参　考　文　献

1) Tatebe, O., Hiraga, K. and Soda, N.: Gfarm Grid File System, *New General Computing*, Vol.28, No.3, pp.257–275 (2010).
2) Szeredi, M.: FUSE: Filesystem in Userspace. `http://fuse.sourceforge.net/`.
3) 石黒　駿，村上じゅん，大山恵弘：Gfarm のためのカーネルドライバへのキャッシュ機構導入の検討，並列／分散／協調処理に関するサマー・ワークショップ（SWoPP2011）(2011).