6J-5

ファイル同期機能を持つファイルシステムの設計と実装

神保 直幸† 鶴岡 行雄‡ 多田 好克†
電気通信大学大学院†
日本電信電話株式会社 ソフトウェアイノベーションセンタ‡

1 はじめに

近年個人が所有する端末数が増加し，端末間でのファイル共有が広く利用されている．そのための手法として，サーバ上でファイルを一元管理する手法と端末上のファイルを同期する手法が挙げられる．

ファイルの一元管理の手法に，LAN 内でファイルを共有する NFS[1]や Coda[2]などのネットワークファイルシステムがある． しかし，これらは，端末がオフラインの際にはサーバ上のファイルにアクセスできないため，スマートフォンなどのモバイル端末では利用できない．

一方，端末間でのファイル同期は，ファイルの実体がそれぞれの端末上に存在するため，オフラインでのファイル操作が必要となるモバイル環境での利用に適している．モバイル端末の利用が一般的となり，一元管理ではなく同期によるファイル共有が今後の主流となると考えられる．ファイル同期の例として，Dropbox[3]などのオンラインストレージを利用したファイル同期サービスや rsync[4]，Unison[5]などのファイル同期ツールがある．

しかし，既存のファイル同期ツールはアプリケーションとして実装されているため，Andoroid や iOS などのアプリケーションがサンドボックス化された環境で利用できない． このような環境では，ファイル同期サービスが提供する API 等を利用して各アプリケーションが個別に同期処理を行う必要があり，アプリケーションを起動するまでサーバ上のファイルを取得できない．また，ファイル同期ツールは同期するファイルをパスで指定するため，ファイルを移動すると同期が維持できないという問題もある．

本研究では，アプリケーションではなくファイルシステムで同期を行うアプローチを選択し，端末間でのファイル同期を行うファイルシステムである FSyncFS を提案する．

FSyncFS はカーネル空間で動作するため，アプリケーションがサンドボックス化された環境であっても利用できる．また，同期に関する設定（同期設定情報）をファイルのメタデータとしてそれぞれのファイルに付与することで，ファイルごとの同期管理をパスに依存せずに行うことを可能にする．

The design and implementation of a file system providing file synchronization
Naoyuki Jinbo†, Yukio Tsuruoka‡ and Yoshikatsu Tada†
†The University of Electro-Communications
‡Nippon Telegraph and Telephone Corporation

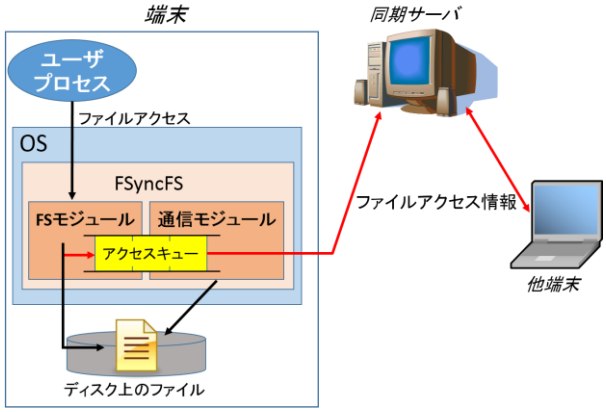

図 1 システムの構成

2 提案システムの設計

端末は単一のユーザによって利用される．また，ユーザはグローバルユニークなユーザ ID，たとえば E メールアドレスを持ち，端末はユーザ ID を記憶する．FSyncFS が満たすべき要求条件を以下のように定義する．

- ファイル同期処理はファイルへの操作をトリガにした適切なタイミングで行われること
- 同期対象ファイルを移動しても同期設定が維持できること

2.1 構成と手順

システムの構成を図 1 に示す．FSyncFS はカーネル空間で動作し，他のクライアント端末との同期は専用の同期サーバを介して行われる．同期対象のファイルはパスではなく各ファイルに付与されたグローバルユニークなファイル ID によって管理される．

端末上の FSyncFS は FS モジュール，通信モジュール，アクセスキューから構成される．FS モジュールはローカルファイルやメタデータの操作を行い，通信モジュールはサーバとの通信を行う．FS モジュールはアクセスキューを用いてファイルアクセス情報（ファイル操作の種類や引数）を通信モジュールに渡す．

同期サーバは各端末と通信する機能と各端末上のファイルとファイル ID を関連付ける機能を持つ．

ユーザプロセスによってファイルに変更が加わるファイル操作が実行されると，FSyncFS はファイルアクセス情報をアクセスキューに追加し，通信モジ

ュールがアクセスキューから取得した情報を同期サーバに送信する．また，通信モジュールが定期的に同期サーバと通信して他の端末からのファイルアクセス情報を取得し，ローカルのファイルに操作を反映することでファイル同期を実現する．

2.2 同期設定情報

FsyncFS では，ファイル同期に必要な情報である同期設定情報をそれぞれのファイルがメタデータとして持つ．同期設定情報には同期の ON/OFF を表す同期モード，同期に利用するサーバの IP アドレス，同期に参加する端末間でユニークなファイル ID，ユーザ ID によるアクセス制御の情報が含まれる．

また，FSyncFS はこれらの同期設定情報をユーザプロセスから取得・変更するための API として以下のものを提供する．

- ファイルの同期モードの変更，ファイルディスクリプタと変更後の同期モードを引数に取る．
- 同期サーバ情報の変更，ファイルディスクリプタと変更後のサーバ情報を引数に取る．
- アクセス制御情報の変更，ファイルのパス，対象のユーザ ID，変更後のパーミッションを引数にとる．

3 実装

FSyncFS は Linux 上で動作するスタッカブルファイルシステム[6]として実装した．そのため，ユーザは下層で動作するローカルファイルシステムを自由に選択することができる．ファイルの実体と，同期設定情報を持つメタデータファイルはローカルファイルシステムでは別のファイルとして保存される．

FSyncFS はユーザプロセスが実行する各種ファイル操作に対応した VFS レイヤの処理をフックし，ファイルの編集や新規作成の際にファイルアクセス情報として操作の種類と引数を抽出し，メタデータファイルから取得した同期設定情報とともにアクセスキューに追加する．

通信時間がファイルアクセスに影響を与えないように，通信モジュールはカーネルスレッドとして非同期に動作する．通信モジュールは同期サーバと通信し，ファイルアクセス情報の送受信を行う．

同期サーバ上には各端末に対応するキューがあり，端末からのファイル操作は同期先となる他端末に対応するキューに記憶される．それぞれの端末は自端末に対応するキューからファイルアクセス情報を取得し，通信モジュールが非同期にローカルファイルに反映する．

4 考察

提案システムである FSyncFS とネットワークファイルシステムおよび同期アプリケーションの特徴を比較する．

ネットワークファイルシステムにおけるファイルアクセスにはサーバとの通信を伴うため，特に低速なネットワーク環境下ではローカルファイルシステムと比較した場合にファイルアクセス速度が低下する．対して既存の同期アプリケーションではファイルの実体が端末上に存在することが保証できるため，アクセス速度の低下が発生しない．FSyncFS ではファイルアクセス処理に同期処理を追加しているため，多少の性能低下が発生するが，サーバとの通信は非同期に行われるため，ローカルファイルシステムに近い性能が期待できる．

また，FSyncFS は各ファイルが同期設定情報を持つため，ファイル単位で同期の ON/OFF や利用するサーバを設定できる．一方，ネットワークファイルシステムではディレクトリ単位で共有を行うため，ファイル単位での設定ができない．同期アプリケーションにはファイル単位で同期設定が可能なもの[4][5]もあるが，同期設定を維持したファイルの移動ができない．

5 まとめ

端末間でのファイル同期を行うファイルシステム FSyncFS を提案した．各ファイルに同期設定情報をメタデータとして持たせることにより，同期を維持したファイルの移動が可能である．

FSyncFS はカーネル空間で動作するため，サンドボックス環境下でも利用できる．

本提案の FSyncFS により，サンドボックスが一般的となったスマートフォンにおいて，基盤的機能としてのファイル同期を，広くアプリケーションに提供できる．

参考文献

[1] S. Shepler, B. Callaghan, D. Robinson, R. Thurlow, C.Beame, M. Eisler and D. Noveck, “Network File System(NFS) Version 4 Protocol, Request for Comments 3530.” April 2003.

[2] M. Satyanarayanan, J. J. Kistler, P. Kumar, M. E. Okasaki,E. H. Siegel and D. C. Steere, “Coda: A highly available file system for a distributed workstation environment,” IEEE Trans. Comput. vol. 39, pp. 447-459, April 1990.

[3] Dropbox, https://www.dropbox.com/, 2013 年閲覧 .

[4] A. Tridgell and P. Mackerras, “The rsync algorithm.” Technical Report TR-CS-96-05, Dept. of Computer Science, The Australian National University, Canberra, Australia, 1996.

[5] B. C. Pierce and J. Vouillon, “What’s in Unison? A formal specification and reference implementation of a file synchronizer.” Technical Report MS-CIS-03-36, Dept.of Computer and Information Science, University of Pennsylvania, 2004.

[6] E. Zadoc, I. Badulescu and A. Shender. “Extending file systems using stackable templates.” Proceedings of USENIX Annual Technical Conference, 1999.