バージョン管理機能を持つファイルシステムの開発

19T325 樋口史弥 （最所研究室）

1 はじめに

近年，ランサムウェアによってファイルが暗号化され，使用できない状態になる被害が多く発生している．ファイルの暗号化から守るためにはスナップショットを作成でき，そのスナップショットを更新できないファイルシステム (SSFS) を用いる必要がある．しかし，SSFS はファイルシステム全体を対象に，スナップショットのバージョンを前回のバージョンの差分として新たに作成する．そのため，多くの差分データが生じるという問題点がある．本研究では，ユーザ自身がファイル単位でバージョン作成の操作をすることにより，差分データ量を抑える YuihaFS を開発する．

2 従来の SSFS の問題点

SSFS である WAFL[1] や nilfs[2] はディスクへの書き込み時にファイルシステム全体のスナップショットを作成する．そのため，不必要なタイミングで必要としないファイルの差分を含むスナップショットが作成され，多くの差分データが生じる．

ファイル総数を S_{fcnt}，ファイルサイズの平均値を F_{size}，期間 (日) を R_{day}，ファイルシステムの数を FS_{cnt} としたこの時の，文献 [3] でのファイルの編集間隔のグラフを参考に任意の期間で生じる差分データ量を表す式 D_{size} は式 (1) になる．

$$D_{size} = 1.9 \cdot 10^{-1} \cdot S_{fcnt} \cdot F_{size} \cdot R_{day} \cdot FS_{cnt} \quad (1)$$

また，文献 [3] では拡張子に txt や c を持つ，ユーザが編集するファイルは多くあるが，このような拡張子を持つディスク使用量は少なく，dll や lib などのユーザが編集することのないファイルのディスク使用量が多いことが示されている．そのため，式 1 から算出される差分データ量の多くが必要のないデータということがわかる．

上記で述べた問題を解決するため，本研究では，スナップショットの作成をファイル単位で行う YuihaFS を開発する．YuihaFS ではファイルのスナップショットをバージョンと呼び，ユーザ自身がバージョン作成を操作することで，重要なファイルやタイミングでのみバージョンを作成し，生じる差分データを抑える．さらに，YuihaFS では編集されるファイルを対象としているため，バージョンの分岐が必要となる．

3 YuihaFS の設計

YuihaFS を開発する上で，特に重要な選択的バージョン作成，バージョン分岐，バージョン間でのデータブロックの共有の設計ついて述べる．YuihaFS のデータ構造を図 1 に示す．それぞれのバージョンのデータは Y-Version が保持し，データブロックへの参照などを保持する．Y-Versions は，複数の Y-Version への参照を保持する ”Y-V-Entry ”を保持する．Y-V-Entry は親バージョン，子バージョン，兄弟バージョン番号などを保持することで，木構造を構築できる．また，Y-Versions，Y-Version をまとめて，バージョン管理領域と呼び，1 つのファイルに対応する．Directory は，ディレクトリ内にあるファイルやディレクトリへの参照を保持する ” Directory Entry ”を保持する．Directory Entry 内には，ファイル名，Y-Versions の i-node 番号の他に，現在参照するバージョン番号を表す reference_version_number を保持する．

3.1 選択的バージョン作成

書き込み OPEN システムコールを実行する際に，バージョン作成を選択できるようにする．バージョン作成の条件は以下の通りである．

1. 子バージョンに対して，バージョン作成指示フラグ (”O_VERSION”) を与えた書き込み OPEN システムコールを実行した場合
2. 子ありバージョンに対して，書き込み OPEN システムコールを実行した場合

条件 1 により，任意のファイルのバージョンを任意のタイミングで作成できる．条件 2 により，子ありバージョンへの書き込みが禁止され，これにより，後述の分岐バージョンを作成の際のバージョン作成指示フラグの与え忘れによるデータの上書きを防ぐ．

3.2 バージョン分岐

親 Y-V-Entry が複数個の子バージョンを所有するするために，子 Y-V-Entry をリンクで繋いた兄弟 Y-V-Entry リストとし，親 Y-V-Entry は兄弟 Y-V-Entry リストの先頭を指す．

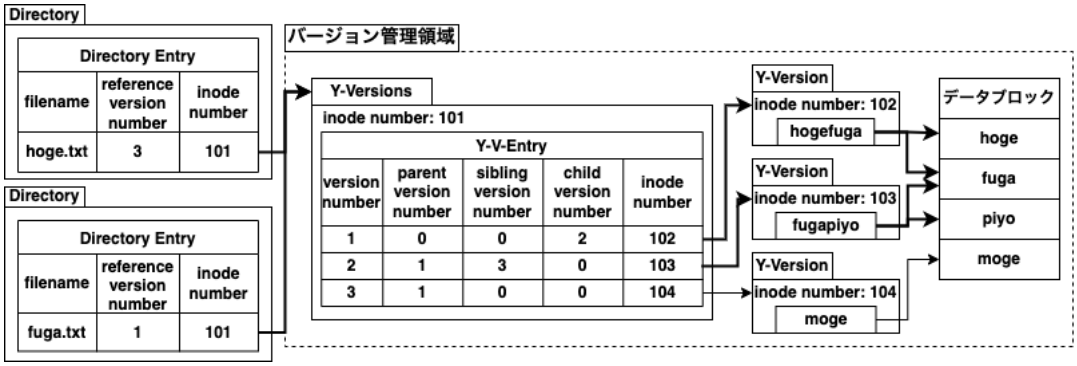

図 1: YuihaFS のデータ構造

3.3 バージョン間でのデータブロックの共有

新たなバージョンを作成する際，子バージョンは親バージョンのデータブロックの参照をコピーする．書き込み前は異なるバージョンであっても，同じデータブロックを参照している．複数のバージョンから参照されるデータブロックに対して書き込みが発生した場合，対象のデータブロックを Copy-On-Write し，バージョンが参照するデータブロックを更新する．

4 YuihaFS の実装

YuihaFS は FUSE(Filesystem in Userspace) を用いて実装した．FUSE はオペレーティングシステムを修正することなく独自のファイルシステムを開発できる機能を提供する．FUSE が要求するインタフェース (read, write, lookup など) を実装することでファイルシステムを構築できる．

5 評価

YuihaFS のディスク使用量を評価するために，YuihaFS と Linux での SSFS である nilfs に対して同じサイズの上書き，追記を行った際の差分データ量を比較する．nilfs はディスク書き込み時にファイルシステム全体のスナップショットを作成するファイルシステムであり，

図 2 は上書きをした際の nilfs の差分データ量に対する，バージョン作成頻度を変化させた YuihaFS の差分データ量を比較したものである．すべての書き込み量においてバージョン作成頻度を減らすと YuihaFS の差分データ量が小さくなり，バージョン作成頻度を半分に減らすと差分データ量も半分に減少している．また，すべての実験において，上書き量が増えるにつれて，nilfs のディスク使用量に対する YuihaFS のディスク使用量が増加し，毎回バージョン作成での 1MB 上書き実験では，YuihaFS の差分データ量が nilfs の差分データ量と比較して 1%ほど多くなった．しかし，毎回バージョン作成をするという YuihaFS の意図しない利用方法をした場合でも，差分データ量の増加を 1%に抑えることができているため，YuihaFS は差分データ量という点で nilfs よりも優れていると言える．追記の際も同様の結果が得られたが，バージョン作成頻度を減らした際の差分データ量の減少が上書きと比べて小さくなった．これは，上書きでは最大で 1 ブロックの差分データしか減らせないためである．

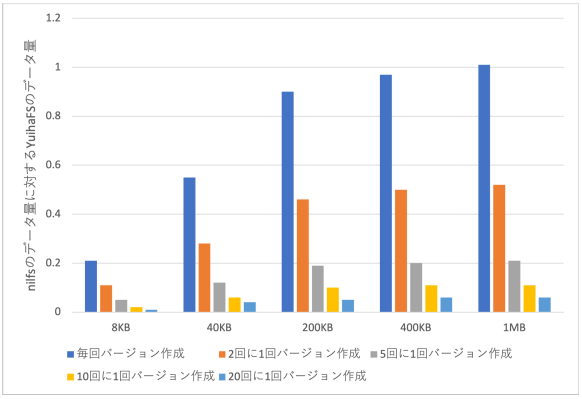

図 2: nilfs の差分データ量に対する YuihaFS の差分データ量 (上書き)

参考文献

[1] Dave Hitz,James Lau,Michael Malcolm File System Design for an NFS File Server Application https://www.cs.princeton.edu/courses/archive/fall04/cos318/docs/netapp.pdf

[2] 佐藤孝治，小西隆介，木原誠司，天海良直，盛合敏 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）,pp.110-122,2009

[3] William J.Bolosky Dutch T.Meyer， A study of practical deduplication, FAST '11: 9th USENIX Conference on File and Storage Technologies, pp. 1–13, 2011.