

ファイルアクセスを契機としたファイルシステム管理情報変換方式

亀井 仁志[†] 揚妻 匡邦[†] 浦野 明裕[†] 中村 隆喜[†]

(株) 日立製作所 システム開発研究所

1. はじめに

IT 技術の進展に伴い、情報を蓄えるファイルシステムには、大規模化と同時に高機能化への要求が増している。特に最近では企業の情報漏えいが社会問題化している状況から、ファイルの暗号化やファイルアクセスを細かく制御可能な ACL (Access Control Lists) [1] といったセキュリティを向上する機能への要求が高い。

ファイルシステムは、ファイル作成時刻などの付加的な情報をメタデータとして、ファイルデータと併せて記録している。例えば、ACL 機能をファイルシステムへ追加するには、既存のメタデータを拡張して、ACL 情報を保持して管理する必要があった。

既に運用中のファイルシステムに、ACL のような新機能を追加するには、そのファイルシステムにあるファイルへ新たなメタデータを付与し、新しい形式のファイルシステムへ移す、ファイルシステム機能移行という作業が必要である。しかし、従来の機能移行方式では、移行用ワークボリュームが必要となったり、移行中にサービス停止が停止するといった問題があった。そこで我々は、ファイル毎の移行処理をファイルアクセス時まで遅延させることで、これらの問題を解決するオンアクセス方式を提案する。

2. 従来の機能移行方式と課題

従来の機能移行方式では、機能拡張前のファイルシステムプログラム用に作成したファイルシステム (旧 FS) のボリュームから、機能拡張を行ったファイルシステムプログラム用に作成したファイルシステム (新 FS) のボリュームへ、ファイルをコピーすることで機能移行する (図 1)。新 FS の新メタデータは、ファイルコピー時にコピープログラムによって付与される。

この方式は、新 FS 用ボリュームが新たに必要となるため、新たにボリュームが追加できない環境では利用できない。

また、ファイルコピー中にコピー元ファイルを編集すると、コピー前後で不整合が発生する

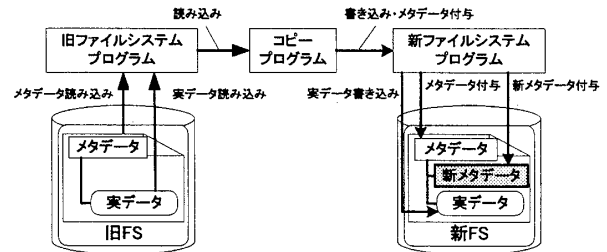


図 1 従来機能移行方式

ため、機能移行中はファイルを利用できない。これらの問題は、大規模ボリュームにおいて特に顕著となる。

以上により、本研究では、大規模ボリュームにおける、ファイルシステム機能移行時のワークボリューム確保不能、及び長時間のサービス停止を課題とする。

3. 新機能移行方式の提案

本章では、2 章で述べた課題に対する解決方式を述べる。

3.1 提案方式(a)：一括変換方式

一括変換方式は旧 FS ボリュームに対して機能移行コマンドを実行し、新 FS へ直接機能移行する方式である。これにより、従来方式で問題となる追加ボリュームが不要となる。一括変換方式の動作概要を図 2 (a) に示す。

まず、ファイルシステムそのものの管理情報である FS 管理情報を、旧 FS を示す情報から新 FS を示す情報へ書き換える。次に、格納している全ファイルに対して新メタデータを付与する。

本方式は、コピー処理が不要な分、従来の機能移行方式よりサービス停止時間が短縮できる。ただし、新メタデータが付与されていないファイルへの操作を避けるため、移行コマンド実行中はファイルへのアクセスを禁止する必要がある。機能移行中はファイルシステムを利用できない。また、機能移行時間はファイル数に比例して長くなる。そのため、従来方式と同様に、大規模ボリュームでは、サービスが長時間停止する場合がある。

A Method for Conversion of Filesystem Functions by File Access

[†] Systems Development Laboratory, Hitachi, Ltd.

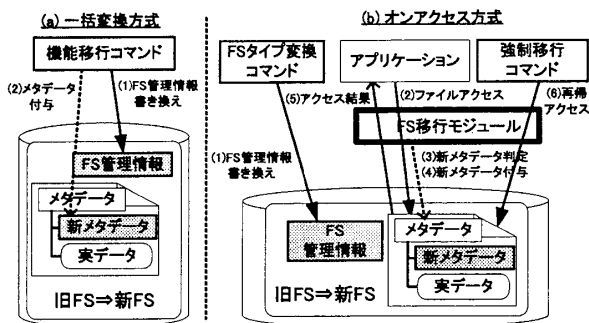


図 2 新機能移行方式の動作概要

3.2 提案方式(b) : オンアクセス方式

オンアクセス方式は、旧 FS ボリュームへのファイルアクセスを契機に新 FS 用のメタデータを付与する事により、旧 FS から新 FS へ直接機能移行する方式である。一括変換方式と同様に追加ボリュームが不要となる。さらに、新メタデータの付与契機をファイルアクセス時とすることでサービス停止時間を抑える。オンアクセス方式の動作概要を図 2 (b) に示す。

機能移行時には、FS 種別が新 FS であることを示すように、FS 管理情報を書き換える処理のみ行う。機能移行後は、アプリケーションからファイルアクセスが発生した時点で、アクセス先のファイルを確認する。その時、新メタデータがなければ、適切な新メタデータを付与する。これにより、機能移行中のファイルシステム利用を可能とする。本方式では、サービス停止時間は FS 管理情報の書き換え時間のみとなる。

ただし本方式は、ファイルアクセスを契機に新メタデータを付与するため、バックアップファイルなどのアクセスが無いファイルは、新メタデータが付与されず、機能移行が完了しない。そのため、旧 FS に格納された全ファイルを再帰的に走査しアクセスすることで、強制的に新メタデータを付与し機能移行の完了を保証する。

4. 評価

従来方式と 2 つの提案方式についての評価結果を表 1 に示す。

ファイルアクセス時間は、一度に 100 ファイルにアクセスした場合の経過時間を 10 回取得し、平均した結果である。一方、サービス停止時間と必要ボリュームサイズは、10GB のファイルシステムに 4kB のファイルが 5 千ファイル格納されている場合と 50 万ファイル格納されている場合を想定して、1 ファイルの実験結果を元に計算した。また、ファイルアクセス時間は従来方式を 1 とした場合の相対性能である。必要ボリュ

表 1 評価結果

比較項目	従来方式	提案方式	
		方式(a)	方式(b)
ファイルアクセス時間(相対値)	1 回目	1.00	4.19
	2 回目		1.00
サービス停止時間	5 千ファイル	約 1 時間	数分 数 10 秒
	50 万ファイル	約 10 時間	
移行時必要ボリュームサイズ(相対値)	1.19	0.19	0.19

ームサイズは機能移行前の容量を 1 とした場合の相対容量である。そして、サービス停止時間は実時間である。

ファイルアクセス時間に関しては、提案方式(b)の 1 回目のアクセスでは性能低下が見られるが、2 回目以降のアクセス時間は他の方式とほぼ同等である。また、2 回目以降のファイルアクセス時間は常に 1.00 であり、アクセス数の多いファイルでは、平均すると他の方式と同等となる。

サービス停止時間に関しては、提案方式(b)が最も優れている。提案方式(a)も従来方式に比べて 1/10 程度に改善している。しかし、提案方式(a)はファイル数に比例してサービス停止時間が増大する。

必要ボリュームサイズに関しては、提案方式(a), (b)いずれも従来方式に対して同等程度に改善している。

以上より、提案方式(b)のオンアクセス方式が、ファイル数の増加、ファイルシステム数の増加に対応できる、スケーラビリティの高い方式であるといえ、大規模ボリュームに対しては優位であると考えられる。

5. おわりに

本論文では、運用中のファイルシステムに機能追加の際のファイルシステム機能移行方式を検討した。ファイルアクセスを契機にメタデータを付与することで機能移行するオンアクセス方式を提案した。提案方式は、機能移行用ワークボリュームを不要とし、サービス停止時間を抑え、円滑な移行を実現した。

6. 参考文献

- [1] ACL (Access Control Lists) :
<http://www.linux.or.jp/JM/html/acl/man5/acl.5.html>