

## クイックリカバリファイルシステムの開発\*

4B-4

山本 勉†

村上 善行†

島村 伸之‡

(株)沖テクノシステムズラボラトリ¶

## はじめに

現在主なUNIX<sup>1</sup>オペレーティングシステムで使用されているファイルシステムでは、高速化のためディスクアクセスを少なくなるように工夫している。

そのため、ファイルシステムの使用中にシステムダウンが発生した場合、アクセス中のファイルの管理情報に不整合を生じてしまう。このままでは正しくファイルへアクセスできなくなるため、システムのレポート時にファイルシステムの管理情報の整合性チェック・修正(以後リカバリ)を行う必要がある。

この管理情報の整合性のチェック・修正はファイルシステム全体を調べるため、ファイルシステムの容量に比例して時間がかかってしまう。これは即時性を求められるような実務処理に使用している場合、問題になる。

そこで我々はリカバリ時間を可能な限り短くするファイルシステム「クイックリカバリファイルシステム」を開発する事にした。

## 1 開発方針

ファイルシステムの使用時にシステムダウンした場合、アクセスしていたファイルの管理情報に不整合が生じる。リカバリに時間がかかる原因は、アクセスしていたファイルが特定できないため、全ての管理情報をチェックし、不整合部分を修正するためである。つまりリカバリ時にどのファイルにアクセスしていたか特定できるようにすれば、リカバリ処理は不整合部の修正だけで済むので短時間になるはずである。このため、ファイルの変更履歴(以後log)を記録する処理を追加するようにした。

\*Development of Quick Recovery File System

†Tutomu Yamamoto

‡Yoshiyuki Murakami

§Nobuyuki Shimamura

¶OKI Technosystems Laboratory

<sup>1</sup>UNIXはUNIX System Laboratories, Inc.が開発し、ライセンスしている。

しかし、単にlogを残す処理を追加するだけでは、ライトスルーとなり、パフォーマンスが悪くなる。そこで、logのディスクへの書き込みを、ファイルのデータと同じようにバッファキャッシュ(ファイルシステムがディスクとのデータ入出力時に使用するキャッシュ)を介して行うようにした。

またlogの書き込み中にシステムダウンしてしまった場合、logの内容が正しくない。このような場合に対応するため、logに通し番号とチェックサムを用いてリカバリ時にlogの正当性をチェックする事とした。

以上方針をまとめると、次に挙げる機能を追加することになった。

1. バッファキャッシュを介したlogの書き込み
2. リカバリ時のlogのチェック

## 2 性能改善

前節で述べたように3つの機能を追加することでクイックリカバリファイルシステムは実現できた。

しかし、ファイルへのアクセスにおいてlogの書き込みがある分だけどうしても通常のファイルシステムよりパフォーマンスが落ちてしまう。

そこで、ファイルシステムのディスクへの入出力を変更する事にした。通常のファイルシステムはバッファキャッシュを介してファイルシステムのブロック長(4096byte等)でディスクとの入出力を行なっている。大きいファイルの場合、このファイルシステムのブロック長に分割されてディスクへ書き込む事になる。

これをまとめて書き込むようにした場合、分割するオーバーヘッド、ディスクの処理時間が減る。

また、まとめて書く事によりファイルがディスク上の連続した領域に書きこまれるので、ファイルの読み込み時間も短くなる。

このことから、

- バッファキャッシュの操作

という変更を行うことにした。

### 3 リカバリ時間とパフォーマンス比較

完成したクイックリカバリファイルシステムと一般のファイルシステムの比較を行なった。

比較に用いた機種は OKIstation7300model30

使用 OS は OKI-UX Ver1.24 である。

まず、アクセスファイル数を変えていき、リカバリにかかる時間を測定したものを測定した。この結果を図1に示す。

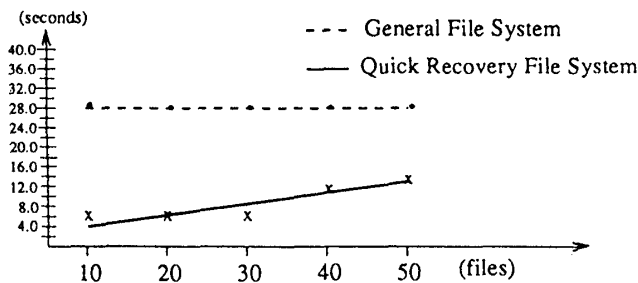


図 1: ファイルシステムのリカバリ時間

この図から、一般のファイルシステムでは、リカバリ時間がほぼ一定である事から「リカバリ時間はアクセスファイル数には依存しない」事がわかる。クイックリカバリファイルシステムではファイルアクセス数に比例して時間が長くなっていくが、一般のファイルシステムのリカバリ時間を越える事はない。

次にファイルシステムの大きさを変更した時のリカバリ時間を測定した。この結果を図2に示す。

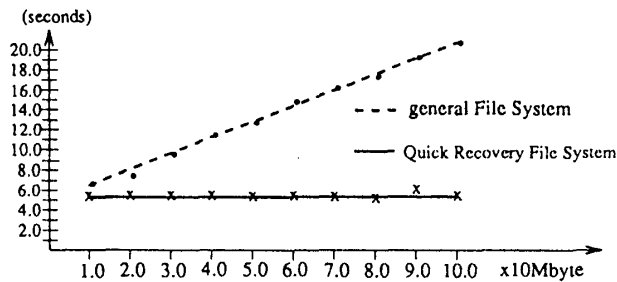


図 2: ファイルシステムのリカバリ時間

この図から、一般のファイルシステムでは、ファイルシステムの大きさに比例して時間が長くなる事から「リカバリ時間はファイルシステムの大きさに依存する」事がわかる。クイックリカバリファイルシステムではリカバリ時間が一定であることから、ファイルシステムの大きさとリカバリ時間が無関係であることがいえる。

ファイルアクセスのパフォーマンスを測定するために BYTE ベンチマークの file copy を用いた。この結果を図3に示す。

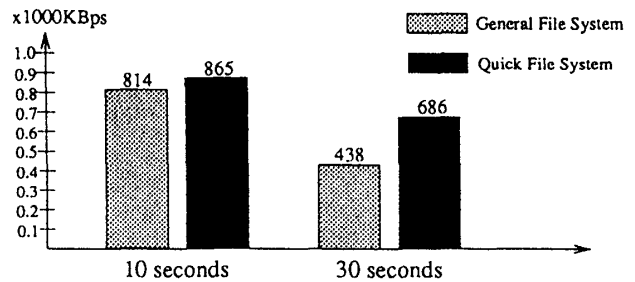


図 3: BYTE ベンチマーク file copy

この図から、クイックリカバリファイルシステムの方が file のコピーを高速に行なっている事がわかる。

### おわりに

通常のファイルシステムに対し

- log の書き込み
- リカバリ時の log のチェック
- バッファキャッシュの操作

という3点の変更を行なってクイックリカバリファイルシステムを開発した。これにより、ファイルアクセスのパフォーマンスを落さずにクイックリカバリファイルシステムが実現できた。

### 参考文献

- [1] Maurice J.Bach, UNIX カーネルの設計, (坂本文, 多田好克, 村井純 訳), 共立出版, 1990
- [2] S.J.Leffler, M.K.Mckusick, M.J.Karels, J.S.Quarterman, UNIX 4.3BSD の設計と実装 (中村明, 相田仁, 計宇生, 小池汎平 訳), 丸善株式会社, 1991