

テープ用ファイルシステムでのファイル・リストアの高速化

長谷川 徹 大石 豊 今井 直樹 神谷 昌範 平田 崇将

A Method to Improve the Restoration of Files from Tape File Systems

Tohru Hasegawa, Yutaka Oishi, Naoki Imai, Masanori Kamiya and Takamasa Hirata

テープ用ファイルシステムであるLTFS (Linear Tape File System) は、テープ上のデータへ特別なソフトウェアを用意せずにアクセスすることを可能にし、オペレーティング・システム付属のコピー・コマンドなどを使用したバックアップなどで使われている。しかし、バックアップ済みの複数のファイルをテープからHDDへリストアする場合に、リストアするファイルの順番によっては非常に長い時間がかかるという課題がある。本論文ではこの課題を解決するため、ファイルの列挙をテープ上の並び順に行う、テープ順列挙法という手法を提案する。実際にLTFS Windows版に適用したところ、リストア時間を90%以上短縮することができた。

Linear Tape File System (LTFS) is a file system specially designed for tape. It allows users to access a tape without using any special software. So, LTFS is used to make backup copies by using the COPY command included in the operating system in question. However, there is a problem in that it takes a long time to restore a backup copy depending on the location of the files on the tape and the order in which they were copied. In this paper, we propose a method to improve the restoration of files by enumerating them in order of their location on the tape. By using this method, the time needed to restore a backup from tape can be improved by more than 90 percent.

Key Words & Phrases : LTFS, テープ, ファイルシステム, バックアップ, リストア
LTFS, tape, file system, backup, restore

1. はじめに

大容量のデータを保管するストレージとして、容量あたりの単価の安さ、省電力などからLTO (Linear Tape-Open) などのテープが注目を浴びている [1]。従来、テープにデータを保存したり、テープ上に保存したデータを読み出したりするには、専用のアプリケーションが必要であった。IBMではこのようなアプリケーションとしてTivoli Storage Managerがある [2]。

このようなアプリケーションを使用するには、使用方法や設定方法など特別な知識が必要であり、テープを使用する上での制限となっていたとも言える。この制限を排除する方法として、テープへのアクセスにLTFS (Linear Tape File System) というファイルシステムを使う方法がIBMにより開発された [3]。LTFSは、コンソーシアムによって承認された標準規格であるLTFSフォーマット [4] に従った形式でテープに情報を記録することで実現され、IBMだけではなく、HPやQuantumなども提供している [5] [6] [7]。

LTFSによってテープへのアクセスが容易になり、USBメモリーのような感覚でテープを使用することが可能となる。特別なソフトウェアを導入することなく、オペレーティング・システム付属のコピー・コマンドなどでアクセスできるため、データのバックアップなどで利用されている [8]。

しかし、テープはその仕組み上、テープ上に書き込まれた

データを読み出すまでに時間がかかることがあるため、バックアップ済みのファイルをLTFSからHDDへリストアする際に、リストアするファイルの順番によっては非常に長い時間がかかってしまう、という課題がある。

本論文では、この課題を解決する方法として、ディレクトリー内のファイルを列挙する順番をテープ上の並び順に行う、テープ順列挙法という手法を提案する。テープ順列挙法を用いることで、オペレーティング・システム付属のコピー・コマンドを用いて読み出ししても、テープ上の並び順にリストアできるため、リストアにかかる時間を大幅に短縮できる。実際にLTFS Windows版に適用し、リストア時間が最大99%短縮できることを確認した。

2. LTFSの概要

2.1 LTFSフォーマット

LTFSフォーマットは、テープ上の記憶域をデータ・パーティションとインデックス・パーティションとに、論理的に分割して利用することを前提としている。図1のように、インデックス・パーティションには主にファイルのテープ上での位置などの、ファイルシステムの内部情報を保管し、データ・パーティションにはファイルの本体 (データ) を保管する。

LTFSは、このLTFSフォーマットに準拠し、パーティション分割機能をサポートしている、第5世代LTO (LTO5) やIBM System Storage TS1140 テープ・ドライブ [9] などに対応している。

提出日:2011年9月20日 再提出日:2011年12月5日

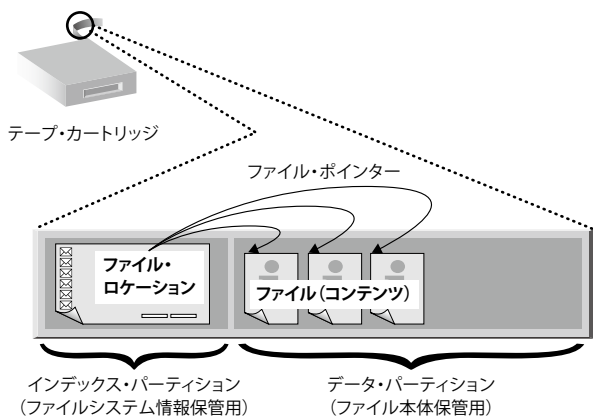


図 1. パーティションの構成

2.2 LTF5でのデータの書き込み

テープは、シーケンシャル・アクセス・デバイスと呼ばれるように、データの読み書きをする際に、先頭から順番にアクセスする。特に、新たなデータを書き込む際には、テープ上の最後に書き込まれたデータの後に追記することが一般的である。

LTF5でもデータ・パーティションに書かれるデータは、テープ上にシーケンシャルに書き込まれ、更新も含めすべての変更は追記により記録される。

例えば、3つのファイル A,B,C を順番に LTF5 に書き込んだ場合、図 2 のように A → B → C の順に記録される。

この時、ファイル A の内容を更新した場合には図 3 のように、更新済みのファイル A のデータは最後に追記される。

更新の単位はアプリケーションに依存し、ファイルの全部を書き換えず、一部分だけ書き換えるアプリケーションも存在する。そのようなアプリケーションでファイル A の一部分を書き換えると、ファイル A のデータは分断され、図 4 のように更新部分のみ追記される。

3. ファイル・リストアにおける課題

3.1 ファイルの頭出し

LTF5 上にあるファイルを読み出す時の流れは、次のようになっている。

まず、テープのマウント時にインデックス・パーティションに書かれているファイルの位置情報を読み出し、メモリーの中に保存しておく。

次に、実際のファイルの読み出しが発生した時に、メモリーの中にあるファイルの位置情報を基に、テープ上のファイルの先頭に移動し、ファイルを読み出す。このテープ上のファイルの先頭に移動する動作を本論文では、ファイルの頭出し、と呼ぶこととする。

LTO5 ではテープの長手方向の長さは約 800 メートルあり、ファイルの頭出しにはこの長手方向のテープの移動を伴うため、任意のファイルの頭出しには平均 45 秒程度かかる。

3.2 ファイルのリストアにかかる時間

LTF5 上のファイルを HDD にリストアする場合、リストアに

かかる時間は、ファイルの頭出しにかかる時間、ファイルのデータ本体の読み出しにかかる時間、HDD への書き込みにかかる時間の合計である。

LTF5 上の複数のファイルを HDD へリストアする場合、ファイルのデータ本体の読み出し時間と HDD への書き込み時間は、ファイルをリストアする順番には依存しないが、ファイルの頭出し時間はリストアする順番に大きく依存する。

例えば、図 2 のように 3 つのファイル A,B,C がテープに書かれている場合について考える。A → B → C の順でリストアすれば、ファイル A の末尾とファイル B の先頭および、ファイル B の末尾とファイル C の先頭がそれぞれ隣接しているため、ファイル B,C の頭出しに時間はかからない。一方、C → B → A の順でリストアすると、ファイル C の末尾からファイル B の先頭および、ファイル B の末尾からファイル A の先頭への移動が必要となるため、各ファイルへの頭出しに時間がかかる。

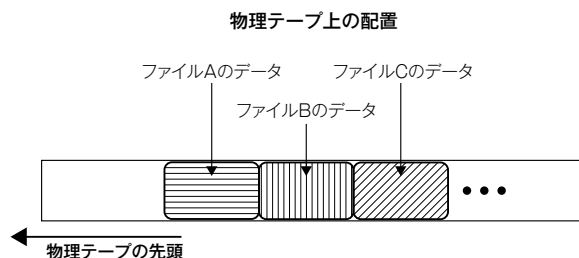


図 2. ファイル A,B,C のテープ上の配置

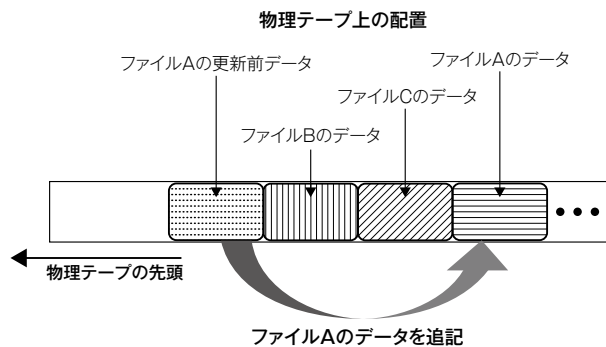


図 3. ファイル更新時のテープ上の配置

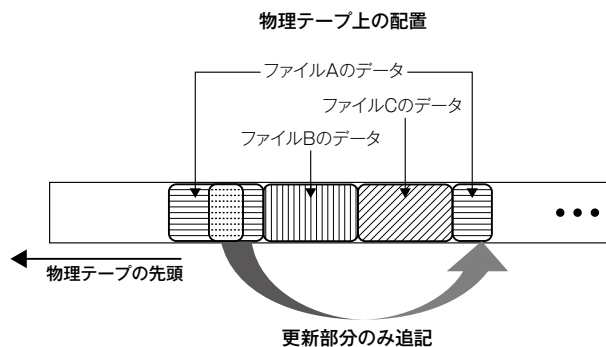


図 4. 一部更新時のテープ上の配置

3.3 コピー・コマンドによるリストアの順番

LTFS の利点として、特別なソフトウェアを必要とせず、オペレーティング・システム付属のコピー・コマンドを使って、ファイルのバックアップ、リストアが行えるということがある。

例えば、Windows 環境では、複数のファイルが存在しているディレクトリーをコピーするには、エクスプローラーでディレクトリーを選んでドラッグ・アンド・ドロップしたり、コマンド・プロンプトから `xcopy` コマンドを使用したりすることでファイルのバックアップ、リストアができる。

コピー・コマンドの実装方法はコマンドに依存するが、一般的に簡単な実装のコピー・コマンドは、`dir` コマンドを実行した場合と同様に、ディレクトリー内のファイルを列挙し、その順にコピーを行う。ここで、簡単な実装のコピー・コマンドとは、Windows の `xcopy` コマンドや Linux の `cp -r` コマンドなどを指す。

以下、具体的なファイルの列挙方法について、Windows を例に説明する。

コマンドやアプリケーションでファイルの列挙を行うと、Windows は、IRP (IO Request Packet) をファイルシステムに発行し、問い合わせを行う。具体的には図 5 に示すように IRP_MJ_DIRECTORY_CONTROL (Minor Function Code は IRP_MN_QUERY_DIRECTORY) をファイルシステムに発行する。これに対し、ファイルシステムは、この IRP により用意されたバッファーに、ディレクトリー内のファイル名などのファイルの情報をコピーして返す。

この IRP には Flag というパラメーターがあり、IRP を複数回に分けて列挙を行う場合には、最初の IRP か 2 度目以降の IRP かを区別できるようになっている。ファイルシステムは、最初の IRP の時には、内部で使用している FileID の 0 から順に列挙し、2 度目以降の場合には前回の IRP で列挙したファイルの次の FileID を持つファイルから順に列挙を行う。

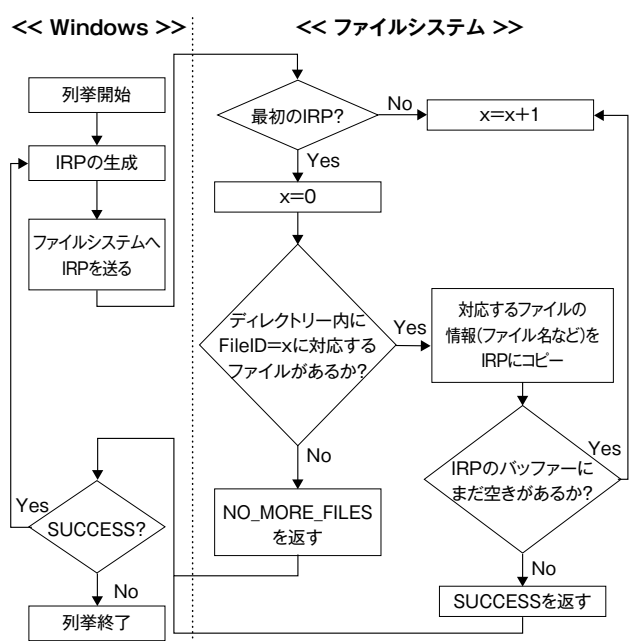


図 5. ファイル列挙の流れ

ここで、「各 FileID にディレクトリー内のどのファイルに対応させるか」については明確な規定はない。すなわち、「ファイルを列挙した場合にディレクトリー内のファイルがどのような順番で列挙されるか」はファイルシステムの実装に依存している。

例えば、Windows で標準的なファイルシステムである NTFS で列挙した場合には、ファイル名のアルファベット順で列挙され、FAT ではファイルの作成時間順で列挙される。これは、`dir` コマンドを実行してみることで確認することができる。

ここでは Windows を例にとって説明したが、Linux などでも同様に、`cp -r` コマンドでディレクトリーをコピーした場合には、ファイルシステムからのファイルの列挙順にコピーが行われる。

3.4 ファイル・リストア時間の課題

LTFS でも FAT と同様に、ファイルの作成時間順で列挙を行っている。これはファイルシステムの実装を単純にするため、および、`dir` コマンドなどでのファイル列挙の結果を、既存のファイルシステム (FAT) と合わせることで、ユーザーにとって直感的に分かりやすいものとするためである。しかし、LTFS では、コピー・コマンドでファイルをリストアする際に、非常に時間がかかる場合があるという課題がある。

LTFS 上のファイルを更新すると、そのデータはテープ上の最後に書き込まれたデータの後に追記される。言い換えると、テープ上のデータの並びは、ファイルの更新時間順となる。このため、ファイルの更新を行うと、テープ上のデータの並びとファイルの作成時間順とは一致しないことがある。

具体的には、ファイルを A → B → C の順で作成した後、B → A の順で更新した場合には、図 6 のようにテープ上にはファイル B、ファイル A の内容が順に追記されていく。すなわち、テープ上、ファイルは C → B → A の順で書かれていることになる。

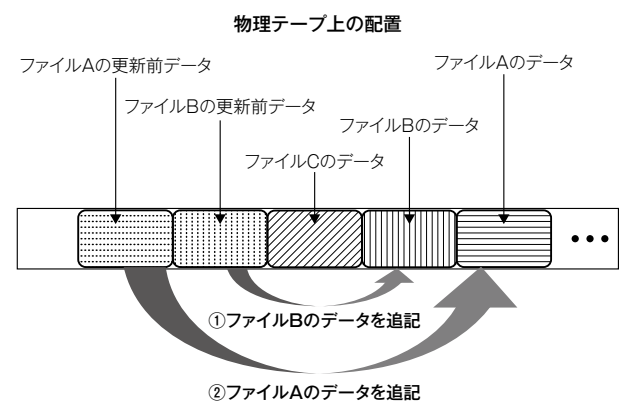


図 6. B → A と更新した時のテープ上の配置

LTFS では、ファイルを列挙した場合には、更新時間順ではなく、作成時間順で列挙されるので、A → B → C の順で列挙される。このファイル A,B,C を含むディレクトリーを `xcopy` コマンドでリストアした場合には、次のような流れになる (図 7 参照)。

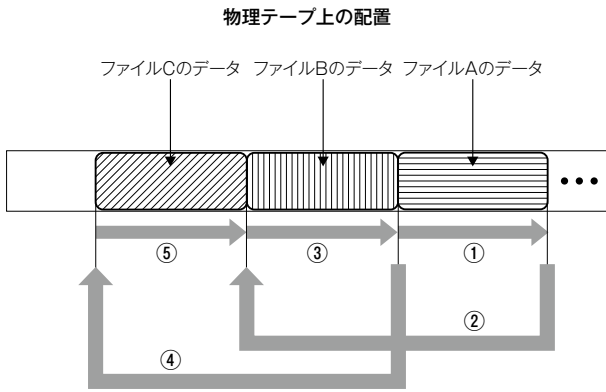


図 7. リストア時の読み出し順

- ① ファイル A を読み出し
- ② ファイル B の頭出し
- ③ ファイル B の読み出し
- ④ ファイル C の頭出し
- ⑤ ファイル C の読み出し

②と④とでファイルの頭出しを行うので、ファイル A の末尾からファイル B の先頭および、ファイル B の末尾からファイル C の先頭に向けテープを巻き戻す作業が必要となり、リストアにかかる時間が長くなる。これが LTFS を用いたファイル・リストア時間の課題である。

3.5 ファイル・リストア時間の測定結果

実際に複数のファイルを含むディレクトリーを xcopy コマンドでテープから HDD へリストアした場合に、リストアにかかる時間を測定した。このときの測定環境は表 1 で、ディレクトリーは、次のような複数のファイルを含むものを 5 つ用意した。

- 1) 100MB のファイルを 10 個
- 2) 50MB のファイルが 20 個
- 3) 10MB のファイルが 100 個
- 4) 5MB のファイルが 200 個
- 5) 1MB のファイルが 1000 個

このディレクトリー内のファイルを一度に LTFS に書き込み、一度も更新されていない場合には、各ファイルのデータはテープ上に作成時間順に配置されている。そのため、従来のファイルを作成時間順で列挙する手法でもファイル・リストア時間は長くない。しかし、各ファイルが更新された場合にはファイル・リストア時間が長くなることもある。

従来手法でファイル・リストア時間が長くなる顕著な例として、ファイルを書き込み後、すべてのファイルを書いた順番と逆の順番で更新した場合について測定を行った。

この場合、上記のディレクトリーで、ディレクトリー内のファイルを列挙すると、テープ上の並びと逆順に列挙される。実際に xcopy コマンドを用いて LTFS から HDD へリストアした時にかかった時間を表 2 に示す。

各ディレクトリーに含まれるファイルの合計のサイズは 1GB

表 1. 測定環境

PC	IBM System x3500 M2
Operating System	Windows Server 2008 R2 SP1
Memory	6GB
Tape Drive	IBM System Storage TS2350 (Firmware Revision B179)
LTFS	IBM LTFS Single Drive Edition (Version 1.2.2.1234)

表 2. リストアにかかる時間

ファイル・サイズ	ファイル数	コピー時間
100MB	10	1 分 56 秒 95
50MB	20	4 分 11 秒 76
10MB	100	15 分 22 秒 71
5MB	200	30 分 35 秒 94
1MB	1,000	33 分 0 秒 02

であるが、1MB のファイルを 1,000 個リストアした場合には約 33 分もかかる。

4. テープ順列挙法

LTFS でファイルを列挙する際に、テープ上の位置を考慮して各ファイルに FileID を付けることによって、テープ上の並びの順に列挙する手法を提案する。この方法をテープ順列挙法と呼ぶこととする。

テープ順列挙法の実装は、図 8 のような流れで実現できる。

従来の列挙方法との実装上の大きな違いは、IRP にファイルの情報をコピーするところ（図 8 の③）だけである。具体的には、次のような実装になる。

まず、各ファイルに、そのファイルの先頭がテープ上に書か

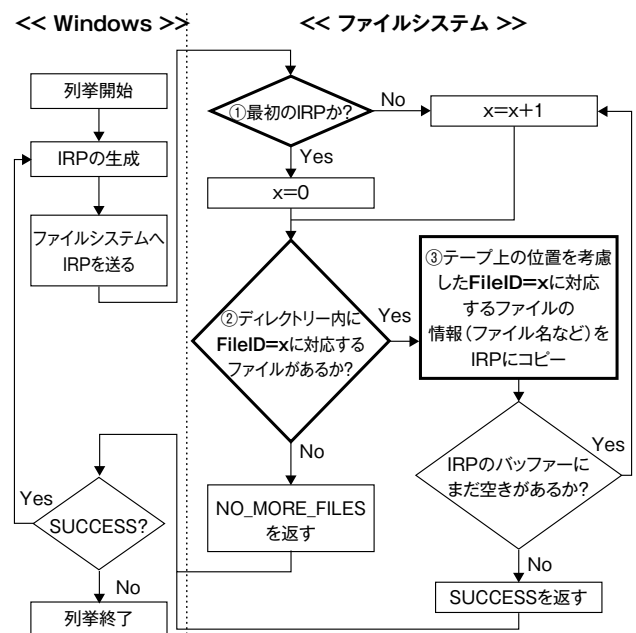


図 8. テープ順列挙法の流れ

れた位置の順で FileID を付けておく。

Windows からの IRP による問い合わせを受けた際に、その IRP が最初の問い合わせによる IRP かどうかを調べる(図 8 の①)。最初の IRP であれば、内部の変数 x を 0 とし、2 回目以降の問い合わせによる IRP であれば、x を 1 だけ増やす。

次に、ファイル列挙の対象のディレクトリー内に、FileID=x に対応するファイルがあるかどうかを調べる(図 8 の②)。FileID=x に対応するファイルがあれば、ファイル名などの情報を IRP にコピーする(図 8 の③)。

情報のコピー後、x を 1 だけ増やし、IRP のバッファーに空きがあれば図 8 の②へ戻る。IRP のバッファーに空きがなければ、Windows へ処理を返す。

この実装により、テープ上の並びの順に列挙することができる。

5. テープ順列挙法の評価と考察

5.1 評価

テープ順列挙法を LTFS Windows 版に適用し、表 2 と同じディレクトリーを xcopy コマンドで LTFS から HDD へリストアした。このときのリストアにかかった時間を測定した結果を表 3 に示す。また、従来の列挙方法を使ったときの転送速度の比較を図 9 に示す。

テープ順列挙法を用いることで、一番効果の少ない 100MB のファイル 10 個をリストアした場合でも、転送時間は約 90% 短縮され、一番効果のあった 5MB のファイルを 200 個リストアした場合には、転送時間は約 99% 短縮された。このとき、転送速度はそれぞれ、9.6 倍、115 倍である。

表 3. テープ順列挙でのリストアにかかる時間

ファイル・サイズ	ファイル数	コピー時間
100MB	10	12 秒 23
50MB	20	14 秒 79
10MB	100	16 秒 02
5MB	200	16 秒 05
1MB	1,000	31 秒 80

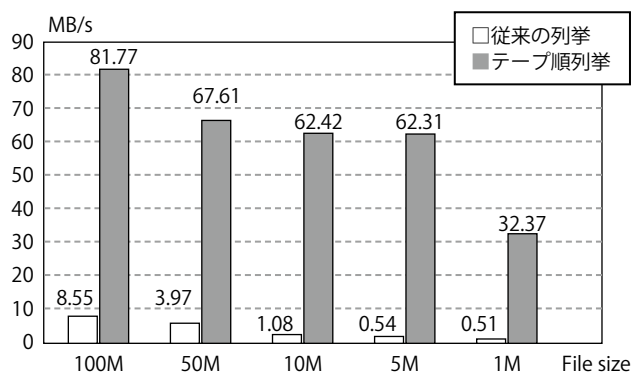


図 9. 転送速度の比較

5.2 HDDの影響について

xcopy コマンドでの複数ファイルのコピーは、1 ファイルずつコピーを行う。そのため、リストアにかかる時間は、LTFS からの読み出しだけではなく、HDD への書き込み時間も影響する。

今回の測定環境での影響を確認するため、メモリーから HDD 上の 1 つのディレクトリーに、複数のファイルを書き込んだ際の書き込み時間を測定した。その結果が表 4 である。

すべてのケースで 1 秒以下であることから、今回の測定環境ではリストア時間の測定に対する影響は十分小さく、測定誤差の範囲に収まると考えられる。

表 4. HDD へのファイル書き込みにかかる時間

ファイル・サイズ	ファイル数	コピー時間
100MB	10	494 ミリ秒
50MB	20	483 ミリ秒
10MB	100	563 ミリ秒
5MB	200	640 ミリ秒
1MB	1,000	977 ミリ秒

5.3 ファイル・サイズと転送速度について

列挙方法によらず、LTFS から HDD へ大きなファイルを少数リストアするのと、小さなファイルを数多くリストアするのでは、転送速度に違いが出ている。これは、以下に説明するように、小さなファイルの数が多くなると、xcopy コマンドではテープ・ドライブから連続してデータを読めなくなるためである。

テープ・ドライブは、データを読み出すためにテープを走行させているが、PC からの読み出し要求がなくなれば走行を止める。実際には、テープ・ドライブの内部にメモリーを持ち、PC の読み出しが連続しなくても、ある程度はメモリーに蓄えながら先読みしている。これによりテープの走行を止めることなく、PC からの読み出しが再開した時には即座にデータを提供することを可能としている。しかし、PC からの読み出しがあまり連続しない場合には、メモリーがいっぱいになってしまうため、テープの走行を止めることとなる。一度テープの走行を止めてしまうと、再度走行を始めるのにも時間がかかるため、結果として PC からの読み出しは遅くなる。

xcopy コマンドは、ファイルをコピーする前に、ファイルごとに、ファイルシステムへサイズの問い合わせなどを行う。そのため、1 つのファイルを読み出した後、次のファイルを読み出すまでの間に数ミリ秒の時間がかかる。ファイルのサイズが小さいと、テープ・ドライブのメモリーに蓄えられたデータを少ししか読まずに、次のファイルのコピーのためのファイルシステムへの問い合わせをすることになる。その結果、テープの走行を止めてしまい、転送速度が遅くなる。

1MB のファイルを 1,000 個リストアした場合には、テープ順列挙法の転送速度は、従来の列挙方法に比べて 63 倍程度と、5MB のファイル 200 個をリストアした場合の 115 倍よりも効果が少ない。これは、1MB のファイルでは、テープ順列挙法でも 5MB のファイルをリストアしたときほど高速な読み出

しを行うことができず、従来の列挙方法との差が明確にならなかったものと考えられる。

5.4 ファイルの分断について

今回の実装では、ファイルの先頭がテープ上に書かれた位置によって、列挙の方法を変える実装を行った。

ファイルの一部を変更するようなアプリケーション（例えば、Microsoft Word など）によって、ファイルの一部が更新された場合には、図 4 で示したように、ファイルのデータが分断されてしまう。このとき、テープ順列挙法を使っても、A → B → C の順で列挙される。そのため、xcopy コマンドで A をリストアする際には、図 10 のように①→②→③→④→⑤の順で頭出しと読み出しを行う必要があり、リストア時間が長くなる可能性がある。ただし、この場合でも、従来のファイル作成時間順による列挙方法と同じ順番でのリストアとなるため、従来の列挙方法に比べて遅くなることはない。

分断されているファイルを高速にリストアする方法として、分断されているファイルを別のファイルとして扱うという方法が考えられる。

例えば、図 10 の例では、ファイル A を 3 つのファイル A~1 (①の部分)、A~2 (③の部分)、A~3 (⑤の部分) というファイルに分割されているものとして扱う。ファイルの列挙時に、テープ上での並び順だけではなく、テープ上分断されているかどうかを判断して A~1 → A~3 → B → C → A~2 と列挙することで、A~1、A~3、B、C、A~2 という 5 つのファイルがあるものとして扱うことができる。

リストアする時には、xcopy コマンドで HDD へリストアし、その後、~ の付いたファイルを HDD 上で番号の順に結合する。これにより、分断されているファイルがある場合でも、高速にリストアすることができる。

この分断されているファイルを別のファイルとして扱う方法には次のような課題があり、LTFS の今後の機能拡張として、現在も検討中である。

◆ HDD 上でファイルを結合するために、xcopy 以外にもツールが必要となる

◆ 実際のファイルとは異なるファイル名が列挙されるため、混乱を防ぐために、リストア用の特別なモードで LTFS を動作させる、などの必要がある

Microsoft Word のようなアプリケーションは、HDD のようなランダム・アクセス・デバイスを想定してファイルへのアクセスを行う。そのため、LTFS 上のファイルを直接編集すると、テープ上の複数の位置への位置合わせが頻発するため、実際の運用は難しい（これは LTFS ではオンライン・ヘルプでもガイドしている [10]）。

例えば、xcopy コマンドで /D オプションを指定して、更新されたファイルだけ定期的にバックアップをとる、という運用の場合には、ファイルの更新時にはファイル全体が更新されるため、ファイル・データの分断は発生しない。そのため、テープ順列挙法は従来の列挙方法に比べて有効である。

以上より、どのような場合でもテープ順列挙法は従来の方法と同等、もしくはそれ以上に高速にリストアできていることから、テープ順列挙法が有効であることが確認できた。

6. おわりに

本論文では、ディレクトリー内のファイルを列挙する順番をテープ上の並び順に行う、テープ順列挙法という手法を提案した。また、本手法を LTFS Windows 版に適用し、LTFS から HDD へファイルをリストアする時間を測定し、その効果を検証した。その結果、バックアップされたファイルがすべて更新されているような場合で、リストアする時間を 90 ~ 99% と、大幅に短縮できることを確認した。また、バックアップされたファイルが更新されていない場合や、ファイルがテープ上で分断されている場合など、本手法の効果が小さくなる場合でも、従来手法と同等の時間でリストアできることを確認した。

この結果は、従来 HDD 向けに使われてきた、xcopy コマンドなどを用いたバックアップを LTFS へ適用するに際し、本手法がその高速化に大きく貢献し、有効な手法であることを示している。

謝辞

本論文の執筆にあたり、LTFS 開発メンバーの皆さんに多くの助言をいただきました。あらためて深謝いたします。

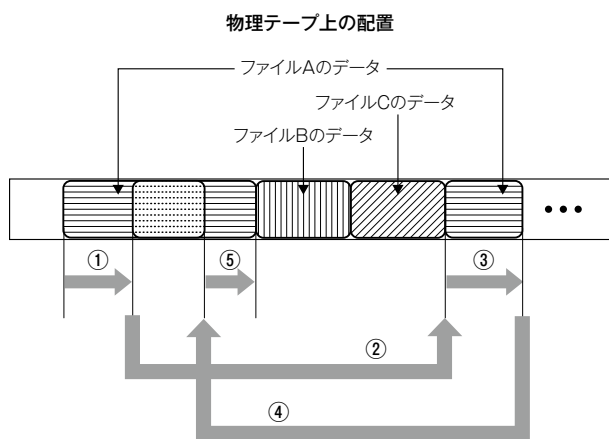


図 10. 分断されたファイルの読み出し順

参考文献

- [1] David Reine and Mike Kahn: "Disk and Tape Square Off Again — Tape Remains King of the Hill with LTO-4" Clipper Notes, TCG2008009 (2008).
- [2] IBM Tivoli Storage Manager Family: http://www-06.ibm.com/software/jp/tivoli/products/storage_mgr/ (2011.11.20).
- [3] David Pease et al.: "The Linear Tape File System" MSST '10 Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), pp.1-8 (2010).
- [4] Ultrium - LTO Technology - LTFS : <http://www.ultrium.com/technology/ltfs.html> (2011.11.20).
- [5] IBM Linear Tape File System 概要: <http://www-06.ibm.com/systems/jp/storage/products/tape/ltfs/> (2011.11.20).
- [6] 日本HP リニアテープファイルシステム: <http://h50146.www5.hp.com/products/storage/solution/ltfs/> (2011.11.20).
- [7] Linear Tape File System : <http://www.quantum.com/Products/TapeDrives/LTOUltrium/LTO-5/LTFS/Index.aspx> (2011.11.20).
- [8] IBM 小規模システム向け災害対策に有効なLTO-5テープとLTFS: http://www-06.ibm.com/systems/jp/storage/solution/lto_ltfs/ (2011.11.20).
- [9] IBM System Storage TS1140 テープ・ドライブ 概要: <http://www-06.ibm.com/systems/jp/storage/products/tape/1140/> (2011.11.20).
- [10] LTFS Information Center: <http://publib.boulder.ibm.com/infocenter/ltfs/cust/index.jsp> (2011.11.20).



日本アイ・ビー・エム株式会社
 大和システム開発研究所
 ストレージシステムズ開発
 アドバイザリー・ソフトウェア・エンジニア

大石 豊 Yutaka Oishi

[プロフィール]

1999年,日本IBM入社。ストレージ関連のソフトウェア開発に従事。2003年より,エンタープライズ・テープ・ドライブ, および, LTO テープ・ドライブのファームウェア開発を担当。IBM Master Inventor.



日本アイ・ビー・エム株式会社
 大和システム開発研究所
 ストレージシステムズ開発
 アドバイザリー・ソフトウェア・エンジニア

今井 直樹 Naoki Imai

[プロフィール]

1985年,日本IBM入社。zOSでのソフトウェア開発, Lotus Notesと高機能コピー機器の連携ソフトの開発に従事。後にストレージ製品のソフトウェア開発を経て現在はテープ仮想化ソフトウェア開発を担当。



日本アイ・ビー・エム株式会社
 大和システム開発研究所
 ストレージシステムズ開発
 スタッフ・ソフトウェア・エンジニア

神谷 昌範 Masanori Kamiya

[プロフィール]

1991年,日本IBM入社。HDD生産のための計測・制御装置の開発に従事。後にストレージ製品のソフトウェア開発を経て現在はテープ仮想化ソフトウェア開発に従事。



日本アイ・ビー・エム株式会社
 大和システム開発研究所
 ストレージシステムズ開発
 アドバイザリー・ソフトウェア・エンジニア

長谷川 徹 Tohru Hasegawa

[プロフィール]

1998年,日本IBM入社。プリンター・コントローラー・ボードの開発を経て,2002年からストレージ製品のソフトウェア開発に従事。現在はLTFSの設計・開発を担当。情報処理学会会員。
 htohru@jp.ibm.com



日本アイ・ビー・エム株式会社
 大和システム開発研究所
 ストレージシステムズ開発
 スタッフ・ソフトウェア・エンジニア

平田 崇将 Takamasa Hirata

[プロフィール]

1997年,日本IBM入社。IAサーバーのシステム管理ソフトウェア開発, ブレード・サーバーの管理モジュール開発を経て,2009年よりストレージ製品のソフトウェア開発に従事。現在はテープ仮想化ソフトウェア開発を担当。