

バージョン管理機能をもつ オブジェクトストレージに適したリストア方式

松本 慎也[†] 根本 潤[†] 荒井 仁[‡]

日立製作所 横浜研究所[†] 日立製作所 RAID システム事業部[‡]

1. はじめに

近年利用が拡大しているクラウドサービスでは、Amazon S3 のようなオブジェクト単位でデータを格納するオブジェクトストレージが用いられる。このオブジェクトストレージに、オフィスのファイル共有に利用するファイルストレージのデータをバックアップするシステムが研究されている。特に Cumulus[1]は、複数ファイルを結合したオブジェクト単位でバックアップするため、任意ファイルをリストアするのに時間がかかる。そのため、ファイルストレージのユーザが破損したファイルを即座に復元する用途には適していなかった。

本論文では、オブジェクトの履歴を保持するバージョン管理機能をもつオブジェクトストレージから、任意世代のディレクトリを瞬時にリストアする方式を提案する。

2. バックアップ・リストアと課題

2.1. システムの概要

本論文が対象とするバックアップシステムは、ファイルストレージとオブジェクトストレージで構成される(図 1)。ファイルストレージはディレクトリとファイルを、オブジェクトストレージはオブジェクトを、それぞれ格納する。これらのストレージはネットワーク接続される。

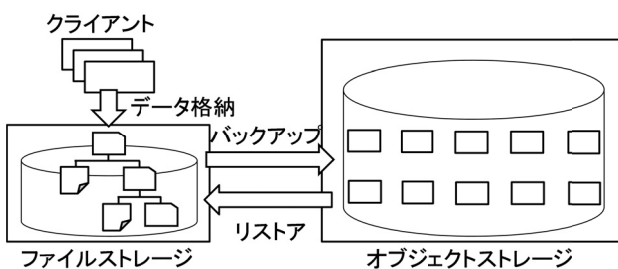


図 1 対象バックアップシステム

ファイルストレージは、オブジェクトストレージに定期的にデータをバックアップする。ディレクトリとファイルをオブジェクトに変換し、オブジェクトストレージに複製する。

ファイルストレージは、複製したオブジェクトからデータをリストアする。オブジェクトを

取得し、ディレクトリやファイルを復元する。

2.2. バックアップ・リストア方式

本論文が前提とするバックアップ・リストア方式(以下、前提方式と呼ぶ)は、ファイルシステム全体のリストア完了を待つことなく、任意ファイルへのアクセスを可能にする。クライアントによるアクセスを契機に、ディレクトリを瞬時にリストアしていくことで実現する。

前提方式によるバックアップ・リストア処理を図 2 に示す。

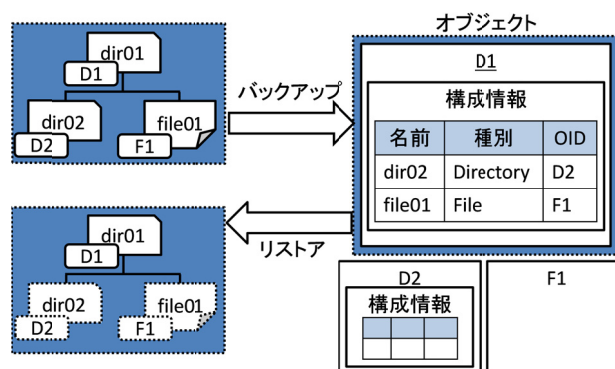


図 2 ディレクトリのバックアップ・リストア

バックアップ処理は、ディレクトリとファイルをそれぞれ 1 つのオブジェクトに変換する。ディレクトリは、その構成情報をもつオブジェクトに変換される。構成情報はディレクトリの配下エントリの情報であり、ファイルストレージ上の名前、ファイル・ディレクトリ種別、OID を含む。OID は、オブジェクト格納時に生成するオブジェクトへのアクセス ID である。ファイルは、その内容をもつオブジェクトに変換される。

リストア処理は、オブジェクトをもとにディレクトリやファイルを復元する。ディレクトリは、オブジェクトに含まれる構成情報をもとに配下エントリを作成し、それらの OID を設定することで復元される。ファイルは、オブジェクトにアクセスし、内容をファイルに格納することで復元される。

2.3. 課題

バージョン管理機能をもつオブジェクトストレージは、オブジェクト格納時に付与されるバージョン ID(以下 VID と呼ぶ)を指定することで、過去のオブジェクトを取得できる。

A Study of Method for Restoring Versioned Objects
[†] Shinya Matsumoto and Jun Nemoto, Yokohama Research Laboratory, Hitachi, Ltd.
[‡] Hitoshi Arai, RAID System Division, Hitachi, Ltd.

このバージョン管理機能を用い、適切なバージョンを取得することで、任意世代のファイルシステムをリストアすることが可能になる。しかし、取得すべきバージョン(VID)を構成情報に格納すると、バックアップ時間が増加する懸念がある。更新されたファイルやディレクトリだけでなく、それを配下に持つディレクトリもまた、バックアップが必要なためである。

そこで、前提方式と同等のバックアップ時間を維持したまま、任意世代のディレクトリを瞬時にリストア可能にすることを課題とする。

3. リスト検索方式

リスト検索方式(以下、提案方式と呼ぶ)は、バックアップ完了日時をキーとしてバージョンリストを検索することで VID を取得し、任意世代のディレクトリとファイルをリストアする方式である。バージョンリストはオブジェクトの全バージョンの格納日時と VID の対応を記録するリストである。バックアップのたびに更新される VID をファイルシステムに格納する代わりに、定期バックアップ完了日時を格納することでファイルシステムの更新を減らし、バックアップ時間の増加を防ぐ。

提案方式によって任意世代のディレクトリをリストアする処理を図 3 に示す。

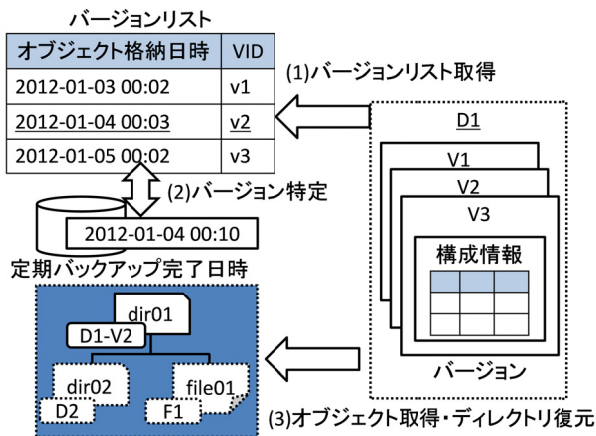


図 3 任意世代のディレクトリリストア

バックアップ処理は、前提方式の手順でバックアップ対象の変換・複製を行った後、バックアップ完了日時を保存する。

リストア処理は、バージョンリスト取得、バージョン特定、オブジェクト取得を行った後、前提方式の手順で復元を行う。まず、OID を指定し、バージョンリストを取得する。次に、バージョンリストに記録されたオブジェクト格納日時と定期バックアップ完了日時を比較し、最も近い過去の VID を特定する。最後に、取得した VID を指定して過去のオブジェクトを取得し、

復元する。

4. 評価

前提方式と提案方式を実装したバックアップシステムを構築し、それらのバックアップ時間・リストア時間を評価する。図 4 に示すファイルシステムのバックアップ時間・リストア時間を測定した。各ファイルのサイズはオフィス環境を想定し 160KB とした。

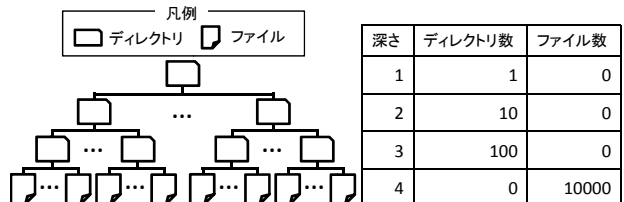


図 4 ファイルシステム

10 回測定した結果の平均を図 5 に示す。

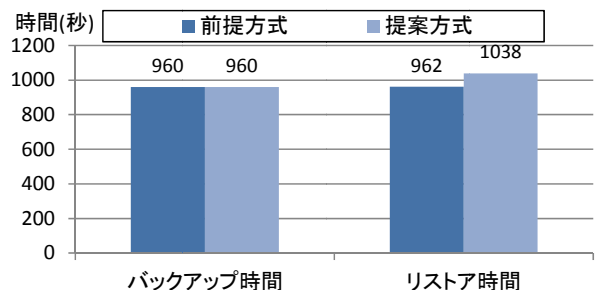


図 5 測定結果

バックアップ時間は両方式ともに同等であった。提案方式がバックアップ完了日時の保存に要するディスク I/O 量は、ディレクトリやファイルの変換・複製に要する I/O 量に比べてごく少ないためと考えられる。

リストア時間は提案方式が前提方式よりやや長く、その差は 76 秒であった。したがって、任意世代のファイルをリストアするためのオーバーヘッドは 7.6 ミリ秒であり、クライアントによるアクセスへの影響は十分小さいと考えられる。

5. おわりに

本論文では、オブジェクトストレージがもつバージョン管理機能を利用し、任意世代のディレクトリを瞬時にリストアする方式を提案した。前提とした方式と同等のバックアップ時間を維持したまま任意世代を瞬時にリストア可能なことを確認し、本方式の有効性を示した。

6. 参考文献

[1] Vrable, M., Savage, S., and Voelker, G.M.: Cumulus: Filesystem Backup to the Cloud, FAST, pp.225-238 (2009).