

高速差分抽出方式による非同期リモートバックアップの効率化

根本 潤[†] 須藤 敦之[†] 中村 隆喜[†]

(株)日立製作所 システム開発研究所[†]

1. はじめに

事業継続性の確保を目的として、ファイルサーバやデータベースシステムなどの各種 IT システムに、ディザスタリカバリ機能を求めるニーズがある。

ディザスタリカバリ機能とは、災害などにより被害を受けたシステムを代替および復旧するための機能のことをいう。その実現手段の 1 つとして、リモートバックアップがある。

リモートバックアップでは、システムを運用している正サイトのデータを、地理的に離れた副サイトにコピーしておく。これにより、災害発生時、正サイトでの運用が継続不可能になった場合に、副サイトのデータを使用して運用を引き継いだり、正サイトのデータを復旧したりすることができる。

リモートバックアップ中の正サイトの負荷を抑えたい場合、正サイトの更新を常時副サイトに反映する同期方式ではなく、定期的に一括して反映する非同期方式が用いられる。

NAS システムの非同期リモートバックアップでは、ファイルシステムの静止化イメージであるスナップショットの単位でバックアップを行う。リモートバックアップが開始されると、正サイトのバックアップ対象スナップショットと、副サイトへ既にバックアップ済みスナップショットで差分が発生している箇所の位置情報を抽出し、その情報を元に、差分データのみを副サイトに転送する。

このような従来の差分抽出方式には、次の 2 つの課題があった。(1) 差分抽出に必要なディスク上のスナップショット管理情報が大きいと、読み込み時間に時間を要し、差分抽出時間についてはバックアップ時間が増大する。(2) 差分抽出中の通常ファイルアクセス性能が低下する。

そこで、本研究では、これらの課題を解決する高速差分抽出方式を提案し、非同期リモートバックアップの効率化を図る。

2. 従来の差分抽出方式と課題

2.1 スナップショット機能

はじめに、従来の差分抽出方式の前提となるスナップショット機能^[1]について述べる。

スナップショットとは、ある時点のファイルシステムの静止化イメージである。スナップショットは、運用ボリューム、差分格納ボリュームという 2 つのボリュームによって再現する。運用ボリュームは、現時点のファイルシステムそのものであり、差分格納ボリュームは、現時点のファイルシステムとスナップショットとの差分データを格納するボリュームである。

運用ボリュームへの更新は、更新要求があるたびに、ビットマップを参照し、運用ボリューム上のデータが差分格納ボリュームに退避済みであるかどうかを確認する。未退避であった場合には、運用ボリューム上の更新前のデータ（差分データ）を差分格納ボリュームに退避した上で、運用ボリュームの更新を行う。この処理を Copy On Write (COW) 処理と呼び、前述のビットマップを COW ビットマップと呼ぶ。退避済みであった場合には、そのまま運用ボリューム上のデータを更新する。そして、差分格納ボリュームに退避した差分データと、運用ボリューム上にある更新されていないデータを組み合わせて、ある時刻のファイルシステムの状態、すなわちスナップショットを再現する。

2.2 従来の差分抽出方式

従来の差分抽出方式では、差分データの退避先を管理するテーブルである COW テーブルを用いる。COW テーブルは、差分格納ボリューム中に格納している。COW テーブルの例を図 1 の左に示す。図中の P_n は、運用ボリュームをチャンクと呼ぶ単位で分割したときの、 n 番目のチャンクを表す。また、 S_k は、 k 世代目のスナップショットを表す。COW テーブルでは、運用ボリューム上のチャンク P_n のデータを、差分格納ボリュームのどのチャンクに退避したか、各世代のスナップショット S_k ごとに記録している。なお、退避されていないデータで、まだファイルシステム上にあるデータは 0 番と表現する。

この COW テーブルを参照し、当該スナップショットのすべてのチャンクについて比較を行う。そして、退避先が同じか、どちらも未退避の場合は、差分なし(0 と表現)、退避先が異なるか、どちらか一方が未退避の場合は、差分あり(1 と表現)とすることで、スナップショット間の差分の有無と位置情報を示す差分ビットマップが作成できる(図 1 右)。

リモートバックアップを行う際には、はじめにこの差分ビットマップを作成し、それにもとづき、正サイトから副サイトへ差分があるチャンクのデータを転送する。

ところが、上述の差分抽出方式では、COW テーブルをディスクから読み込む必要があり、この処理に時間を要して実際のデータ転送開始が遅れていた。加えて、差分抽出中のファイル書き込み性能が、通常時に比べて低下していた。

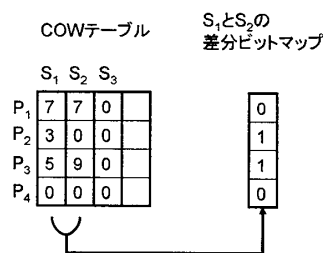


図 1 従来の差分抽出方式

3. 解決方式

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 解決方式(a)：帯域保証付き事前抽出方式

帯域保証付き事前抽出方式は、バックアップ対象スナップショットが作成されてからリモートバックアップが開始されるまでの間に、事前に差分抽出処理を行い、差分ビットマップを作成しておく方式である。これにより、リモートバックアップを行う際に、即時データ転送を開始する。

ただし、このように事前に差分抽出を行っても、ファイル書き込み性能に与える影響は変わらない。そこで、差分抽出時に COW テーブルを読み込む量を一定量に制限することで、ファイル書き込み性能の帯域を保証する。

3.2 解決方式(b)：COW ビットマップ応用方式

COW ビットマップ応用方式について述べるにあたり、まず、COW ビットマップと COW テーブルの関係について図 2 を用いて説明する。

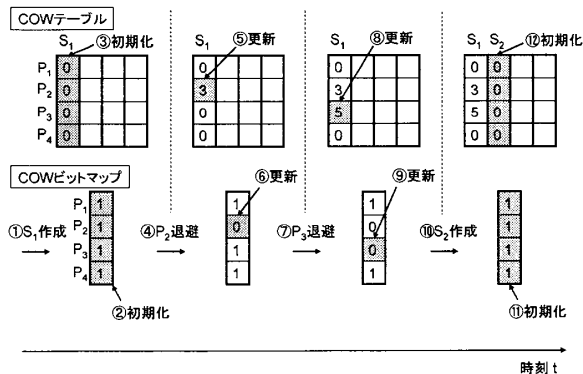


図 2 COW ビットマップと COW テーブルの関係

スナップショット S1 の作成が指示されると、②COW ビットマップを初期化し、また、③COW テーブルを初期化する。COW ビットマップは初期化により、全てのチャンクのビットを 1 (COW が必要な状態) とする。COW テーブルは初期化により、全てのチャンクを 0 番 (当該チャンクのデータが運用ボリュームにある状態) とする。次に運用ボリューム上のチャンク P2 に書き込み要求があると、④P2 のデータを差分格納ボリューム上の空領域 (図 2 の例では 3 番) に退避し、⑤COW テーブルのチャンク P2 を 3 番に更新する。この時点で、チャンク P2 は COW が不要となるため、⑥COW ビットマップのビットを落とす。同様に、P3 に書き込み要求があった場合、⑦～⑨の処理を行う。また、次の世代のスナップショットの作成が指示されると、①～③と同様に、⑩～⑫の処理を行う。

ここで、スナップショット S2 を作成する直前の COW ビットマップ (1001) とスナップショット S1 と S2 の差分ビットマップ (0110) について見てみると、両者は反転の関係にあることがわかる。これは、スナップショット S1 を作成してから COW 処理を行った箇所が、更新処理を行った箇所、すなわち、差分の発生した箇所であることに起因する。したがって、スナップショット S2 を作成し、スナップショット S1 と S2 の差分が確定した際の COW ビットマップの反転が、スナップショット S1 と S2 の差分

ビットマップと同一となる。

よって、スナップショット作成する際、COW ビットマップを初期化する直前に退避しておけば、リモートバックアップを開始する際に反転して、差分ビットマップとして使用することができる。本方式は、ディスクアクセスを必要とせず、メモリ上の処理だけで差分ビットマップが作成可能なため、非常に高速で、かつ、ファイル書き込み性能に影響を与えないものと考えられる。

4. 評価

従来方式、帯域保証付き事前抽出方式(方式(a))、COW ビットマップ応用方式(方式(b))を、リモートバックアップ開始時の差分抽出時間と差分抽出中のファイル書き込み性能の低下率で評価する。

3 方式の評価結果を表 1 に示す。従来方式と方式(b)については、512GB ファイルシステムの環境で、プロトタイプを用いて実測した結果である。方式(a)については、従来方式と方式(b)の実測結果をもとに、性能低下率を 15% に設定して机上計算した値である。

表 1 評価結果

		従来方式	解決方式	
			方式(a)	方式(b)
差分抽出時間	通常	30 秒	1 秒	1 秒
	最大		120 秒	
性能低下率		60%	15%	13.2%

評価の結果、方式(a)と方式(b)は、通常時の差分抽出時間が、約 1 秒であり、差分抽出中の性能低下率は約 15% であった。いずれの方式も従来方式の課題を解決していることがわかる。

しかしながら、方式(b)では、バックアップ対象スナップショットが作成されてからリモートバックアップが開始されるまでの間に事前抽出が完了していなかった場合、リモートバックアップ開始時の差分抽出時間が最大で 120 秒となってしまう。よって、方式(b)の COW ビットマップ応用方式がより優位であるといえる。

5. おわりに

本研究では、非同期リモートバックアップの効率化を目的として、低負荷で高速な差分抽出方式を検討した。COW ビットマップを差分ビットマップとして応用する方式を提案し、ファイル書き込み性能の低下を抑えた上で、差分抽出時間を大幅に短縮した。

6. 参考文献

- [1] 中野隆裕, 山崎康雄, 藤井直大: RAID システム内蔵型 NAS (2) 一多世代スナップショット機能一, 情報処理学会第 66 回全国大会講演論文集, pp. 1-33