



MIC Associates, Inc.

スナップショットとVSS

エムアイシー・アソシエーツ株式会社

2009/02/16

ここに記載された内容は更新される可能性があります。この文書に記載されている内容はこの文書の発行時点におけるエムアイシー・アソシエーツ株式会社の見解を述べたものです。エムアイシー・アソシエーツ株式会社が、この文書に記載された内容の実現に関して確約するものではありません。また発行日以降については、この文書に記載された内容の正確さは保証しません。

この文書は情報の提供のみを目的としており、明示的または黙示的に関わらず、この文書の内容について エムアイシー・アソシエーツ株式会社はいかなる保証をするものでもありません。

エムアイシー・アソシエーツ株式会社は、本書に記載してあるすべて、または、一部の記載内容に関し、許可なく転載、または、引用することを禁じます。

VSSはMicrosoft社の登録商標です

その他、記載されている会社名、製品名は各社が所有する商標の場合があります。

バージョン	作成日付	旧バージョンからの 変更点	総ページ数
1.0	2009/02/16	—	13

本書作成、編集、管理



エムアイシー・アソシエーツ株式会社
〒103-0004 東京都中央区東日本橋3-12-12
櫻正宗東日本橋ビル9F
Tel. 03-5614-3757 Fax. 03-5614-3752

2009/02/16

目次

はじめに

データバックアップの目的

バックアップ運用の制約

バックアップ運用の問題点

スナップショットとバックアップ

COWスナップショットの仕組み

COWスナップショットの長所と短所

スナップショットクローニングの仕組み

スナップショットクローニングの長所と短所

スプリットミラーズスナップショットの仕組み

スプリットミラーズスナップショットの長所と短所

スナップショットバックアップの運用と課題

エージェント方式によるスナップショットバックアップ

エージェント方式の問題点

VSSによるスナップショットバックアップ

VSSの特徴

Xyratex社製ストレージのVSS対応

はじめに

最近、コンテンツの品質向上や業務の IT 化の浸透によって、システムの必要とするデータストレージの容量は増えるばかりです。数年前までギガバイトのオーダーで十分であったシステムが、当たり前のようにテラバイトのストレージを要求するようになり、ペタバイトオーダーへのニーズも見受けられるようになってきています。

技術革新によるドライブの大容量化も進んだおかげで、大容量ストレージの導入事例はどんどん増えていますが、それにとまって大容量ストレージシステムを活用あるいは運用する上での課題も顕在化してきています。

課題のひとつにバックアップがあります。大容量化したストレージを従来の手法でバックアップをとるのは、限界になりつつあります。

今回は、これからの大容量ストレージのバックアップについて重要な技術であるスナップショットと VSS について紹介します。

データバックアップの目的

データバックアップには、大きくふたつの目的があります。1つはシステム障害時に復旧させるためのバックアップであり、もうひとつは業務的理由または人為的ミスに備えて過去のデータを参照することを目的としたバックアップです。後者は世代管理、又はアーカイブなどと呼ばれることもあります。

この2者は、データを保存するという点では共通しますが、前者は、システムの障害に備えてできるだけ最新の状態(データとシステム)に戻すことが目的であるのに対し、後者は過去のデータに遡って活用することを主目的とする点で異なります。また、前者は、システムの可用性維持のための複数のソリューションの中のひとつであり、バックアップ以外のソリューションとの兼ね合いで議論が必要になります。

表1 バックアップの目的と比較

	障害復旧目的のバックアップ	世代管理目的のバックアップ
バックアップ対象	ユーザーデータ以外にシステムパラメータ、アプリケーションなどが含まれる場合がある。	通常ユーザーデータのみ。
バックアップ世代数	原則1世代あればよい。	目的に応じて複数世代保存する必要。
バックアップサイクル	可能な限り細かいサイクル。	業務上必要と判断されるサイクル。
リストア時間要件	可能な限り短時間。	業務上支障のない範囲。

しかしこれまでのシステム運用では、障害復旧用に取得したバックアップデータを長期保存することで、世代管理用バックアップと兼用する運用形態が多かったため、バックアップとして一緒に議論されてきた経緯があると思われます。

バックアップ運用の制約

従来は、バックアップは業務時間終了後の夜間や週末などにおこなうケースがほとんどでした。業務中におこなわない理由は、データバックアップ中にデータの更新が発生すると、保管したデータに不整合が発生する可能性があるからです。

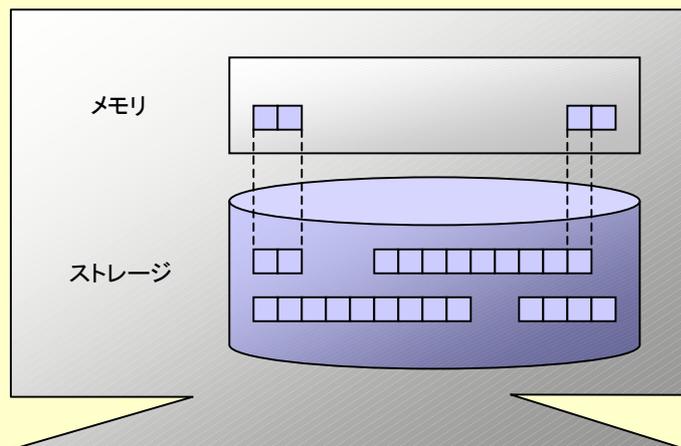
たとえば、アプリケーションが稼動していてファイルをオープンしているときなどは、アプリケーションがメモリ上に持っているデータとディスクに書き込まれたデータが混在するため、ディスクのデータだけをバックアップするとデータが不整合になってしまいます。

また、複数のファイルが連携して更新される場合などは、それぞれがバックアップされる順序とタイミングによっては、ファイル間の連携が壊れたり、部分的なデータの重複や消失が発生してしまいます。

これらのことから、バックアップはアプリケーションを止めて行うことが必要になり、業務時間外におこなうことが一般化していました。

図1 アプリケーション稼動中のデータの所在

アプリケーションの稼動中は、メモリにデータが残っているため、ディスクのデータだけをバックアップしてもデータが不整合になる可能性がある



バックアップ運用の問題点

最近では、IT化された業務が増え、データ量が増えたことによって、バックアップにかかる時間が長くなりつつあります。一方、オンラインサービス時間の拡大やインターネットサービスの普及などにより、業務を停止できる時間は短くなる傾向があります。また、以前なら電話と紙の伝票で対処できたものが、IT化が進むにつれてシステムなしには業務サービスの品質を維持することが困難になってきています。さらに、データの大容量化に伴いストレージを構成する HDD ドライブの数が増え、ストレージまわりの障害頻度も増える傾向にあります。

その結果、より大量のデータをより短いダウンタイムでより確実にバックアップすることがより重要になり、それを実現することが頭痛の種になっているシステムの運用担当者が増えているのではないのでしょうか。

この問題を解決する技術のひとつが、スナップショットです。

スナップショットとバックアップ

スナップショットは、ここまで述べてきたバックアップの問題点を解決するキーテクノロジーのひとつです。スナップショットを実行すると、極めて短い時間でその時点でのデータイメージを残すことができるので、オンライン業務を中断することなく(あるいは非常に短時間の中断で)、整合性のあるバックアップをとることが可能になります。

スナップショットは、実際に更新され続けるデータと別にスナップショットを取った時点から更新されることのないイメージデータを複製するところから、シャドウコピーと呼ばれることもあります。

ストレージの管理している数百 GB から数 TB のデータを瞬時に別メディアに保管するのは技術的には不可能です。スナップショットではデータの移動なしに、(あるいはデータの移動を極力減らして)スナップショットイメージを生成する仕組みを用意しています。また、スナップショットは、実行の主体がソフトウェアかハードウェアか、また実現する手法においていくつかの種類があり、それぞれに長所と短所があります。以下に、そのうちのいくつかを紹介します。

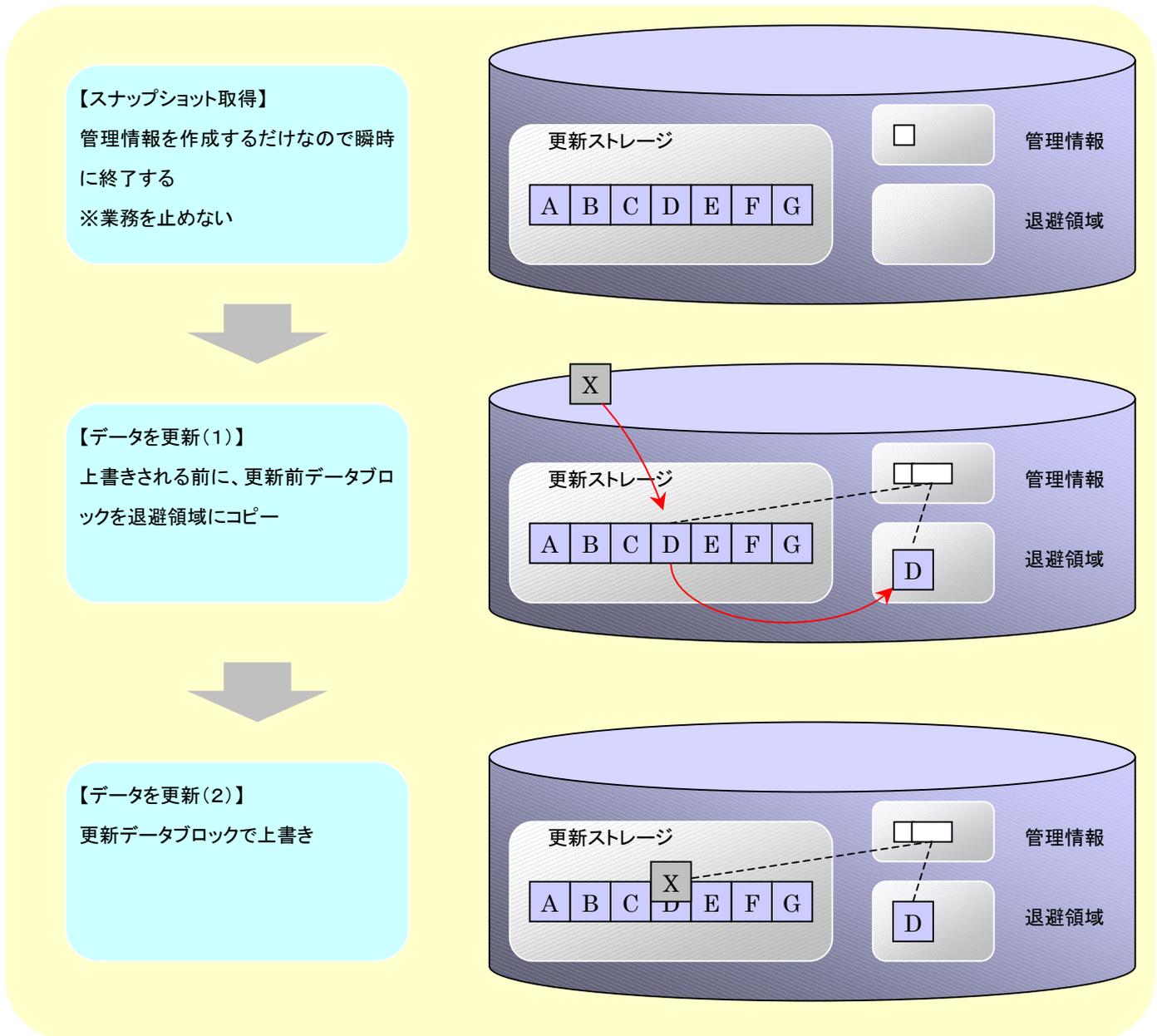
COWスナップショットの仕組み

COW (Copy On Write) スナップショットは、スナップショットを取った時点ではデータに対して何も処理をおこないません。スナップショットのタイムスタンプなどの管理情報を残すだけなのでスナップショットの取得は極めて短い時間に完了します。しかしこのままではデータの更新が発生するとスナップショットを取った時点のデータが上書きされてしまい、あとから参照できなくなります。

COWスナップショットでは、スナップショットを取得後にデータを更新する際、更新前のデータブロックを別の場所に退避し、その退避前の場所についての管理情報を残します。そのあとで、通常と同じデータの更新(データブロックの上書き)を実行します。

COW スナップショットには、RAID 装置などのハードウェアで実現するものとバックアップツールやOSなどのソフトウェアで実現するものの両方があります。

図2 COW スナップショットの処理の流れ



更新ストレージは最新の状態になっているので、普通にアクセスすれば最新のデータを参照できます。過去のスナップイメージを見る場合には、データブロックを参照する際に管理情報から退避したデータブロックの有無を判断し、退避データブロックがあればそれを参照し、なければ最新データブロックを参照します。退避データと最新データを組み合わせてアクセスすることでスナップショット取得時点のデータイメージを参照することを可能にしています。スナップショットイメージは仮想のストレージなので、読み取り専用であり更新することができません。

また、退避領域のデータブロックを管理情報に従って元あった場所に戻すことによって、過去のデータに完全に戻すことができます。これをスナップバックと呼びます。

COWスナップショットの長所と短所

COWスナップショットは、スナップショットイメージを瞬時に生成できるので、オンライン業務やアプリケーションの停止時間によって発生する業務のダウンタイムを最小にできます。また、更新されるデータのみ退避するので、ストレージの容量が少なくすみます。さらに、多くの世代のスナップショットをとることができ、任意の世代のデータを参照したり、任意の世代のデータにスナップバックできることも、大きな特徴です。

その反面、スナップショット取得後の更新処理の時に、更新前のデータブロックを退避領域にコピーする処理が増えるため、ストレージへの書き込み性能が低下します。また、退避した情報だけではデータを完全にリカバリすることができないため、更新されているデータストレージに障害が発生した場合データを復旧することができません。通常は、スナップショット取得後、スナップショットイメージを別のメディアにコピーすることによって初めてデータの冗長化が完了します。しかし、オンライン業務とスナップショットイメージからのバックアップコピーを並行しておこなうと、ただでさえ遅い更新処理がさらに遅くなります。

スナップショットクローニングの仕組み

スナップショットクローニングは、前出のCOWスナップショットの発展形です。COWスナップショットとそのあとのスナップショット実体化をハードウェアで一連の処理でおこないます。

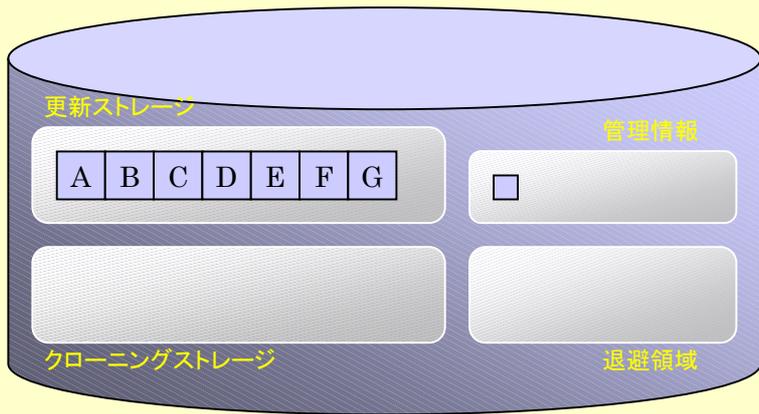
スナップショットクローニングの場合、スナップショット用の退避領域と別に、クローニング用のストレージが必要になります。クローニングを実行すると、まずクローニング用スナップショットを取得します。取得が完了するとスナップショットの終了を応答します。ここまでは、COWスナップショットと同じになります。

このあと、ストレージは、実データストレージからクローニング用ストレージへ物理ブロックコピーを開始します。このとき、システムはスナップショットの終了を受信しているので更新処理が再開されていますが、ブロックコピー処理は、実データストレージへのスナップショット後の更新の有無にかかわらず、全ブロックを順にコピーします。

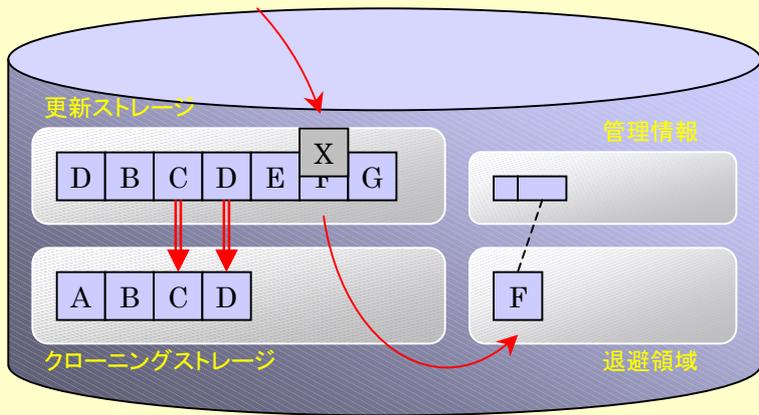
コピーが完了すると、クローニングストレージにはスナップショット後の更新前のブロックと更新後のブロックが混在した不整合な状態になっています。そこで、最後にクローニング用スナップショット退避データを、クローニングストレージにスナップバックすることによって、クローニングストレージをスナップショット取得時の完全なコピーにします。

図3 スナップショットクローニングの処理の流れ

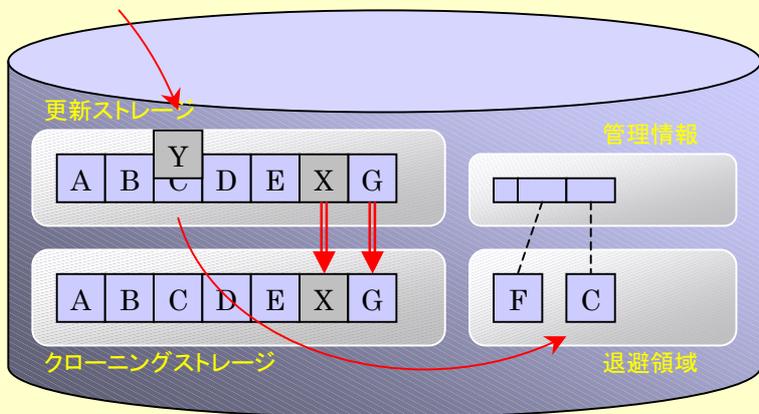
【スナップショット取得】
クローニング用のスナップショットを取得する



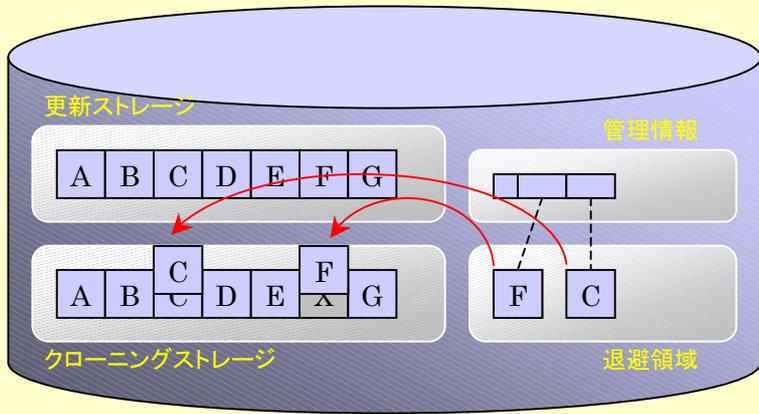
【コピー開始】
更新ストレージからクローニングストレージへデータブロック連続コピー
※データの更新は再開されている



【コピー完了】
データブロックコピーが完了した時点では、スナップショット後の更新前データと更新後データが混在している



【スナップバック】
クローニングストレージへスナップバックしてクローニングが完了する



スナップショットクローニングの長所と短所

COWスナップショットの場合、スナップショットイメージは仮想ストレージとして作成されるので、障害に備えるためには物理メディアへのコピー処理が必要になります。しかし、物理ストレージへの更新処理と仮想ストレージへのアクセスが並行すると、かなりの長時間にわたり、オーバーヘッドが大きくなる問題がありました。

スナップショットクローニングの場合、コピー処理はハードウェアが無判断でおこなうため、最後のスナップバック処理の間を除き、更新処理の性能低下が軽減されます。また、クローニングが完了すれば、物理的には別のドライブから構成されるので、更新ストレージとクローニングストレージへの並行アクセスはコントローラハードウェアの性能の許す限り高性能になります。しかも、COWクローニングの機能も共存できるので、必要に応じて仮想ストレージのスナップショットイメージと物理ストレージのスナップショットクローニングを使い分けることもできます。

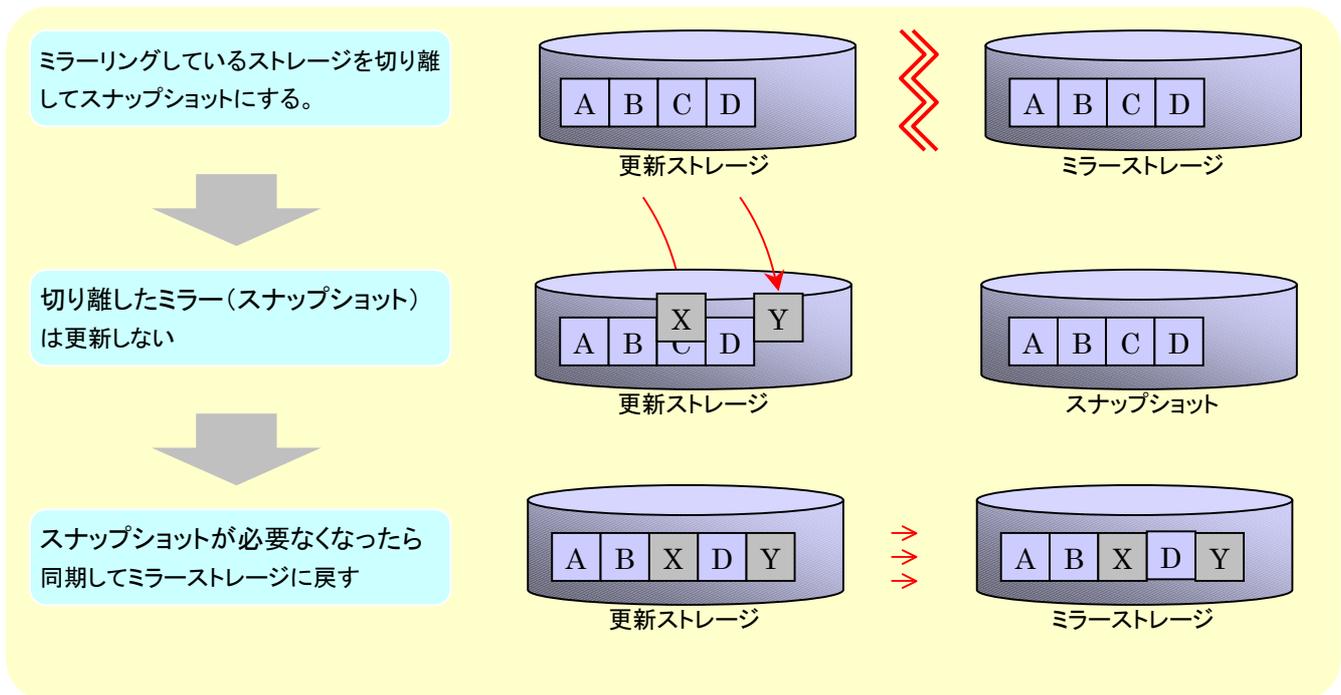
そのかわり、スナップショットクローニングの場合には、スナップショット退避エリアと別にクローニング用ストレージが必要なため、はじめからデータの倍以上のストレージを用意する必要があります。また、現時点では対応しているハードウェアが限定されてしまうという問題もあります。

スプリットミラーズスナップショットの仕組み

スプリットミラーズスナップショットは、あらかじめミラーリングされているストレージを切り離すことによって、瞬時にデータイメージを固定するものです。スナップショット取得の際に一切処理が不要なため、スナップショットの取得に一番時間がかかりません。

切り離したストレージは、スナップショットとしての役目を終えたらミラー元のストレージと同期することによって、再度ミラーリングストレージとして使用できます。

図4 スプリットミラーナップショットの処理の流れ



スプリットミラーナップショットの長所と短所

スプリットミラーナップショットは、他のスナップショットより高速に、しかも実ボリューム上にスナップショットイメージを固定することができます。また、切り離れた後は別ストレージになるので、スナップショット中の性能低下がありません。

その反面、ミラーリングしていることによって常時オーバーヘッドが発生し、性能が低下します。さらに、スナップショットからミラーに戻すときの同期処理中の更新処理は性能の低下が顕著になるでしょう。また、最初から倍以上のストレージが必要になるため、コスト効率は悪くなります。

また、この方法はミラーリング (RAID1) あるいはレプリケーションに対応しているハードウェア、ソフトウェアなら何でも対応できそうに思えますが、実際には切り離れたスナップショット側を正常に認識させることが困難な場合があるので、実運用には注意が必要になります。

スナップショットバックアップの運用と課題

ここまで、紹介してきたスナップショットはそれぞれに特徴はあるものの、どの方式でも非常に短時間でストレージのデータイメージを固定することが可能になり、それをバックアップすることができます。しかし実際の運用を考えた場合、共通する課題があります。

最初に説明したように、システムは常にすべてのデータをストレージに持っているわけではなく、メモリ上に持っている場合があります。稼動しているアプリケーションは、必要に応じてメモリ上にデータを持っていますし、OSもデータをキャッシュしている場合があります。せっかくスナップショットでストレージイメージを固定しても、その時点でのメモリ上のデータは含まれないため、やはりデータに整合性が保障ができません。そこで、スナップショットをとる前には、アプリケーションのメモリとOSのキャッシュのデータをストレージに書き戻す必要があります。

これらを考慮して運用するとなると、以下のような手順になります。

- 1) アプリケーションからのデータの更新を停止する。(または、スナップショット終了まで待たせる)
- 2) アプリケーションのメモリキャッシュをフラッシュする。(または、アプリケーションを停止する)
- 3) OSのキャッシュをフラッシュする。
- 4) スナップショットをとる。(ソフトウェア、またはハードウェア)
- 5) アプリケーションを再開する。
- 6) スナップショットを別のメディアにバックアップする。

しかし、サーバーで稼動しているアプリケーション個々にフラッシュさせたり停止させる手順やフラッシュが正常に完了したことを判定する方法が違うため、運用は煩雑になります。また、性能のよいハードウェアスナップショットを使用しようとしても、ベンダーごとに制御するユーティリティが違うため、管理者がすべてのユーティリティの操作に習熟しなくてはなりません。スクリプトやプログラムで自動化するのも困難になります。

これを改善するため、バックアップツールのなかには、一連の処理をすべてバックアップツールからおこなえるように対応したものがあります。

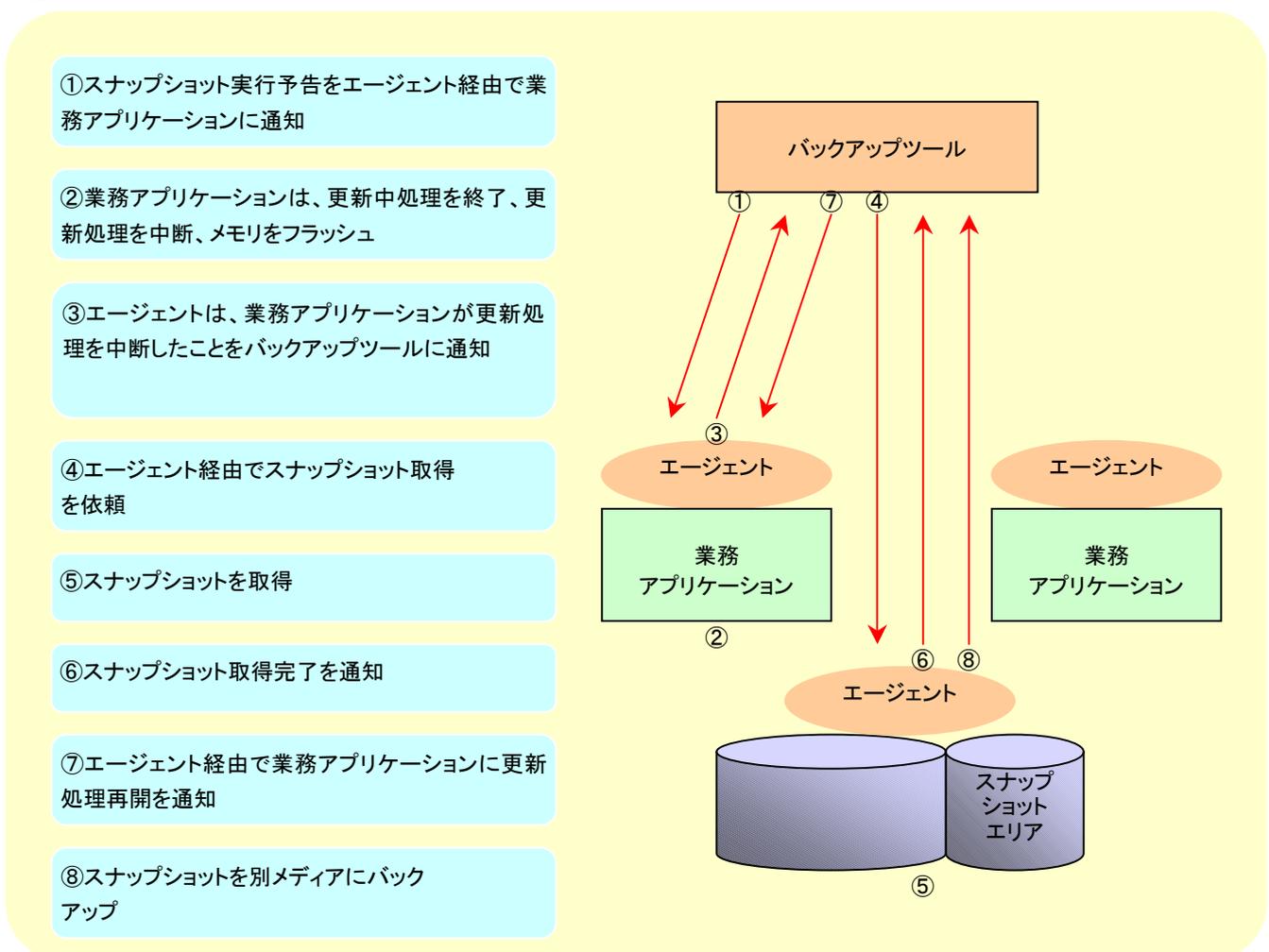
エージェント方式によるスナップショットバックアップ

エージェント方式とは、バックアップツールの側でアプリケーションやハードウェアを制御するためのエージェントを用意し、スナップショットをバックアップするのに必要な一連の処理をバックアップツールから実行できるようにする方法です。具体的には、以下のような処理をおこないます。

- 1) アプリケーションに対応したエージェントが、メモリフラッシュや更新処理の停止、遅延などの処理をアプリケーションにおこなわせます。
- 2) スナップショットをとるハードウェアやソフトウェアに対応したエージェントが、スナップショットを作成させます。
- 3) スナップショット作成が終了すると、アプリケーションに対応したエージェントがアプリケーションの更新処理の再開を指示します。
- 4) 作成されたスナップショットデータをバックアップツールが別のメディアにバックアップします。

これによって、バックアップツールからバックアップ処理の自動実行をスケジュールするだけで、管理者は面倒な運用から解放されることになります。

図5 エージェント方式によるスナップショットバックアップ



エージェント方式の問題点

エージェント方式の場合、バックアップツール側で、様々なアプリケーションとハードウェアすべてに対してエージェントが用意されていないかもしれませんが、バージョンアップまで含めてそれらにすべて対応するのは非常に困難になります。

また、使用する側も、稼動しているアプリケーションと稼動しているストレージハードウェアベンダーすべてにエージェントが用意されているバックアップツールを使用せざるをえなくなり、すべての組み合わせに対応したものがなければ運用での対応が必要となります。新しいアプリケーションやハードウェアを使用する際にも様々な制約が発生することになります。

VSSによるスナップショットバックアップ

VSS (Volume Shadow copy Service) とは、Windows XP および Windows Server 2003 以降に実装されたシャドウコピー (=スナップショット) バックアップのための共通インフラストラクチャです。

VSSは、以下のコンポーネントから構成されます。

表2 VSS を構成するコンポーネント

コンポーネント	内容
VSSコーディネータ	リクエスタ、ライタ、プロバイダの処理を適切に制御して、スナップショットバックアップを管理する。
VSSリクエスタ	バックアップをおこなうアプリケーション。
VSSライタ	ストレージへのデータの更新をおこなうアプリケーション。業務アプリケーション。
VSSプロバイダ	ハードウェアベンダが作成したスナップショット制御用ソフトウェア、またはソフトウェアスナップショットを実行するソフトウェア。

VSS リクエスタがスナップショットバックアップを取得する場合、VSS コーディネータと VSS に対応した各コンポーネントは、協調してスナップショットバックアップを実行するための処理をおこないます。

図6 VSSによるスナップショットバックアップ

【スナップショットの準備】

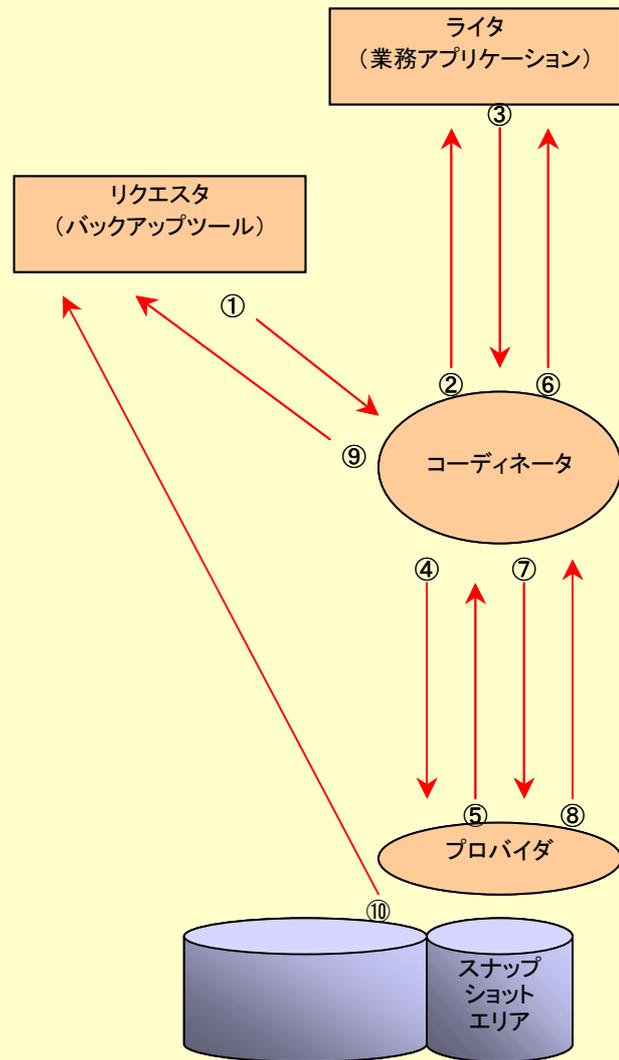
- ①まず、リクエストがバックアップを実行することをコーディネータに通知する
- ②コーディネータはライタに、更新データの完了、更新処理の中断、メモリのフラッシュなどを指示する
- ③ライタは処理を完了するとコーディネータに応答する

【スナップショットの作成】

- ④ハードウェアベンダの用意したプロバイダに指示してスナップショットを作成させる
- ⑤プロバイダはスナップショットが完了したことを通知する
- ⑥コーディネータはライタに更新処理の再開を指示する

【スナップショットのバックアップ】

- ⑦プロバイダにスナップショットイメージをアクセス可能にするよう指示する
- ⑧プロバイダはアクセス可能になったことを通知する
- ⑨コーディネータはバックアップの準備が完了したことを通知する
- ⑩リクエストはスナップショットのバックアップを開始する



VSSの特徴

VSSでは、バックアップツール、アプリケーション、ストレージハードウェアの各ベンダーがVSSの機能に対応さえしていれば、スナップショットバックアップを簡単に実現できます。VSSライタ対応のアプリケーションを使えば、オンラインのままバックアップをとることが可能になります。VSSプロバイダ対応のストレージ装置を使用すれば、高速なハードウェアスナップショットを使用することができます。

Xyratex 社製ストレージのVSS対応

Xyratex 社製 Storage は、COW スナップショット機能(標準 4 世代、オプションで 24 世代)とスナップショットクローニングを実装しています。

また、VSS ハードウェアプロバイダが用意されていて、以下のようなファンクションがサポートされます。

表3 Xratex VSS への対応

機能	内容	使用例
スナップショット取得	スナップショットイメージを取得します	vshadow -p ドライブレター(最大4つまで)
スナップショットリスト表示	取得したスナップショットの一覧を表示します	vshadow -q
スナップショットボリュームの有効化	スナップショットイメージを実ドライブと別のドライブとして見えるようにします	vshadow -el=[スナップショット ID] ドライブレター

Xyratex の COW スナップショット/スナップショットクローニングは、GUI ツールの StorView、CUI インターフェースの XCR から使用できます。