C-002

スケールアウト型ストレージにおけるスナップショットのノード間移行手法の提案
A Study for Inter-Node Migration of Snapshots in Scale-Out Storage

佐藤 賢太† 山賀祐典† 松下貴記† 出口 彰†
Kenta Sato Yusuke Yamaga Takaki Matsushita Akira Deguchi

1. はじめに

近年、複数ノードをクラスタ化するスケールアウト型のアーキテクチャを採用するストレージ製品が増えている[1][2]。このようなアーキテクチャでは、ボリュームやスナップショットのデータ格納領域がノード毎に独立しており、容量や負荷の平準化のために、ノード間でボリュームやスナップショットを移行するリバランスが必須である。

スナップショットは作成元ボリュームや他のスナップショットとデータを共有することで、消費する記憶容量を削減する。そのため、単純にノード間でコピーして移行すると、データ共有関係が失われ、容量消費が増加する問題が生じる。本研究では、移行時にスナップショットのデータ共有関係を特定し、移行先ノードで再構築することで、容量消費の増加を防ぐ移行手法を提案する。

2. スナップショットの基本原理

スナップショットは、ボリュームのある瞬間の論理的な複製を作成するストレージの機能である。スナップショットの取得方式は、Copy-on-Write（CoW）や Redirect-on-Write（RoW）など幾つか存在するが[3]、近年は処理効率のよい RoW が主流であり、本研究でも RoW を対象とする。

図 1 に本研究が対象とするスナップショットのデータ構造の概要を示す。図に示すように、アプリケーションが読み書きするボリュームやスナップショットのデータは、ブロック単位で、容量プールに格納される。容量プールは、HDD や SSD を RAID や Erasure Coding といった技術によって束ねた物理的な記憶領域である。そして、論理－物理のマッピング（論物マップ）により、ボリュームやスナップショットの論理的な記憶領域と、容量プール内の物理的な記憶領域が対応付けられる。

スナップショットの作成は、作成元ボリュームの論物マップを複製することにより実現する。これにより、スナップショットと作成元ボリュームの間で、容量プールに格納されたデータの共有関係を構築でき、ボリュームの論理的な複製を作成できる。ボリュームやスナップショットがライト I/O を受け付けた場合は、図中「C1」のように、元のデータは変更せずに容量プールの未使用領域にデータを書き込んだうえで、論物マップを更新する。

3. スナップショット移行の課題

スナップショットは作成元のボリュームとデータを共有している。このため、前述したスケールアウト型のストレージにおいて、ボリュームを別のノードに移行する際は、移行対象のボリュームから作成されたスナップショットのデータ共有関係を維持して移行しなければ、移行後に記憶容量の消費が増大する問題が生じる。これには、ボリュームとスナップショット間の共有関係を特定し、既に転送済みのデータと重複のない差分データだけを抽出・転送したうえで、論物マップの共有関係を再構築する必要がある。

† 株式会社 日立製作所 研究開発グループ
Hitachi, Ltd. Research & Development Group

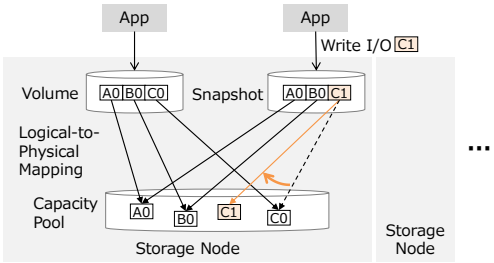

図 1　スナップショットのデータ概要

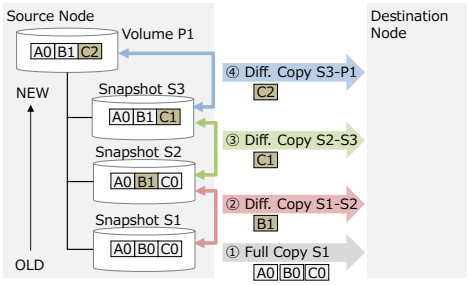

図 2　世代間で探索する手法

4. アプローチと提案手法

本研究では、スナップショットの用途により生じる共有関係の特徴を活用して探索範囲する削減するアプローチをとる。スナップショットの主要な用途は、バックアップと Virtual Machine（VM）デプロイ、DevTest である。そのうちバックアップと VM デプロイは、ボリュームとスナップショットの一方にしかライト I/O が発生しないという特徴があるため、これを利用して、移行後の記憶容量の増大を抑えつつ、探索範囲の絞りこみによる探索コスト削減を期待できる。一方、DevTest はこのような特徴がなく、運用次第でボリュームとスナップショットの両方にライト I/O が発生し得る。そこで、本研究ではバックアップ向けの手法と VM デプロイ向けの手法を組み合わせることで、DevTest に適用可能なハイブリッド手法を提案する。

スナップショットの各用途における共有関係の特長と、それを考慮した移行手法を下記に示す。

(a)　バックアップ向け手法（世代間で探索する手法）

バックアップは、データ保護を目的としてアプリケーシ

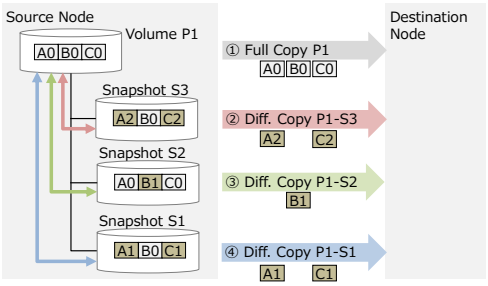

図 3　親子間で探索する手法

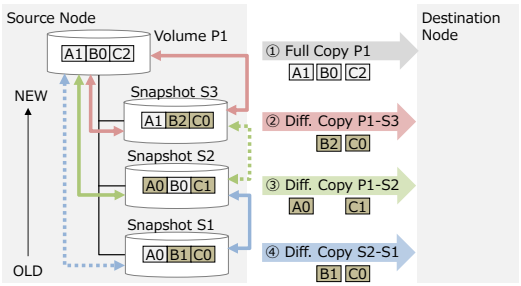

図 4　提案：ハイブリッド手法

ョンが使用するボリュームの複製としてスナップショットを作成する用途である。ライト I/O はボリュームに対してのみ発生し、スナップショットには発生しない。このため、スナップショットの世代間で同一の物理データへの参照有無を探索するだけで共有関係を特定できる。具体的には図 2 に示すように、最初に最も古いスナップショットの全データを転送し、以降はスナップショットの作成世代順に、転送済みの前世代との差分データを抽出して移行先に転送することで移行できる。

(b)　VM デプロイ向け手法（親子間で探索する手法）

VM デプロイは、ボリュームを VM のテンプレートとして扱い、VM をデプロイする際にスナップショットを作成して VM のディスクとして割り当てる用途である。ライト I/O はスナップショットに対してのみ発生し、ボリュームには発生しない。このため、ボリュームとスナップショットの親子間で同一物理データへの参照有無を探索するだけで共有関係を特定できる。具体的には図 3 に示すように、最初にボリュームの全データを転送し、以降は各スナップショットについて、作成元ボリュームとの間で抽出した差分データを転送することで移行できる。

(c)　提案：DevTest 向け手法（ハイブリッド手法）

DevTest ではボリュームとスナップショットの両方にライト I/O が発生し、IO パターンによって、共有関係を特定するための適切な探索対象が変化する。本手法では、世代間と親子間の探索を組み合わせることで対処する。

図 4 に本手法の概要を示す。本手法では最初にボリューム P1 の全データを転送し、その後は、各スナップショット S1~S3 について、世代が新しい順に差分データを抽出して転送することで移行する。差分データの抽出は、世代間と親子間の両方について行い、差分データ量が小さい方を採用する。図の例では、スナップショット S2 について、世代間（S3-S2）で抽出した差分データ（A0、B0、C1）と親子間（P1-S2）で抽出した差分データ（A0、C1）を比較し、差分データ量が少ない親子間の差分データを転送している。これにより、ボリューム P1 と共有関係にある B0 を別データとして再送することを防いでいる。

表 1　評価条件

項目	条件
スナップショット数	64
ボリューム容量	4 [GiB]
データブロックのサイズ	8 [KiB]

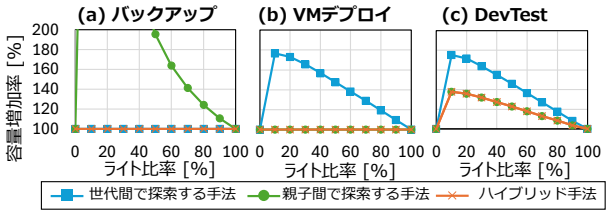

図 5　評価結果

5. 評価

本研究では探索コストと移行時の容量消費を評価する。

まず探索コストについて原理に基づいて評価する。世代間と親子間で探索する手法については、共有関係を特定するための探索対象がボリューム又はスナップショットあたり特定の 1 か所（世代間または親子間のどちらか一方）である。ハイブリッド手法の場合は特定の 2 か所（世代間と親子間の両方）になる。探索コストとしては何れの手法も O (1)と O (N)の線形探索と比べて十分に小さいと言える。

移行時の容量消費については、各用途でのライト I/O 及び各手法による移行をシミュレーション評価した。ライト I/O は、各スナップショットの差分データ量が均等になるように、用途毎にボリューム又はスナップショットに対してランダムに発生させた。その他条件を表 1 に示す。

評価結果を図 5 に示す。横軸は、ボリュームやスナップショットの容量のうち、ライト I/O が発生したブロックの割合を、縦軸は移行による容量消費の増加率を示している。

図から分かるように、ハイブリッド手法は、全ての用途で増加率が最小である。バックアップと VM デプロイでは 100%と増加は発生せず、DevTest でも 138%である。世代間で探索する手法と親子間で探索する手法は、各々対応する用途であれば、増加率は 100%と増加は生じないが、それ以外の用途では大幅に増加する。DevTest に関しては、今回の IO パターンでは親子間で探索する手法の方が増加率を抑えられる結果だが、IO パターン次第で逆転が予想できる。

以上の評価結果より、ハイブリッド手法を使うことで、O (1)という少ない探索コストで、移行による容量消費の増加を 3 手法のなかで最も抑えられることが分かった。

6. おわりに

本研究では、スケールアウト型ストレージを対象とし、スナップショットを有するボリュームをノード間で移行する手法として、世代間と親子間で共有関係を探索する手法を併用するハイブリッド手法を提案した。ハイブリッド手法は、O (1)という少ない探索コストで実現でき、シミュレーションの結果、バックアップや VM デプロイ、DevTest といったスナップショットの主要用途全てにおいて容量消費の増加を他の 2 手法よりも抑えられることが分かった。増加量の更なる低減は、今後の課題である。

参考文献

[1] Hitachi, "Hitachi Virtual Storage Software Block", https://www.hitachi.co.jp/products/it/storage-solutions/products/hybrid_cloud/vsp_one_sds_block/index.html.
[2] Dell Technologies, "Dell EMC PowerStore Breaks Ground in Storage Infrastructure Performance and Flexibility", https://www.dell.com/en-us/dt/corporate/newsroom/announcements/detailpage.press-releases~usa~2020-05~20200505-dellemc-powerstore.htm, May. 2020.
[3] W. Xiao, et.al., "Design and Analysis of Block-Level Snapshots for Data Protection and Recovery," in *IEEE Transactions on Computers*, vol. 58, no. 12, pp. 1615-1625, Dec. 2009, doi: 10.1109/TC.2009.107.