# RAID システム内蔵型 NAS(2) ―多世代スナップショット機能―

Embedded NAS for RAID System (2) -Design of Multiple Generations Snapshot Function-

中野 隆裕[1] 山崎 康雄[2] 藤井 直大[3]
Takahiro Nakano Yasuo Yamasaki Naohiro Fujii

## 1. はじめに

ファイルシステムの瞬間イメージを維持・提供するスナップショット機能は，オンラインで参照可能な一時的なバックアップを実現する NAS の特徴機能の一つである．スナップショットの多世代化は，バックアップ頻度の増加を可能にするため，信頼性を向上させる．

RAIDシステム内蔵型NASは，大容量，高信頼，高可用を特徴とする大型RAIDのNAS拡張である．そのスナップショット機能には，多世代対応に加え，TBクラスのボリュームサイズ対応が必要となった．以下，Linux☆LVM☆☆のスナップショット機能を置き換え，独自に多世代化を実現する方式（多世代化方式）について記述する．

## 2. 基本機能

### 2.1 Linux 標準機能の問題点

Linux には，標準でスナップショット機能を備える LVM がある．この LVM のスナップショット機能は，スナップショット毎に差分領域を用意し，運用ボリューム(P-Vol)への更新アクセスの際，更新前データを差分領域に退避(CoW;Copy-on-Write)する．また，更新部分から，退避した領域にマッピングするハッシュエントリを作成し，ハッシュテーブルに登録する．このハッシュテーブルを用いて，未更新部分を P-Vol に，更新部分を差分領域にマッピングすることでスナップショットを実現する（図 2.1）．

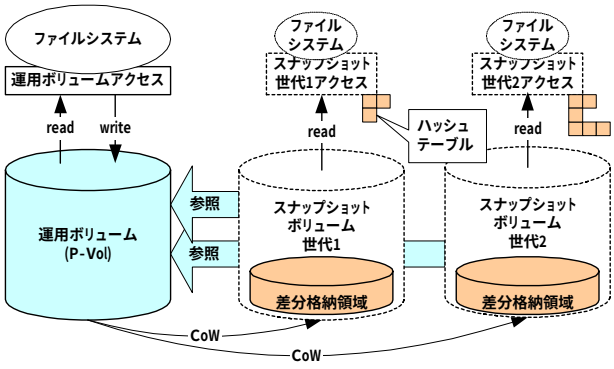


**図 2.1 標準機能の処理方式**

この方式を RAID システム内蔵型 NAS に適用する場合，次の点で問題を生じる．

(1)P-Vol への write 処理で世代数回の CoW 処理が生じる可能性がある

(2)ハッシュであるため全ハッシュエントリをメモリに上に展開しなければならない．

RAID システム内蔵型 NAS では，100 世代のスナップショットを目標とするが，(1)により CoW が最大 100 回発生する方式は非効率的である．また，大容量，多世代のスナップショットを維持するためだけに，大量のメモリ占有は許されない．(2)では，更新量 100GB を 100 世代維持するために，4GB のメモリを必要とする．

### 2.2 多世代化方式による解決方法

多世代化方式では，上記(1)(2)の問題を解決するため，以下の方針で設計を行った．

(a)最大世代数を固定．（現在 124 世代）

(b)最大世代分のマッピングを一つのテーブルで管理し，差分領域を共有．

(c)テーブル全体は差分領域に格納し，参照・更新時に必要な部分をメモリに展開．

(b)により，コピーしたデータは，世代間で共有することで，(1)の CoW 処理が 1 回となる．また，(c)により，大量のメモリ消費が回避でき，(2)の問題を解消する．

## 3. 多世代化方式の詳細

### 3.1 ボリューム構成

多世代化方式は，スナップショットを採取したい P-Vol に差分格納用の差分ボリューム(D-Vol)を一つ組み合わせ，一つの D-Vol で多世代のスナップショットを実現する機能を提供する．また，P-Vol と D-Vol から構成される各スナップショットは，仮想ボリューム(V-Vol)を用いて参照する．V-Vol(スナップショット)は，LVM の論理ボリュームとしてアクセスできる（図 3.1）．

スナップショット管理テーブルは，P-Vol 全領域に対応するエントリを備える．各エントリは，

[1] (株)日立製作所 システム開発研究所 Systems Development Laboratory, Hitachi, Ltd.
[2] (株)日立製作所 中央研究所 Central Research Laboratory, Hitachi, Ltd.
[3] (株)日立製作所 ソフトウェア事業部 Software Division, Hitachi, Ltd.

CoW 要否，および，マッピングの格納領域を最大世代数分確保する．スナップショットを採取する場合には，全エントリに対して，当該世代のマッピングが P-Vol を指し，CoW 要否を要するよう設定する．

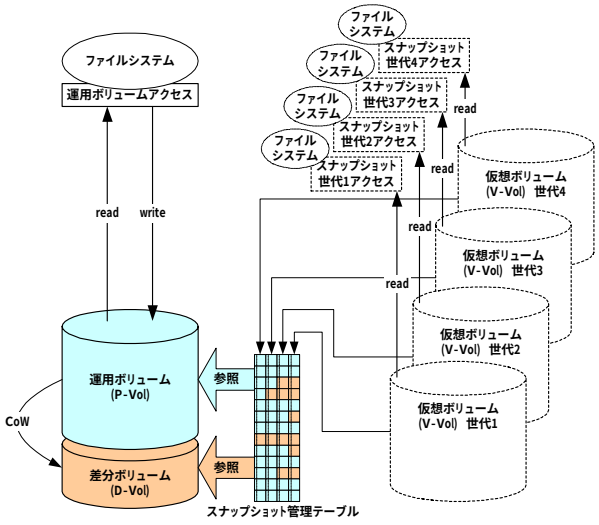


**図 3.1　各論理ボリュームの関係**

## 3.2　スナップショット維持

P-Vol への write 処理では，write 対象領域に対応するエントリの全世代の CoW 要否をチェックし，1 世代でも CoW 要の場合，CoW 処理を行う．CoW 処理では，write 対象領域の更新前データを P-Vol から D-Vol の空き領域にコピーすると同時に，CoW 要であった各世代のマッピング情報にコピーした差分ボリュームの領域のアドレスを代入するとともに，CoW 要否を否に変更する．

P-Vol に更新データを書き込む際の処理手順を図 3.2に示す．図中(1)～(3)は，以下の処理の実行順序を示している．

(1)更新前データを P-Vol から D-Vol にコピー

(2)テーブル更新(CoW 要否が要の世代全て)

(3)更新データを P-Vol に書き込み

多世代化方式では，スナップショットの世代数によらず，CoW 処理回数は，1 回である．

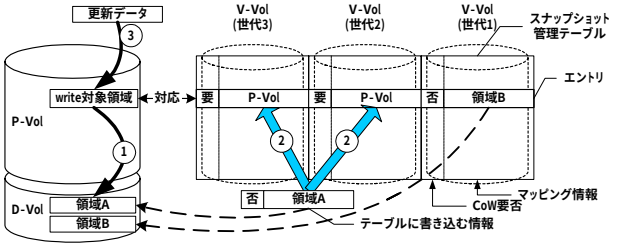


**図 3.2　P-Vol への write 処理時 CoW 処理**

## 3.3　スナップショット参照

V-Vol（スナップショット）の read 処理では，read 要求アドレスに対応するスナップショット管理テーブルのエントリから，当該世代のマッピング情報を参照する．マッピング情報には，当該領域が未更新であれば初期設定のとおり P-Vol を，更新済みであれば CoW 処理により D-Vol にコピーした領域を指す情報が格納されている．

図 3.3は，以下の順序でスナップショットが採取されたケースにおいて、世代 1～世代 3 の V-Vol から同じアドレス A,B のデータを読み出す場合に得られる結果を示している．

(1)世代 1 のスナップショット採取

(2)A-1 を A-2 に更新(A-1 を D-Vol に CoW)

(3)世代 2 のスナップショット採取

(4)B-1 を B-2 に更新(B-1 を D-Vol に CoW)

(5)世代 3 のスナップショット採取

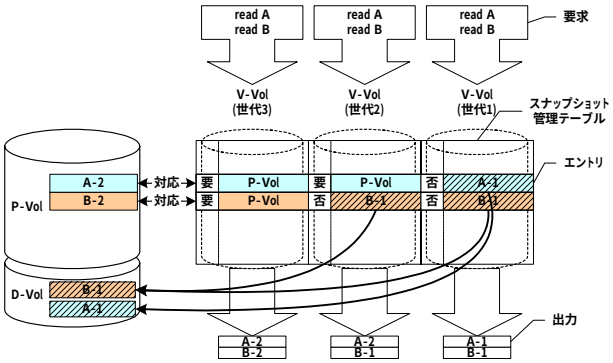


**図 3.3　V-Vol の read 処理時**

マッピング情報が指すデータを出力することにより，スナップショットが実現できる．

## 4. まとめ

多世代，大容量に対応する RAID システム内蔵型 NAS のスナップショット機能の設計を行った．

Linux 標準機能では，CoW 処理が世代数回必要となるケースがあること，および，差分管理を全てオンメモリで行い，大量の差分発生時にメモリ占有量が膨大となることから，多世代，および，大容量に不向きであることを示した．

最大世代数を固定し，全世代のマッピングを管理するテーブルを導入することで，多世代，大容量に対応したスナップショットが実現できることを示した．

---

☆ Linuxは，Linus Torvaldsの米国およびその他の国における登録商標または商標である．

☆☆ Logical Volume Manager
http://tldp.org/HOWTO/LVM-HOWTO/