B-011

RAIDシステム内蔵型NAS(4)
― 多世代スナップショット機能における最大論理ボリューム数拡大 ―
Embedded NAS for RAID System (4)
– Expanding the Number of Logical Volumes of Multiple Generation Snapshot Function –

清水 晃† 山崎 康雄†
Akira Shimizu Yasuo Yamasaki

1. まえがき

ファイルシステムのある時点におけるイメージを維持・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バックアップを生成する NAS の特長機能のひとつである。過去の複数の時点のイメージを保持でき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，高可用性を特徴とする大型 RAID の NAS 拡張である。スナップショット機能の多世代化により 1 つのファイルシステムに対して多くのスナップショットを取ることが可能になった。しかし、複数ファイルシステム運用時では LVM (Logical Volume Manager)の最大論理ボリューム数の制限により十分な数のスナップショットを作成することができない。そこで最大論理ボリューム数の拡大を行い、運用上問題ないか検討を行った。Linux LVM による試作を行い、最大論理ボリューム数拡大に伴う実課題の摘出と改善を行った。

2. 多世代スナップショット機能における課題

RAID システム内蔵型 NAS において、多世代スナップショット機能は LVM のレイヤで実現されている。多世代スナップショット機能は、ファイルシステムが実際にある運用ボリュームとスナップショット情報を保持する差分ボリュームで構成される。スナップショットを作成すると、それ以降の運用ボリュームへの更新があると、オリジナルデータを差分ボリュームへ退避してから、運用ボリュームへの更新を行う。作成されたスナップショットは運用ボリュームとは別の論理ボリュームとして見える。（図 1）

大型 RAID は数十 TB 以上と大容量であるため、RAID システム内蔵型 NAS では複数ファイルシステムで運用される。このため、複数のファイルシステムで多世代スナップショット機能を用いた運用すると、膨大な数の論理ボリュームが必要となる。この際に、NAS 全体で保持できるスナップショットの数が LVM の最大論理ボリューム数の制限を受けてしまう。

このため多世代スナップショット機能を実現するにあたり、最大論理ボリューム数が十分に足りているか確認

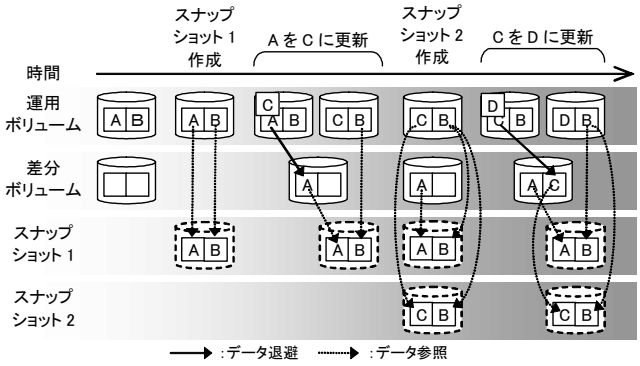

図 1：多世代スナップショット機能

する必要がある。もし足りないようであれば、要求仕様に合わせ最大論理ボリューム数を拡大する。実装に際しては、以下の観点に注意する必要がある。

(1) メモリ使用量
最大論理ボリューム数を拡大すると必要となるメモリ量が増加する。その際、計算機で処理可能なメモリ量に収まるかを確認する。メモリには(1-1)一時的に利用するメモリと、(1-2)恒久的に利用されるメモリがある。

(2) 応答時間
アルゴリズムにより、単純に最大論理ボリューム数を拡大すると応答時間が極端に増加する可能性がある。実運用に耐えうる応答時間に収まっているかを確認する。

3. 最大論理ボリューム数拡大の試作

Linux LVM 1.0.7 をベースにした多世代スナップショット機能における最大論理ボリューム数を拡大した。

128 個のファイルシステムを運用しているケースを想定する。スナップショットは 1 日 2 回作成し、1 ヶ月分（31 日分）のスナップショットを保持する運用を想定し、1 ファイルシステムあたり 62 個のスナップショットを取る。このような運用ケースでは、128 個のファイルシステムで 62 世代のスナップショットを保持するため、8192 個（=64×128）の論理ボリュームが必要となる。

ベースとなる Linux LVM では最大論理ボリューム数は 256 であった。256 個の論理ボリュームではファイルシステムあたり平均 2 個のスナップショットしか保持できない。このため、Linux LVM を用いた多世代スナップショット

† (株)日立製作所 中央研究所
Central Research Laboratory, Hitachi Ltd.

機能の試作では最大論理ボリューム数の拡大が必要になった。

最大論理ボリューム数を 8192 に拡大した際、メモリ使用量および応答時間の確認をおこなった。

3.1 一時的に利用するメモリ量の確認

一時的に利用するメモリ量として、LVM コマンド実行時のメモリ使用量を調査した。大規模構成の環境において各種 LVM コマンドが正常に実行できるかの確認を行った。

論理ボリューム数が 256 以下では出現しなかったが、大規模構成時になるとメモリ割当てに失敗する LVM コマンドが存在した。

ソースコードを調査した結果、この原因は単純なメモリリークであった。このため、このメモリリークを対策したことによりメモリ割当てが失敗するケースはなくなった。

3.2 恒久的に利用するメモリ量の確認

恒久的に利用するメモリ量として、カーネル内のメモリ使用量を調査した。カーネル内のメモリ使用量に関しては、机上計算による見積もりによって確認を行った。

見積もりの結果、多世代スナップショット機能を用いているファイルシステム（運用ボリューム 1TB、差分ボリューム 2TB）を 128 個運用している場合、1TB 以上のメモリが必要となることが判明した。

内訳を調べると、メモリ使用量の 30%ほどが LVM レイヤにおける IO 統計情報のために使われていることが判明した。この種の統計情報は OS でも採取されているため、LVM レイヤにおける IO 統計情報の採取を中止した。

これにより、カーネル内のメモリ使用量を 30%削減した。

3.3 応答時間

応答時間として、各 LVM コマンドの応答時間を調査した。多世代スナップショット機能を用いているファイルシステムを 128 個運用している環境で、各種コマンドの応答時間を測定した。

測定の結果、LVM 構成情報を再作成するコマンド vgscan が数分かかっており、運用上問題となると判断した。その他のコマンドに関しては、運用上問題となる程は増加していなかった。

vgscan の性能劣化に対する対策については次節で述べる。

4. vgscan の性能改善

性能劣化の原因となっている処理を見つけるため、ソースコードを詳細に調査した。その結果、論理ボリューム番号の重複チェック処理で効率の悪いチェックをしていることを突き止めた。

vgscan は使用可能なディスクから LVM 管理情報を再作成する。この際、登録する論理ボリュームの番号が既に使われていないか重複チェックを行っている。重複チェックはライブラリ関数として提供されており、そのチェックは以下のように行っていた。

(a) 使用中論理ボリューム番号リストを作成する。
(b) 未使用論理ボリューム番号リストを作成する。
(c) 論理ボリューム番号が未使用かチェックする。

vgscan では論理ボリュームごとに上記ライブラリ関数を用いていたため、効率の悪い重複チェック処理となっていた。

効率よく重複チェックが行えるように、vgscan における重複チェック処理を変更した。まず、(a)、(b)、(c)を別関数に分離し、新しいライブラリ関数を作成した。vgscan の重複チェック処理においては、コマンド実行中は使用中論理ボリューム番号リストを保持し、論理ボリュームを登録するたびに使用中論理ボリューム番号リストを更新するようにした。これにより、無駄に使用中論理ボリューム番号リストを作成しないよう変更した。

図 2に対策後の vgscan の処理時間を示す。本体策により処理時間を 98%削減することができ、運用上問題とならない応答時間にすることが出来た。

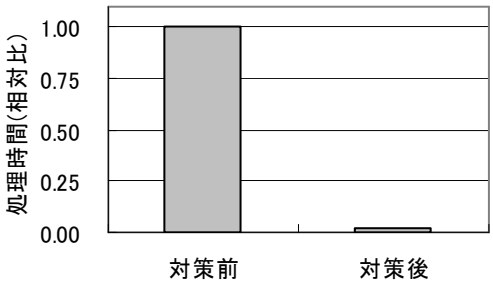

図 2：vgscan 処理時間の比較

5. おわりに

RAID システム内蔵型 NAS で多世代スナップショット機能を実現するにあたり、十分なスナップショットを保持できるよう最大論理ボリューム数の拡大が必要な場合がある。その際に運用上問題になりそうな、メモリ使用量および応答時間に関して調査を行った。

Linux LVM 1.0.7 をベースに試作を行い、最大論理ボリューム数の拡大を行った。メモリ使用量および応答時間調査の過程でさまざまな問題が発見されたが、適切に対処することで、最大論理ボリューム数を拡大しても運用上問題ないことを確認した。

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) － 多世代スナップショット機能－，情報処理学会第 66 回全国大会，5D-3(2004)