

C-019

大規模ストレージシステムにおける動的容量割り当て(Dynamic Provisioning)機能の研究開発

Dynamic Provisioning Function in large scale enterprise storage system

江口 賢哲†
Yoshiaki Eguchi

1. はじめに

企業情報処理システムのデータ量は年率 1.4~1.6 倍で増加している。それに伴い、企業情報処理システム内のストレージ装置やデータの管理は複雑化し、コストの増加を招く傾向にある。

一方で、ストレージ装置の導入では、増加データ量を予測して、容量不足にならないように、サーバ毎のボリューム容量を決定し、業務に影響が生じないようにするのが一般的である。さらに、ビジネス環境の急激な変化等によるデータ量の急増に備えも必要のため、一般的に SAN (Storage Area Network) 環境のストレージ装置の容量使用率は 30~40%といわれている。

以上から、ストレージ装置の容量使用率向上の実現は、情報処理システムの大きな課題の 1 つである。本課題に対し、業務アプリケーションからのデータの書き込みに応じて、ストレージシステムの実容量を動的に割り当て、容量使用効率を最大化する「動的容量割り当て (Dynamic Provisioning) 機能」が注目されている。

本研究では、従来ストレージ装置と比較して、シーケンシャルアクセス性能を維持し、かつストレージ装置の容量使用効率を高めた Dynamic Provisioning 機能の方式を検討した。

2. Dynamic Provisioning 機能とその課題

2.1 Dynamic Provisioning機能

本節では、Dynamic Provisioning 機能について説明する。

従来、エンタープライズクラスのストレージ装置は、信頼性や性能の観点から、複数の HDD で RAID グループを構成し、RAID グループから 1 つ以上の論理ボリュームを定義し、その論理ボリュームをサーバからアクセス可能にすることで、データの確実な格納と読み出しを実現している^{[1][2]}。従来の各論理ボリュームの記憶容量は、論理ボリューム間で共有することはできず、ある論理ボリュームはデータが満杯で実記憶領域が不足し別のボリュームはまだデータが格納されていない実記憶領域(未使用容量)が残っている場合でも、その未使用容量を有効に使用することはできない。

それに対し、Dynamic Provisioning は、図 1 に示すように、以下の機能を提供し、未使用容量を有効に使用することを可能とする。

- (1) 実記憶領域である論理ボリュームを論理的な容量プール化する機能
- (2) 容量プールをデータの格納先とする仮想ボリューム機能 (サーバは、仮想ボリュームに対して、データのリード/ライト処理を行なう)

- (3) 仮想ボリュームに対するデータのライト処理に対して、容量プールから動的に実記憶領域を割り当てる機能

また、実記憶領域の動的な割り当て処理は、特定の RAID グループへのアクセス集中と性能低下を起こさないために、複数のハードディスクドライブにデータを分散して格納し、ボリューム毎のアクセスの負荷の平準化を可能としている。

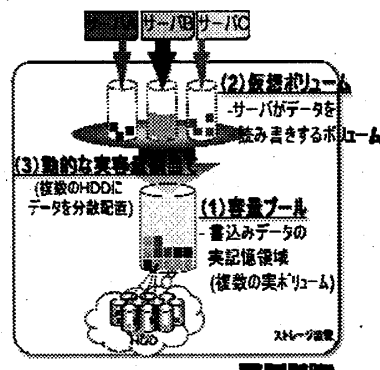


図 1: Dynamic Provisioning 機能の動作概要

2.2 課題と目的

Dynamic Provisioning 機能による、「動的な実容量割り当て」は、RAID 構成を基本とするストレージ装置で実現するためには、性能上、以下の 3 点の問題を抱える。

- (1) 仮想ボリュームへのシーケンシャルアクセスデータのランダム配置
- (2) 仮想ボリューム間のパリティ生成競合
- (3) 動的容量割り当て処理オーバヘッド

(1)は、RAID のパリティサイクル内のデータ領域のサイズと動的容量割り当てサイズの問題である。例えば、ライトデータサイズが大きく、書き込まれるデータのみで新しいパリティ生成が作成可能なシーケンシャルライト処理で、割り当てサイズとパリティサイクル内データのサイズが整数倍の関係になっていないと、データがパリティサイクルの途中で分断される。そのため、従来はシーケンシャルライト可能だったデータが、RAID5 ランダムライト処理となり、RAID5 のパリティ生成に必要な、ハードディスクドライブからの旧データ、旧パリティのリード処理が必要となり、性能上、従来の論理ボリュームに比べて、不利となる。

(2)は、動的な実容量割り当ての結果、複数の仮想ボリュームで、1つのパリティサイクルを共有する場合、あるパリティ生成処理を行う際に、複数の仮想ボリューム間でのパリティ生成競合が発生する問題である

(3)は容量割り当てサイズが小さいと、実容量割り当て処理にかかるオーバヘッドが、最悪の場合、データの書き込み毎に発生する問題である。

† (株) 日立製作所 システム開発研究所
Systems Development Lab., Hitachi, Ltd.

本検討の課題は、上記(1)~(3)の問題を解決して、RAID構成を基本とするストレージ装置で、容量使用効率の向上の実現と、従来の論理ボリュームと同等の性能を両立させる、動的容量割り当てサイズの決定である。

3. 動的容量割り当てサイズの決定

まず、サーバ OS のファイルシステムの作成時発生するメタデータのライトパターンから、動的容量割り当てサイズの見当をつける。一般的に、サーバ OS はフォーマットシステムを作成するために、データを管理するためのメタデータ領域を作成して、ファイルを格納できるようにする。この際、メタデータ領域にデータを書き込む。

動的容量サイズとメタデータの配置によっては、このメタデータ領域作成にかかわるライト処理で、仮想ボリューム全体に実容量が割り当てられるため、容量使用効率の向上にならない。そこで、動的容量割り当てサイズの最大値の目安として、主な OS のファイルシステム作成時のメタデータの書き込みがどのようになっているかを調査した。その結果を、表1にまとめる。

表1: 主なファイルシステム毎のファイルシステム作成時のメタデータライト

#	ホスト OS	ファイルシステム	ファイルシステム作成時のメタデータライト
1	Win2003 Srv	NTFS	先頭アドレス付近等4ヶ所
2	Linux	XFS	2GB 単位(A.G.単位)
3		Ext3	128MB 単位
4	Solaris	VxFS	先頭アドレス付近
5	AIX	JFS2	先頭アドレス付近
6	HP-UX	JFS	先頭アドレス付近

Linux の Ext3 の調査結果に注目し、このファイルシステムで、容量使用効率を従来の論理ボリュームの30%を実現できるサイズ

$$\frac{128}{3} [\text{MB}] - (A)$$

を動的容量割り当てサイズの最大値とする。

次に、2.2 節にあげた問題点(1)と(2)を解決について見当する。これは、動的容量割り当てサイズを、ストレージ装置の中に存在しうる、複数の RAID レベル、多様な RAID 構成ごとに異なるパリティサイクルの大きさの最小公倍数にすれば良い。よって、ストレージ装置のブロックサイズを Sb、ストレージ装置がサポートする RAID Level、構成を表2に示すとおりとすると、

表2: サポート RAID レベル、RAID 構成

#	RAID レベル	RAID 構成
A	RAID1	aDaD
B	RAID5	bD1P、cD1P
C	RAID6	dD2P

(動的容量割り当てサイズ)

$$= \text{lcm}(a, \text{lcm}(b, \text{lcm}(c, d))) \cdot Sb - (B)$$

更に、2.2 節の(3)の解決に関して、検討する。ここでは、動的な容量割り当て処理で生じる OVH を、従来の論理ボリュームに対して、平均で 1%程度の発生頻度にすることを目標とした。そこで、(B)で求めたサイズに対してα倍のサイズが Sb の 100 倍程度の割り当てサイズとなるように考える。

(動的容量割り当てサイズ)

$$= \text{lcm}(a, \text{lcm}(b, \text{lcm}(c, d))) \cdot Sb \cdot \alpha - (C)$$

最後に、(C)の結果が(A)の目安値の以下となる、動的容量割り当てサイズを(D)に基づいて、決定する

$$\frac{128}{3} > \text{lcm}(a, \text{lcm}(b, \text{lcm}(c, d))) \cdot Sb \cdot \alpha > 100 \cdot Sb - (D)$$

4. 検証

設計データ用に 33T バイト、部門毎共有データ用に 20T バイト、個人 PC データ用に 52T バイト、予備の未割り当て容量 36 T バイト、合計容量 141 T バイトを導入しているストレージ装置に Dynamic Provisioning 機能を適用する。

図2に示すように、従来の論理ボリュームでの運用では、実際のデータの量は全ストレージ容量の約 36%だが、サーバに割り当てられた容量は、別用途に使用することはできないため、新規にボリュームを追加する場合に使用できる容量は 36T バイトのみである。一方、Dynamic Provisioning 機能を適用することで、全ストレージ容量のうちまだ使用されていない約 64%の容量(90T バイト)を利用可能にした。

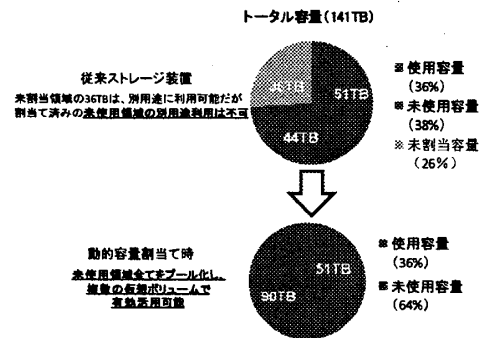


図2 動的容量割り当ての容量使用効率の効果

シーケンシャル性能は、従来の論理ボリュームと比べて、90%以上の性能が出せている。更に Oracle で 1[TB]の表領域を作成するまでの時間を計測した際の結果を表3に示す。

表3 Oracle で 1Tbyte の表領域作成時間

ストレージ	1 回目	2 回目	3 回目
従来論理ボリューム	1 時間 56 分 07 秒	1 時間 52 分 32 秒	1 時間 52 分 37 秒
動的容量割り当て (仮想)ボリューム	1 時間 54 分 20 秒	1 時間 53 分 25 秒	1 時間 53 分 19 秒

このケースでは、従来論理ボリュームと同等の時間で終了していた。従来論路ボリュームと同等の性能が出せている考える。

5. まとめ

RAID 構成をもつストレージ装置で、容量使用効率の向上と、従来論理ボリュームと同等のシーケンシャル性能を実現するための動的実容量割り当てサイズを決定し、その効果を検証した。

参考文献

[1] David A. Patterson, Garth A. Gibson, Randy H. Katz: Case for Redundant Arrays of Inexpensive Disks (RAID), SIGMOD Conference 1988: pp 109-116

[2] http://www.hds.com/assets/pdf/wp_195_exchange_best_practices.pdf