C-044

ミッドレンジストレージ向け動的容量割当て機能の研究
Dynamic Provisioning Function for Midrange Storage

大平 良徳† 内海 勝広‡ 岩満 幸治‡
Yoshinori Ohira Katsuhiro Uchiumi Koji Iwamitsu

1. はじめに

企業が保有するデータ量は，年率約 1.5 倍で爆発的に増加している。これにより，企業内システムの設計・管理が複雑化しコストが増大する傾向がある。複雑化の例として，ボリュームの容量設計が挙げられる。ストレージ導入時，システム設計者は，将来のデータ量を予測し，容量不足が発生しないよう十分な大きさのボリュームを設計しなければならない。しかしながら，ビジネス環境の急変によるデータ量急増などの理由で，正確な予測は難しく，一般的なボリュームの容量使用率は約 30～50%と言われている。ここで、容量使用率とは、用意された実容量に対する使用された容量の割合である。

このような背景から，予め用意する実容量を削減し，容量使用率を向上させる動的容量割当て機能が注目されている。本研究では特に，ミッドレンジストレージの機能の 1 つである Dynamic Load Balance Controller（DLBC）との連携を可能とした，ミッドレンジストレージ向け動的容量割当て機能について述べる。

2. 従来技術

2.1. 動的容量割当て機能

ストレージは，サーバが書込むデータを，HDD 等の記憶媒体に作成したボリュームに格納する。一般的なストレージは，ボリュームを RAID 技術によって保護し，データの信頼性を高めている。このボリュームは，システム設計者が指定したサイズ分の実容量を予め用意する必要があり，ボリューム間で実容量を共有できない。例えば，あるボリュームはデータが満杯で実容量が不足しており，別のボリュームは全くデータが格納されておらず実容量が余っているという状況が起こりうる。

これに対して動的容量割当て機能は，図 1 のようにデータ書込みに応じて動的に実容量を割当てるボリューム（仮想ボリューム）を作成できる。動的に割当てる実容量（ページ）は，実容量をプールした容量プールから取得できる。

従来と違い，仮想ボリュームはサイズ分の実容量を予め用意する必要がなく，複数仮想ボリューム間で，容量プールにプールした実容量を共有できるため，容量使用率を向上できる。[1]

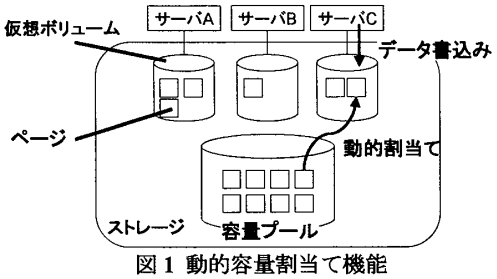

図 1 動的容量割当て機能

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

‡(株)日立製作所 RAIDシステム事業部
Disk Array Systems Division, Hitachi, Ltd.

2.2. Dynamic Load Balance Controller

一般的なミッドレンジストレージは，記憶媒体の制御などを行うコントローラ（CTL）を冗長化して信頼性を高めている。また，ミッドレンジストレージは，ボリュームごとに処理を担当する CTL を予め定めておき，1 つのボリュームに関する処理を 1 つの CTL で行っている。

このためシステム設計者は，ボリューム作成時，一方の CTL に処理負荷が集中して性能が低下しないよう，予め各ボリュームの処理負荷を見積もって処理担当 CTL を決定する必要があり，設計が難しいという問題がある。

これに対して DLBC は，一方の CTL の処理負荷が大きい場合に，この CTL が担当する幾つかのボリュームの処理を他方の CTL へ自動変更し，CTL 間の処理負荷を均等化できる。これにより，予め処理担当 CTL を決定する作業が不要となり，設計を簡単にできる。

3. 課題

本節では，動的容量割当て機能を DLBC と連携する場合の課題を示す。

従来技術では，容量プールを RAID 技術で保護する場合，容量プールを共有する全仮想ボリュームの処理担当 CTL が同じになるという制約が生じる。以下，理由を説明する。まず，RAID 技術では，記憶媒体の故障時でもデータを復旧できるよう，パリティと呼ぶ冗長コードを作成し，データと共に記憶媒体に格納している。ミッドレンジストレージは一般的に，パリティの作成単位であるパリティ列の処理を 1 つの CTL で行っている。ここで，図 2 に例示するように，容量プールにプールした各ページはパリティ列内に混在し得る。混在した各ページは，異なる仮想ボリュームに割当て可能であるが，1 つの CTL で処理されなければならない。一方で 2.2.節で述べたように，ボリュームごとに処理担当 CTL が定まっているため，仮想ボリュームに割当てた全ページは，1 つの CTL で処理されなければならない。結果，容量プールを共有する仮想ボリュームに割当てた全ページを 1 つの CTL で処理せざるを得ない。以上より，前述の制約が生じる。

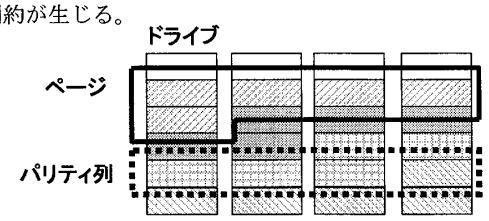

図 2 同一パリティ列内に複数ページが混在する例

この制約によって，ミッドレンジストレージは，一方の CTL の処理負荷が大きい場合に，従来のようなボリューム単位ではなく，容量プールを共有する全仮想ボリューム単位で処理担当 CTL を変更せざるを得ず，振り分ける処理負荷の粒度が粗くなり，DLBC の処理負荷分散効果が低減するという問題が生じる。特に，容量プールが 1 つしか存在しない場合，全ての処理負荷が片側の CTL に集中してしまう。

この問題を解決するには，システム設計者が容量プールを複数個作成し，振り分ける処理負荷の粒度を細かくしなければならない。しかし，容量プールを共有しない仮想ボリュームは実容量を共有できないため，容量プールが複数個存在すると，無駄が発生し，容量使用率が低くなってしまう。

以上より，動的容量割当て機能が DLBC と連携するには，容量プールを共有する各仮想ボリュームの処理担当 CTL を個別に変更可能にし，複数個の容量プールを作成せずに処理負荷を分散させることが課題となる。

4. 解決方式

本章では，前章で述べた課題の解決方式を提案する。解決方式は大きく 2 種類に分類できる。

(1) 複数 CTL に跨ったパリティ作成を可能にする方式
(2) 複数 CTL に跨ったパリティ作成が発生しないように，容量プール内の実容量を分割する方式

このうち(1)の方式は，CTL 間の排他処理によって大幅に性能が低下するため採用が難しい。そこで本研究では，(2)に分類される，チャンク分割方式と呼ぶ解決方式を提案する。

チャンクとは，複数ページをグループ化したものであり，チャンク内の全ページは 1 つの仮想ボリュームに割当てられる。チャンク分割方式は，動的容量割当て機能が，容量プール内の実容量をチャンクで分割し，仮想ボリュームにチャンク単位で実容量を割当てる方式である。

本方式では，図 3 に例示するように，1 つのパリティ列内にチャンクを混在させない分割方法をとる。本分割方法により，ページはパリティ列内に混在し得るが，混在したページは必ず 1 つの仮想 LU に割当てられる。これにより，複数 CTL に跨ったパリティ作成が発生しない。この結果 DLBC は，仮想ボリュームごとに処理担当 CTL を変更することができる。

以上より，動的容量割当て機能と DLBC とを連携させることができた。

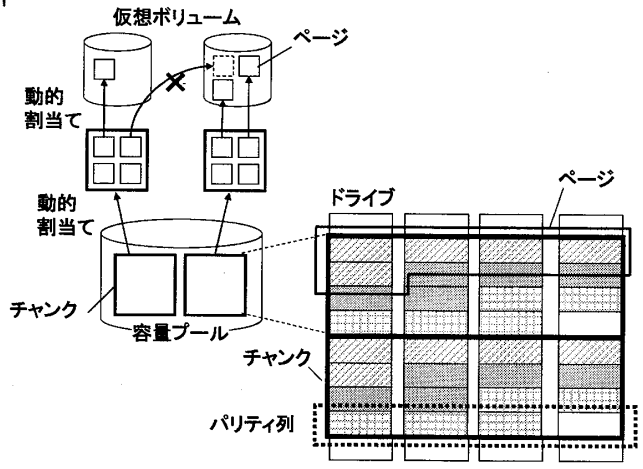

図 3　チャンク分割方式

次に，本方式の容量割当て方法が動的であることを示す。

本方式では，動的容量割当て機能が，仮想ボリュームに対するページの割当てを二段階で行う。一段階目では，本機能が容量プールからチャンクを取得し，仮想ボリュームに割当てる。二段階目では，本機能がデータ書込みに応じ，割当てられているチャンクからページを取得して仮想ボリュームに割当てる。この時，割当てられていないチャンクからページを取得しないようにし，チャンク内の全ページが 1 つの仮想ボリュームに割当たるようにする。以上の方法で，本方式でも動的に容量を割当てることができる。

ところで，(2)に分類される方式には他に，ページ分割方式が考えられる。ページ分割方式は，動的容量割当て機能が，容量プール内の実容量をページで分割し，仮想ボリュームにページ単位で実容量を割当てる方式である。本方式では，図 4 に例示すように，1 つのパリティ列内にページを混在させない分割方法をとっている。

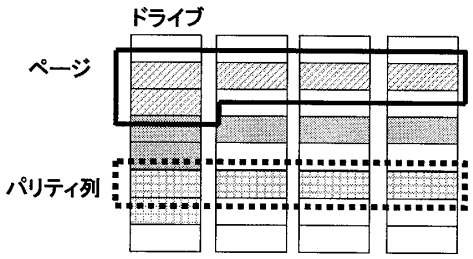

図 4 容量プールをページで分割する例

しかしながら本方式の場合，パリティ列内にページを混在させないために発生する余り領域がページごとに発生し，無駄が大きい。実際，余り領域の大きさは，チャンク分割方式では約 0.1%であるのに対し，ページ分割方式では約 2.3%と大きく，チャンク分割方式の方が優れている。

5. 評価

チャンク分割方式による動的容量割当て機能を使って，DLBC による処理負荷分散の効果をあることを検証した。

検証では，容量プールを共有する 4 個の仮想ボリュームの処理担当 CTL を全て CTL 1 に定め，全仮想ボリュームに対して等負荷となるような処理を実行した。結果を図 5 で示す。本結果より，処理負荷が高かった CTL1 の処理が CTL2 に切替えられており，CTL1 と CTL2 の負荷が均等されたことがわかる。以上より，DLBC の効果を確認できた。

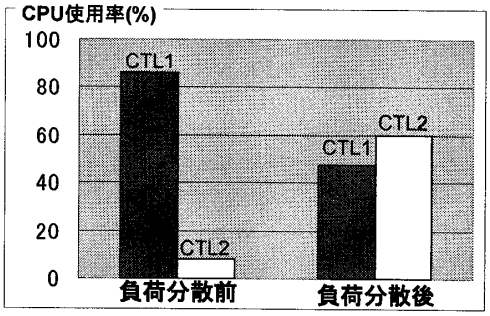

図 5 仮想ボリュームの処理負荷均等化

6. まとめ

本研究では，DLBC と連携を可能とするミッドレンジストレージ向け動的容量割当て機能を実現するため，チャンクと呼ぶ複数ページをグループ化した単位で容量プール内の実容量を分割し，仮想ボリュームにチャンク単位で実容量を割当てる方式を提案した。

また提案方式にて，処理負荷分散の効果を検証し，本方式が DLBC と連携可能であることを確認した。

参考文献

[1] 江口 賢哲，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，FIT 2008