Vol.2016-ARC-223 No.10
2016/11/30

データ圧縮機能搭載ストレージにおける空き容量を考慮したライト制御方式

松下貴記[†1]　川口智大[†1]　関俊哉[†2]

概要：格納データの圧縮は，高価なフラッシュストレージでビットコスト低減を目的として利用される．フラッシュストレージの高速性を活かすため，圧縮処理に伴う IO スループット性能低下を抑止する「圧縮オフロード」技術が有用である．圧縮オフロードは，ストレージコントローラに接続されたドライブ側で圧縮を行う技術であり，複数ドライブでの圧縮処理分散を可能とする．各ドライブは予想圧縮率を基に算出した物理的な記憶容量より大きな容量をストレージコントローラに提供する．そのため，圧縮率が想定よりも悪い場合には，物理的な記憶容量が枯渇しデータを格納できなくなる．枯渇を防止するため，ストレージコントローラはライト IO を受領した時点で，ドライブ未格納であるキャッシュ上のデータ量が格納先ドライブの物理的な空き容量以下であるかを判定する必要がある．しかし，ライト IO 契機にドライブやキャッシュの情報収集処理を実施すると，ライト応答時間が 2 倍に伸びる．この課題に対し，情報収集は周期的に実行することとし，次回の情報収集までに発生しうる予想ライトデータ量を加味して判定する方式を提案した．情報収集処理の非同期化により，ライト応答時間を非圧縮時と同等にできることを確認した．

1. はじめに

近年，高速なフラッシュデバイスが普及している．ストレージアレイにおいても，フラッシュデバイスのみで構成されたオールフラッシュアレイ（All Flash Array, 以降 AFA）が登場し，急速に市場が拡大している．2019 年には，AFA の売上高はブロックストレージ市場全体の 33.6%にまで拡大すると予測されている[1].

AFA の市場が拡大している要因には，データ量削減技術の登場により，安価に高性能なフラッシュ環境が利用可能になったことが挙げられる．データ量削減技術として代表的なものとして，データ圧縮が知られている．データ圧縮は，データのビット列の規則性を利用し，短い符号に置き換えることでデータ量を削減する技術である．ファイルシステムやデータベースなど，幅広いデータに対して削減効果がある．

一般にストレージで実現される圧縮機能では IO 処理を行うストレージコントローラが圧縮処理及び圧縮データを非圧縮状態に戻す伸張処理を実行する為，IO 性能が低下することが知られている．しかし，圧縮機能を AFA に適用する際，圧縮・伸張処理によって IO 性能が低下してしまっては高性能が求められる AFA には適用できない. 本問題を解決するには，処理負荷の大きい圧縮・伸張処理を，ストレージコントローラから AFA 内に複数搭載されるフラッシュデバイスにオフロードし圧縮・伸張処理の負荷分散を行うことで IO 性能の低下を抑止する方式（以降，圧縮オフロード）が有効である（図 1）[2][3].

圧縮オフロードでは，ストレージコントローラからフラッシュデバイスへのデータ書き込み時にフラッシュデバイス内で圧縮を行う．フラッシュデバイスは物理的な記憶容量より多くのデータを格納可能とするために，物理的な記憶容量より大きな記憶容量を仮想的にストレージコントローラに提供するが，圧縮率が想定よりも悪い場合には，物理的な空き容量が枯渇しデータを格納できなくなる．

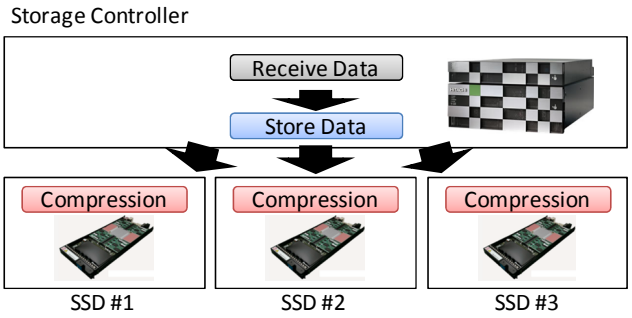

図 1　圧縮オフロード方式

本稿では圧縮オフロードにおいて，物理的な空き容量の枯渇を防止する方式の検討を行った．ストレージコントローラはライト IO に同期して，ドライブ未格納であるキャッシュ上のデータ量が格納先ドライブの物理的な空き容量以下であるかを判定する必要がある．しかし，この判定に必要なドライブやキャッシュの情報収集をライト IO 契機に実施すると，ライト応答時間が 2 倍に伸びる．この課題に対し，情報収集は周期的に実行することとし，次回の情報収集までに発生しうる予想ライトデータ量を加味して判定する方式を提案する．

2. 圧縮オフロード方式

本節では圧縮オフロード方式の詳細について説明する．

2.1 機能概要

圧縮オフロードの動作概要を図 2 を用いて説明する．

†1 (株)日立製作所　研究開発グループ
Hitachi Ltd., Research & Development Group
†2 (株)日立製作所　ICT 事業統括本部
Hitachi Ltd.,Information & Communication Technology Business Division

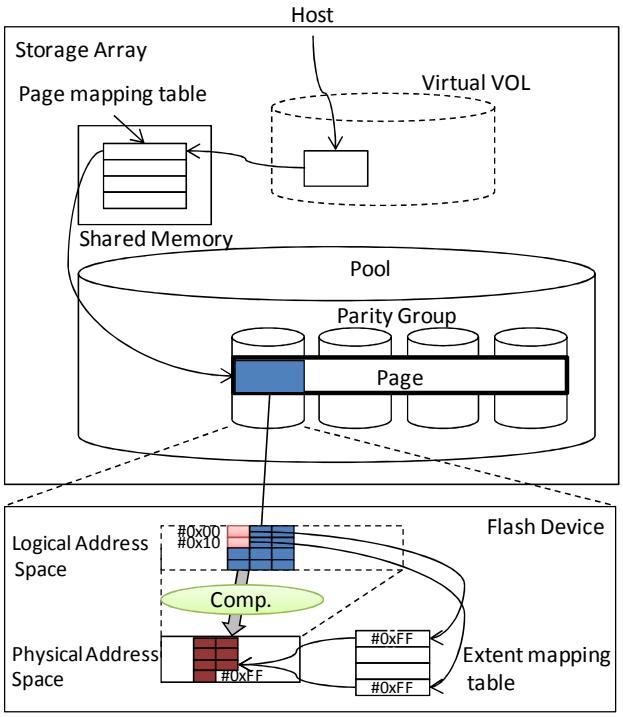

図 2　圧縮オフロード動作概要

圧縮オフロードは容量仮想化機能（一般に Thin Provisioning 機能とも呼ばれる）の構成で利用する．容量仮想化機能とはホストサーバに対して仮想的な容量のボリューム（以降仮想ボリューム）を提供し，ホストサーバから仮想ボリュームに対する書き込みに応じて，ストレージ装置内に有する物理的な容量プールから小領域単位（以降ページ）で割当てを管理する機能であり，ストレージ容量の使用効率を向上させる．容量仮想化機能は，仮想ボリュームのアドレスと，プール側アドレスの対応関係をページ単位のマッピングテーブルに保持し，ストレージアレイの共有メモリに格納する．

フラッシュデバイス内でも，ストレージコントローラ側の容量仮想化機能と同様に，論理と物理のマッピング管理を実施している．つまり，ストレージコントローラの Thin Provisioning 機能が仮想ボリュームに対して割当てたプール側アドレス（フラッシュデバイスがストレージコントローラに提供している領域の論理アドレス）と，フラッシュデバイスが内部で管理するフラッシュメモリの物理アドレスの対応関係を小領域単位（以降エクステント）のマッピングテーブルに保持し，フラッシュデバイス内メモリに格納する．

フラッシュデバイス内ではストレージコントローラから格納された論理データをエクステント単位で圧縮し，圧縮されたデータを格納するためのエクステント物理領域を確保し格納する．ここで，論理データをエクステント単位で圧縮してエクステントサイズ未満になった圧縮データを，物理領域エクステントに格納すると，物理領域エクステント内にはまだ空きがある．この空きは別の論理エクステントデータの圧縮データを格納する．

又，図示していないが，圧縮データへのアクセス時に，複数の論理エクステントデータが，同一物理エクステント領域に格納されている場合でも，マッピングテーブルに圧縮データの length 情報を保持しており，どの物理アドレスから読み出せば所望のデータにアクセスできるか判断できるテーブル構造となっている．

図 2 では，フラッシュデバイスの論理アドレス 0x00 と論理アドレス 0x10 は同じ物理アドレス 0xFF を参照していることを示している．このように，割当てられたエクステント単位の論理アドレスとフラッシュメモリの物理領域が 1 対多の関係で関連付けられる．このマッピングテーブルによって，ストレージコントローラからフラッシュデバイス論理領域全体にデータをライトされた場合でも，データ圧縮によって物理領域に格納されるデータサイズは小さくなる為，論理アドレスに関連付けられない物理アドレスが余ることになる．フラッシュデバイスでは，この余った物理領域に更にデータを書き込めるように，論理領域を拡張してストレージコントローラに提供する．論理領域は物理領域の n 倍の大きさとする．

2.2 ライト方式

図 3 に圧縮オフロードにおけるライト処理方式を示す．

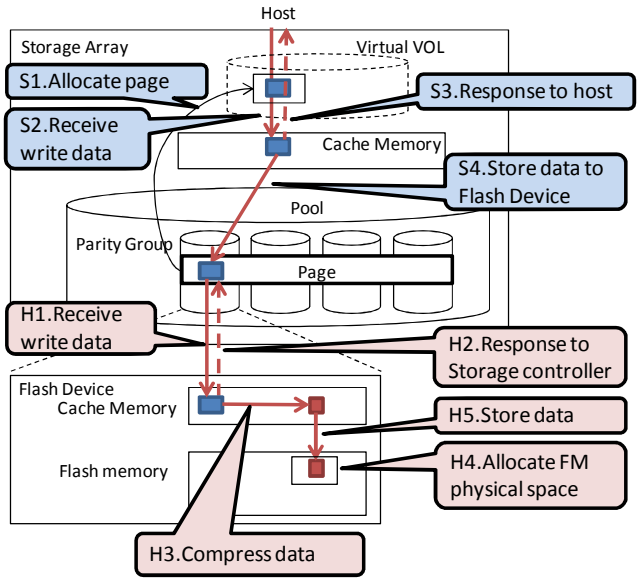

図 3　圧縮オフロードのライト方式

ホストサーバから仮想ボリュームの未書き込み領域へライトする場合，ストレージコントローラはプールの未割当領域をページ単位で仮想ボリュームのライト受領論理アドレスに割当てる（図 3 S1）．データ更新（既にページ割当済みの領域へのライト）の場合は，このページ割当処理は不要である．その後，キャッシュメモリの領域を確保し，ホストサーバからライトされたデータを確保したキャッシ

ュメモリに転送し（図 3 S2），ホストサーバへライト完了の応答を返却する（図 3 S3）．その後，プールへユーザデータを書き込む．プールはフラッシュデバイスの RAID グループにより構成されているため，データはフラッシュデバイスに格納されることになる（図 3 S4）．ライトデータを書き込む際，ページ単位のマッピングテーブルを参照し，書き込み先プールアドレスを特定する．このようにホストサーバへのライト完了応答後に，ドライブへの書き込みを実行する方式をライトアフタと呼ぶ．

書き込み要求を受けたフラッシュデバイスは，フラッシュデバイス内のキャッシュメモリ領域を確保後，ストレージコントローラからライトされたデータを確保したキャッシュメモリに転送し（図 3 H1），ストレージコントローラへライト完了の応答を返却する（図 3 H2）．その後，フラッシュデバイス内の圧縮エンジンを利用してデータを圧縮する（図 3 H3）．圧縮データを格納する前に，エクステント単位のマッピングテーブルを読み出し，未割当の物理アドレスを特定して物理領域を割当てる（図 3 H4）．最後に割当てた物理領域に対し圧縮データを格納する（図 3 H5）．

新規ライト（まだライトされたことが無い論理アドレスへのライト）の場合は，決定した物理アドレスとの関係をエクステントマッピングテーブルに保持する．データ更新（ライトされたことがあり，既に物理アドレスが関連づいた論理アドレスへのライト）の場合は，新しい物理アドレスを確保し，エクステント単位のマッピングを張り替える．更新前データが格納されていた物理アドレスはガベージとなり，リクラメーション（ガベージコレクション）論理によって，再度利用可能な状態となる．

3. 課題と目標

3.1 課題

前章に示した通り，フラッシュデバイスはデータ圧縮により増加した物理的な空き領域に更にデータを書き込めるように論理領域を拡張している．論理領域は物理領域の n 倍の大きさである．ここで，圧縮率が n 分の 1 よりも悪い場合には，論理領域にまだライトしていない領域が余っているにも関わらず，先に物理領域に空きがなくなってしまう．この状態を物理容量枯渇と呼ぶ（図 4）．

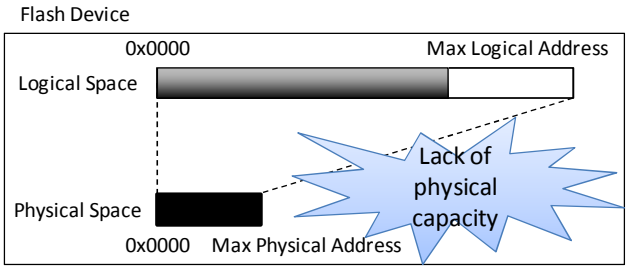

図 4　フラッシュデバイスの物理容量枯渇

物理容量枯渇状態になるとフラッシュデバイスはストレージコントローラから論理領域にデータをライトされても，割当てられる物理領域がない為，データが格納できない．この状態になるとフラッシュデバイスはストレージコントローラに対してライト不可を応答する．しかし，ストレージコントローラは前章に記載の通りライトアフタ方式であり，既にホストサーバからのライトデータをキャッシュに転送し，ライト完了の応答をホストサーバに返却している．したがって，フラッシュデバイスが物理容量枯渇状態で書き込めなかったデータはストレージアレイのキャッシュメモリに残り続けてしまうことになる．このドライブ未格納データによって，キャッシュの利用効率は低下する．それればかりか，キャッシュ容量がドライブ未格納データで満杯となり，ホストサーバから IO が出来なくなってしまう事態を招く可能性まである（図 5）．

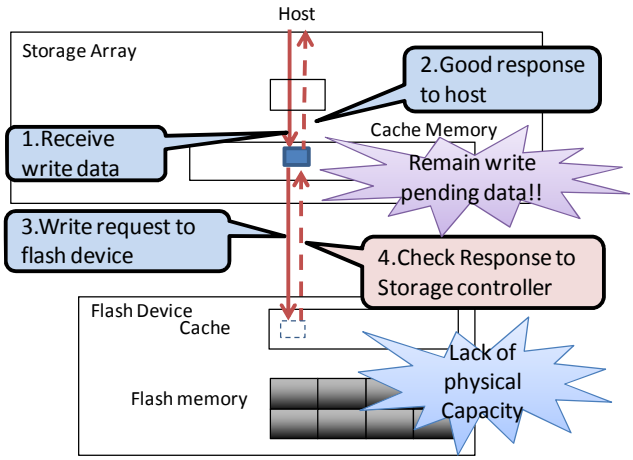

図 5　物理容量枯渇に伴うライト不可の問題

フラッシュデバイスが物理容量枯渇状態になることを防ぐ為には，ストレージコントローラがホストサーバからのライト IO を受領した時点で，キャッシュ上の未格納データ量（ライトペンディングデータ量，以降 WP 量）が格納先フラッシュデバイスの物理空き容量以下であるか否かを判定する必要がある．

この判定に必要な情報をホストサーバからのライト IO 契機に収集するとライト応答時間が約 2 倍に増大する．なぜなら，フラッシュデバイスの物理空き容量はフラッシュデバイスへのコマンド発行により取得する必要があり，当該コマンドの応答時間は Read コマンドの応答時間と同程度を要するためである．又，キャッシュの WP 量算出処理はストレージアレイの共有メモリへのアクセスが複数回必要であり，ライト応答時間に対して無視できない処理時間となる．

3.2 目標

本研究では，圧縮オフロードにおいてライト応答時間を圧縮オフロード非適用時と同等とし，且つフラッシュデバ

イスが物理容量枯渇状態となることを防ぐ物理容量枯渇防止方式を導出する事を目標とする．

4. 物理容量枯渇防止方式

4.1 方針

フラッシュデバイスの物理容量枯渇の課題を解決する為，ホストサーバからのライト契機で，当該ライトデータ及びキャッシュメモリのライトペンディングデータが全て，フラッシュデバイスへ書き込めるか否かを判定する（以降，枯渇判定）．枯渇判定の結果，格納可能と判定された場合は，ライトデータをキャッシュメモリへ転送し，ホストサーバへライト完了を応答する．格納不可と判定された場合は，ライトデータはキャッシュメモリへ転送せず，ホストサーバへライト不可を応答する．

枯渇判定のために必要な情報取得処理による，ライト応答時間の悪化を防ぐ為，情報取得処理は IO 非同期で周期的に収集する．ライト契機には非同期に算出された情報を参照して枯渇判定のみを行う．

一方，枯渇判定用の情報収集を周期的に収集する方式としたことにより，次の収集タイミングまでの間にライトされるデータ量を考慮する必要がある．つまり，情報収集したある時点（図 6 時刻 t1）の WP 量より，ライト契機の枯渇判定の時点（図 6 時刻 t4）における実際の WP 量は大きくなっている可能性がある．又は，情報収集したある時点（図 6 時刻 t1）の物理空き容量より，ライト契機の枯渇判定の時点（図 6 時刻 t4）での実際の空き容量は小さくなっている可能性がある．

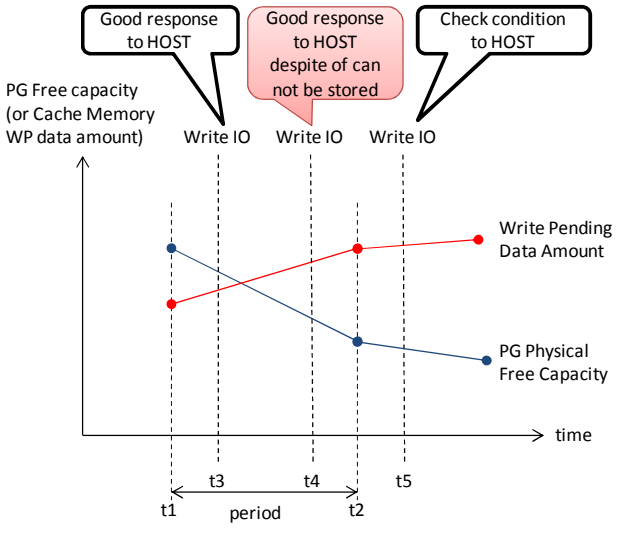

図 6　非同期情報収集による枯渇問題

したがって，情報収集してから次の情報収集タイミングまでの時間（図 6 における t2-t1 の時間）で予想されるライトデータ量（以降，枯渇判定マージンと呼ぶ）を加味し，以下の判定式で枯渇判定を実施する．

(*A*)当該*PG*に対する*WP*量＋(*B*)枯渇判定マージン
＜(*C*)当該*PG*の物理空き容量

上記判定式が真の場合（図 7 左），キャッシュ上に存在するライトペンディングデータやホストサーバからライトされたデータが全く圧縮されないデータであっても，書き込み先の RAID グループには確実にデータ格納できるだけの物理空き容量があると判断し，ライトデータをキャッシュに転送後，ホストサーバへライト成功の応答を返却する．一方，上記判定式が偽の場合（図 7 右），書き込み先の RAID グループの物理空き容量が無くなり，データを格納できない可能性がある為，ホストサーバへライト不可の応答を返却する．

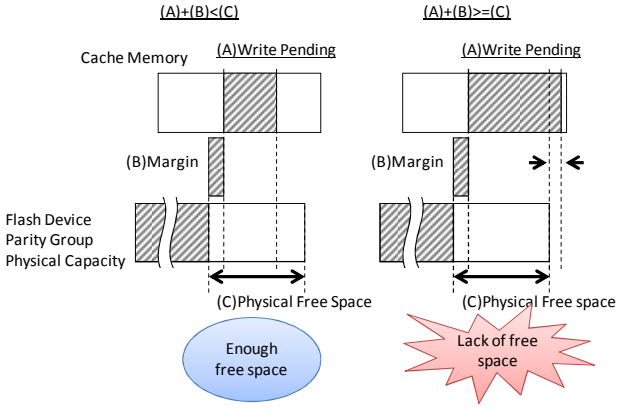

図 7　枯渇判定

次節では，物理容量枯渇防止方式の詳細（フラッシュデバイス物理空き容量の収集，キャッシュ WP 量収集，枯渇判定マージン量算出，枯渇判定）に関して述べる．

4.2 フラッシュデバイス物理空き容量収集

フラッシュデバイスの物理空き容量は，コマンドで取得することが出来る．図 8 に，コマンドに対してフラッシュデバイスが返却するデータの仕様を示す．

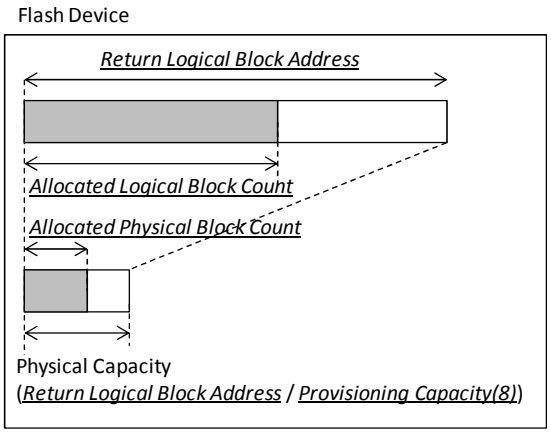

図 8　容量情報取得コマンド仕様

Return Logical Block Address にはフラッシュデバイスの論理容量（最終論理ブロックアドレス）が返却される．Logical Block Length in Bytes には，論理ブロックサイズが返却される．Provisioning Capacity はフラッシュデバイスの物理容量に対する論理容量の倍率が返却される．ストレージコントローラは Return Logical Block Address と Provisioning Capacity の除算によって，フラッシュデバイスの物理容量を認識することが出来る．Allocated Logical Block Count はフラッシュデバイスがストレージコントローラからのデータライトに対して割当てた論理ブロック数が返却される．Allocated Physical Block Count はフラッシュデバイスが圧縮後のデータに対して割当てた物理ブロック数が返却される．ストレージコントローラは Allocated Logical Block Count と Allocated Physical Block Count の除算によって圧縮率を認識することが出来る．又，物理容量と Allocated Physical Block Count の差によって，物理空き容量を算出することが出来る．

上記，フラッシュデバイス単体の物理空き容量から RAID グループ単位の物理空き容量を算出して枯渇判定に利用する．なぜなら，ストレージアレイは複数台のフラッシュデバイスで RAID グループを構成し，ストライプ列毎にパリティを計算して格納するため，ライトデータの格納先フラッシュデバイスの物理空き容量が十分でも，同一ストライプ列の他のフラッシュデバイスの物理空き容量が無ければ，ストライプ列単位ではデータやパリティを格納できない場合があるためである．

RAID グループ単位の物理空き容量は，同一 RAID グループ内で物理空き容量が最も少ないフラッシュデバイスの物理空き容量とデータドライブ数(RAID5 3D1P の場合は 3 台）の積で計算する（図 9）．RAID グループ内で最小の物理空き容量を基準にして算出することで，情報収集時点で当該 RAID グループに格納が保証されるデータ量を求めることが出来る．

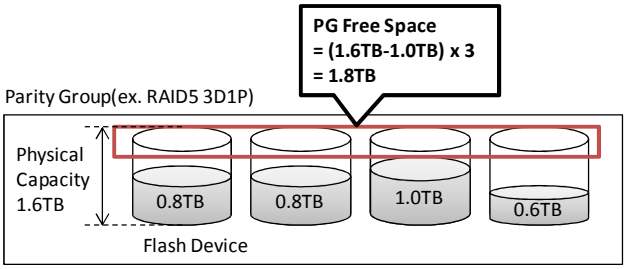

図 9　RAID グループ内物理空き容量計算方法

上記算出方式で，RAID グループ毎の物理空き容量を一定周期で算出・更新する．

4.3 キャッシュ WP 量収集

WP 量も RAID グループ毎に算出する．算出方法は，ドライブ毎に書き込み予定のライトペンディングデータを保持するキューの接続データ数の総和で算出する．RAID グループ毎の WP 量を一定周期で算出・更新する．

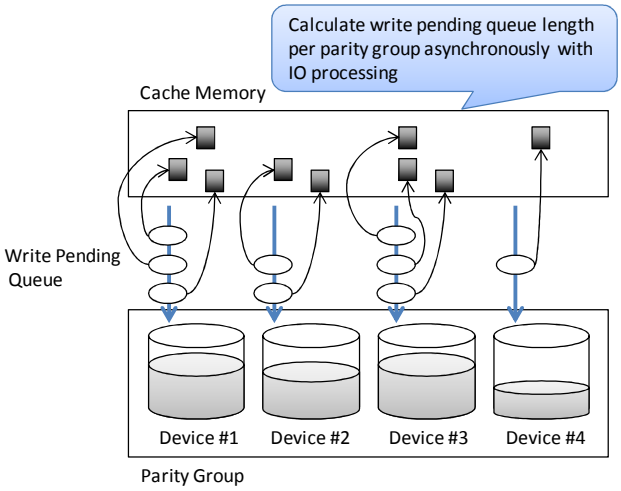

図 10　WP 量算出方法

4.4 枯渇判定マージン量算出

枯渇判定マージン量は，ストレージアレイが受けられる最大のライト速度（最大シーケンシャルライト性能）に基づいて決定する．

最大シーケンシャルライト性能を $W[Byte/s]$，枯渇判定用情報収集周期を $t[s]$ とすると，RAID グループ毎に $Wt[Byte]$ の枯渇判定マージン量があれば，ホストサーバからのライト要求に対して，確実に格納可能なデータ量をキャッシュに転送することが出来る．

4.5 枯渇判定

ホストサーバからのライト契機で，格納先がフラッシュデバイス RAID グループである場合，前述の判定式により枯渇判定を行う．

判定式が真であれば，ライトデータをキャッシュに転送し，ホストサーバにライト完了を応答する．判定式が偽であれば，ライトデータをキャッシュに転送せず，ホストサーバにライト不可を応答する．

5. 評価

本章では，前章までで説明した物理容量枯渇防止方式を適用した場合のライト IO 性能（ライト応答時間，ライトスループット）への影響を評価する．

5.1 応答時間

図 11 にランダムライトの応答時間を示す．

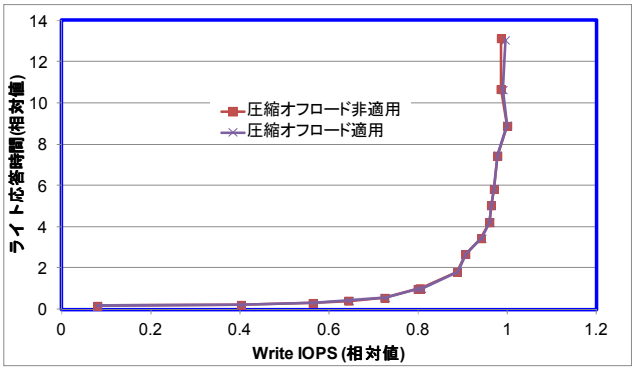

図 11　応答時間（ランダムライト）

圧縮オフロード非適用時と圧縮オフロード適用時で応答時間の増大は見られない. 圧縮オフロード非適用時(物理容量枯渇防止が不要)と同様に，最大ライトスループットの80%程度の負荷まではライト応答時間が悪化しないことが確認できる.

5.2 スループット性能

以下に圧縮オフロード適用時のスループット性能に関して，圧縮オフロード非適用時に対する性能比を示す.

表 1　スループット性能

IO 種別	圧縮オフロード性能比
Random Read	100%
Random Write	99.1%
Random Read/Write Mix	99.7%
Sequential Read	100%
Sequential Write	100%

圧縮オフロード非適用の場合と，圧縮オフロード適用の場合で，スループット性能の差は高々1%程度であることがわかる.

6. まとめと今後の課題

6.1 まとめ

本研究では，フラッシュの高性能とデータ圧縮による低価格化を両立する圧縮オフロードにおいて，フラッシュデバイス物理容量枯渇の課題に対する解決方式を検討した.

(1)　ホストサーバからのライト契機でフラッシュデバイス物理容量枯渇判定を行う．枯渇の可能性ありと判定された場合には，ホストサーバにライト不可を応答する方式を提案した.

(2)　当初は，物理容量枯渇判定に必要なキャッシュライトペンディングデータ量とデータ格納先ドライブの物理空き容量をホストサーバからのライト同期で取得する方式であったため，ライト応答時間が 2 倍程度増加する見積りとなった．枯渇判定用情報を周期的に収集し，次の収集までに予想されるライトデータ量を加味した枯渇判定方式を提案し，ライト応答時間への影響をゼロ化した.

(3)　提案方式のライト応答時間を測定し，圧縮オフロード非適用時からライト応答時間が悪化しないことを確認した.

6.2 今後の課題

枯渇判定に用いるドライブ物理空き容量及びライトペンディングデータ量はライト応答時間の増加を防ぐため，周期的に収集することとした．したがって枯渇判定では，次回の情報収集までの予想ライトデータ量（枯渇判定マージン量）を加味する必要があった.

一方，枯渇判定マージン量はストレージの容量効率を少なからず悪化させる．しかし，枯渇判定マージン量は，ストレージアレイの最大ライト性能で書き込まれても枯渇しないように算出された値であり，実際には常に最大ライト性能で書込みが継続するということは考えにくい．容量効率を改善する為，枯渇判定マージン量の適正化が検討する予定である.

参考文献

[1] Gartner, “Forecast Analysis: External Controller-Based Storage, Worldwide, 3Q15 Update”, G00275623, 28 January 2016
[2] 河村篤志，新井政弘．フラッシュメモリストレージにおける効率的圧縮データ格納･管理方式. 平成 28 年電気学会全国大会, 2016
[3] 河村篤志，小川純司．保証コードを考慮したエンタープライズフラッシュメモリストレージ向けの最適データ格納方式．平成 25 年電気学会電子・情報・システム部門大会，2013