ストレージシステムにおける 制御情報キャッシングによる性能向上

工藤 晋太郎† 下薗 紀夫‡

†‡株式会社日立製作所 システム開発研究所 〒244-0817 神奈川県横浜市戸塚区吉田町 292 番地
E-mail: †shintaro.kudo.gs@hitachi.com, ‡norio.shimozono.zf@hitachi.com

あらまし 複数のプロセッサにネットワーク接続された共有メモリに各種機能の制御情報を格納したストレージシステムにおいて，高性能化の為，制御情報を各プロセッサのローカルメモリにキャッシュすることが考えられる．制御情報への頻繁なアクセスを伴う各種機能の性能を保証する為には制御情報のキャッシュヒット率を保証する必要があり，それに必要なローカルメモリ容量を明らかにすることが課題である．本研究では，構成や機能の運用規模に応じた制御情報アクセス量，およびアクセス量とキャッシング用メモリ容量の関係についてモデルを立てて必要なローカルメモリ容量を見積もる手法を提案した．プロトタイプにて本手法を適用して見積もったローカルメモリ容量を与えた時機能の運用規模に依らず目標とするヒット率が得られることを確認した．

キーワード ストレージシステム，キャッシング，性能

Performance Boost of Storage system using Control Information Caching

Shintaro KUDO† Norio SHIMOZONO‡

†‡Systems Development Laboratory, Hitachi Ltd. 292 Yoshida-cho, Totsuka-ku, Yokohama-shi, Kanagawa, 224-0817 Japan

E-mail: †shintaro.kudo.gs@hitachi.com, ‡norio.shimozono.zf@hitachi.com

Abstract On the storage system that stores information that controls various functions in the shared memory attached by network with multi-processors, for higher performance, each processor caches control information into its local memory. In order to guarantee the performance of various functions with frequent access to control information, we must ensure the cache hit rate of the information, the challenge is to reveal the amount of local memory needed. In this study, we propose a method to estimate the capacity of the local memory required by making a model for the amount of information access according to configuration and the scale of management, and the relationship between the capacity of memory for caching and the amount of information access. We confirm the target hit rate regardless of the scale of management with estimated local memory capacity with this method.

1. 背景

ストレージシステムでは，多数の HDD を備え，大容量のデータを扱い，大量のボリュームをサポートする必要があると同時に，高性能，高信頼，高機能であることが求められる．

ストレージに求められる機能としては,例えばボリュームのミラーリング機能[1][2], スナップショット機能[3][4]等がある．

従来の大型ストレージでは,多数の組み込みプロセッサが共有メモリ(Shared Memory :SM)を介して分散処理することで高性能化していた．バッテリーバックアップされた共有メモリに制御情報（データの管理情報や各機能の管理情報）を格納しており，各種機能を実現するには，頻繁にこの制御情報にアクセスしながら処理する必要があった．

ところが，プロセッサの性能向上に伴い，ネットワークを介してアクセスする SM のレイテンシが相対的に大きくなってきた．その為，プロセッサが速くなっても，SM アクセスの多い機能の性能が向上しない問題があった．

2. 本研究の対象とするストレージシステム

2.1. アーキテクチャ

そこで,プロセッサ直結の高速かつ大容量の主記憶を持ったプロセッサを搭載して，そこに制御情報を格納してアクセスするというアプローチを採るストレージシステムが考えられる．図 1 に本研究の対象とするストレージシステムのアーキテクチャを示す.

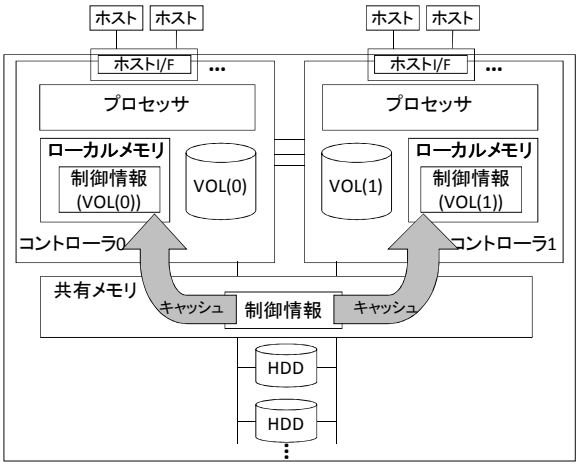

図 1 ストレージシステムアーキテクチャ

ただし機能の処理は一般に多種多様でプログラム規模が大きく，制御情報へ SM から LM への格納方法を個別に変更すると工数が大きい．また制御情報全体のサイズが大きく，全て格納する為のローカルメモリ容量を用意すると高コストになる.

そこで,機能共通のドライバ層で自動的にキャッシュする SM キャッシュ方式を採る(図 2). ドライバ層でキャッシュすることで機能の処理の変更を不要とし，また頻繁にアクセスされる情報のみをキャッシュすることで限られた容量で性能向上を実現できる．SM キャッシュ方式はドライバ層でキャッシュする為，ラインキャッシュ方式を採る.

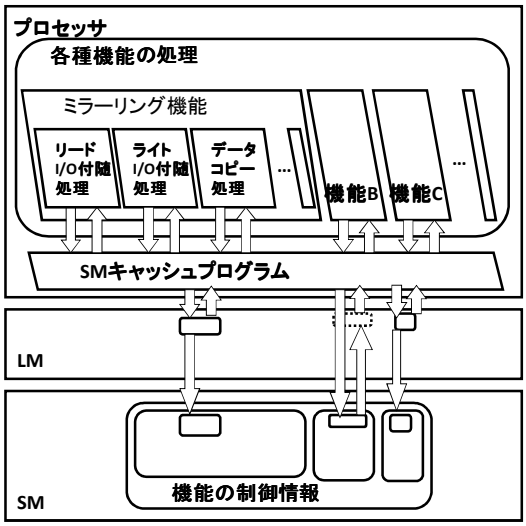

図 2 SM キャッシュ方式

2.2. 課題

ストレージにおいて性能は重要な設計要件であり,事前検証が必要である．各種機能の適用時の性能に関しては SM キャッシュ方式による SM アクセス OVH の削減効果の保証が重要である.

機能適用時の性能目標を達成する為の SM キャッシュヒット率をいかなる構成においても保証する為に，必要な SM キャッシュ用のローカルメモリ容量を明らかにすることが課題である.

3. アプローチ

機能適用時の処理は多種多様であり，構成，運用形態も様々な為，個別に検証するアプローチであらゆる状況での SM キャッシュ効果を網羅的に検証するのは現実的ではない．そこで，機能一般に共通のモデルを立てて見積もるアプローチを採る.

まず機能適用時の処理のうちで,主要処理に絞って高頻度にアクセスされる制御情報を解析し(3.1 節)，次に機能の運用規模に対してそれら高頻度にアクセスされる情報量がどう増えるかについてモデルを立てて推定し(3.2 節)，最後にヒット率と用意すべきローカルメモリ容量の関係を機能一般に共通のモデルを立てて推定した(3.3 節)．これにより，SM キャッシュ方式で目標ヒット率を達成する為のローカルメモリ容量を算出することができると考えた.

3.1. 各機能の主要処理の解析

SM キャッシュヒット率は，制御情報内でアクセス量に偏りが無いと仮定すれば，ヒット率=制御情報量全体に対して用意するローカルメモリ量の割合となるが,高いヒット率を保証するのに制御情報量の大半を格納可能なローカルメモリ容量を用意するのは現実的ではない.

そこで，各機能の処理の性質に着目した．一般に，機能適用時の処理には I/O 処理の付随処理やボリュームコピー処理といった通常運用中の処理と，機能適用に際しての初期設定や状態変更等の管理操作に対応する処理が存在し,両者でアクセスする制御情報が異なる．ここで前者の通常運用中の処理が実行時間の大半を占めるということに着目し，これら主要処理におけるアクセス対象をキャッシュ対象とすれば，十分な性能向上効果が得られると考えた.

よってこれら主要処理に限定してキャッシュヒットを期待するべき情報量(キャッシュ対象制御情報量と呼ぶ)を絞り込む.

3.2. 機能毎の制御情報アクセス量のモデル化

ユーザによって機能の運用規模（ボリューム数等）の条件は様々である．そうした条件によって，機能の主要処理のアクセスするキャッシュ対象制御情報量は変化する．

この変化の仕方について,制御情報の特性に次の三つに分類できることに着目して，モデル化するアプローチを採った．

第一は,機能適用規模に関係なくアクセス量が固定的な情報である．例えば制御情報自体の有効・無効を管理する情報等である．これについては主要処理のアクセスする情報全体をキャッシュ対象情報として扱う．

第二に,ストレージ装置内の制御対象資源に関連づいている制御情報である（資源の量はハードウェア構成等に依存する為，構成依存情報と呼ぶ）．資源には，物理的な資源（ディスク装置，転送バッファ等）や論理的な資源（論理ボリューム等）がある．この種類の制御情報へのアクセス量は，その制御情報の関連する資源の利用数に応じて増える．その為，それぞれの機能がその制御対象資源を扱う最大規模に応じて，その制御情報についてキャッシュ対象情報量の最大量を見積もることができる．

第三に，I/O 範囲の大きさに応じて，機能適用時の処理のアクセス量が増える制御情報である（ユーザデータの容量に対応して存在する情報であり，容量依存情報と呼ぶ）．この種類の制御情報は，その総量が膨大であり，全体をキャッシュ対象情報とすることは現実的ではない．こうした情報については，実用性能上性能向上効果が大きい I/O 範囲に基づくべきと考えた．例えば，ストレージのサポートするボリューム空間全体ではなく，ユーザデータキャッシュ用容量に相当する I/O 範囲までを,キャッシュ対象情報量の最大量とし，ユーザデータキャッシュにヒットする I/O パターンで高いヒット率を保証する方針が考えられる.なぜなら，ユーザデータキャッシュにヒットしない場合，I/O 処理においてディスクアクセスの為の処理 OVH が増す為,SM アクセス OVH が相対的に小さくなる為である．

以上の考え方で,機能の主要処理がアクセスする制御情報について分類し，それぞれキャッシュ対象情報量の最大量を見積もる．

例えば,機能 A を適用したボリュームへの I/O 処理において,ボリュームに関連した制御情報(機能 A ボリューム制御情報と呼ぶ)への I/O 処理でのアクセス量を合計 $AVOL_{io}$ [Byte]とする．ユーザの機能 A の適用ボリューム数が n 個である場合，機能 A ボリューム制御情報への I/O に伴うアクセス量は合計で $AVOL_{io}\times n$[Byte]である．

機能 A がサポートするボリューム数が最大 $AVOL_{max}$ 個であるとすると，機能 A ボリューム制御情報についてキャッシュヒットを期待すべき情報量は $AVOL_{io}\times AVOL_{max}$[Byte]と計算できる．

他の制御対象資源に関しても同様であり,それぞれ上記のように計算した値の合計値で，機能 A に関するキャッシュ対象情報量を計算できる．各機能について計算した値の合計値が，全体としてのキャッシュ対象情報量である(図 3).

機能の集合$\{P_1,P_2,...P_n\}$
制御情報種別$\{C_1,C_2,...C_m\}$
機能A主要処理での制御情報Xアクセス量:AX_{io}
機能Aの扱うX最大量$AX_{\max}$

$$\text{キャッシュ対象情報量}=\sum_{x=1}^{n}\sum_{y=1}^{m}P_xC_{y_{io}}\times P_xC_{y_{\max}}$$

図 3 キャッシュ対象情報量計算式

3.3. 容量依存情報による構成依存情報の追い出し防止

容量依存情報はユーザデータの一定容量ごとに存在する情報等であり，構成依存情報に比較すると総量が大きく,情報量あたりのアクセス頻度が低い.その為，構成依存情報が容量依存情報へのアクセス量増大によって SM キャッシュ上から追い出されるのを防ぐ為，両者の SM キャッシュ用ローカルメモリ容量をパーティションして，I/O 範囲が大きくなっても構成依存情報のヒット率が下がらないようにした．

3.4. キャッシュ対象情報量とヒット率の関係のモデル化

次に，見積もったキャッシュ対象情報量に対して，どの程度のローカルメモリ容量を与えれば目標とするヒット率が得られるのか，機能共通のモデルを立てて推定する．

ラインキャッシュの一般的なデータ構造としてセットアソシアティブでは，各キャッシュエントリに格納できるライン数(way 数)の上限に達することによるミス(競合ミス)が発生する．この競合ミスの発生率は，way 数と，制御情報アクセスの偏りや順序等のアクセスパターンに依存する．

ただしアクセスパターンに関しては,全ての処理についてアクセスパターンを網羅的に調べるのは現実的でない為，一般的なアクセスパターンとしてランダムアクセスを仮定したモデルを立てて見積もるアプローチを採った．

各 way 数について，セットアソシアティブ方式においてランダムアクセスでの競合ミスの発生確率を計算すると，キャッシュに載せる情報量とメモリ容量の関係は図 4 のようになる．例えば，4-way セットアソシアティブ方式で，目標とする SM キャッシュヒット率

を 95%なら載せる制御情報の 2 倍，80%なら 1 倍の容量が必要であることがわかる．

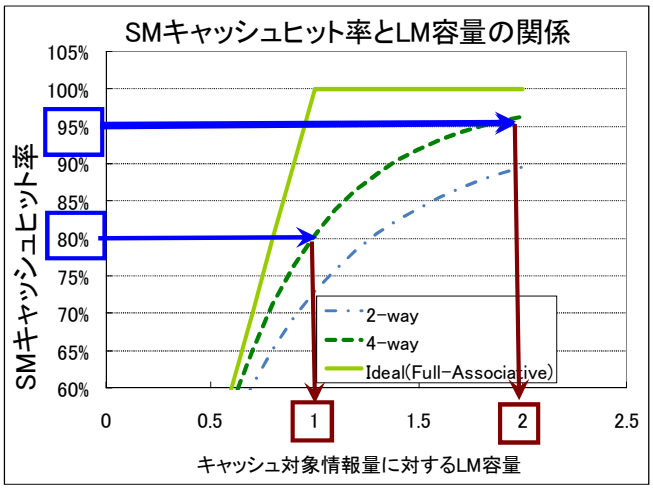

図 4 LM 容量に応じたヒット率のモデル

4. 評価

4.1. 評価条件

2.1 節に述べたような，高速なローカルメモリ直結のプロセッサと，それらが接続された共有メモリを持ち，ボリュームのミラーリング，スナップショット等の機能を提供していて，各種機能の制御情報が共有メモリで管理されているようなプロトタイプストレージシステムを用いて評価を行った．

このプロトタイプにおいて，SM キャッシュ方式を適用した時の，各機能運用時の SM キャッシュヒット率が本研究の手法によって保証できるかを評価する．

本研究の手法を用いるには，まず目標とするヒット率を決める必要がある．本評価においては，目標ヒット率を 95%以上と設定した．一般には目標ヒット率は，目標とする性能向上率と，処理時間に占める SM アクセス割合 SM と LM のレイテンシから決まる．

機能毎に 3.1-3.3 節のアプローチにより推定したキャッシュ対象情報量に対して，3.4 節で述べたモデルに基づく考え方を適用し，95%を達成する為のローカルメモリ容量としてその 2 倍を SM キャッシュ用に確保すべきローカルメモリ容量として推定した．

まずシミュレーションにより各機能の主要処理に関して目標としたヒット率 95%以上が見積もった容量を与えた時に得られるかどうかを検証し，次に実機にて実際に各機能を適用したボリュームを最大規模まで構成し，適用規模によらず目標 SM キャッシュヒット率 95%が得られるかを検証し，各機能が目標性能を達成することを確認する．

4.2. シミュレーションによるヒット率評価

シミュレータの入力は，ローカルメモリ容量，およびプロトタイプシステムでの各機能処理における SM アクセスのトレースログである．トレースログで SM アクセスパターンを再現し SM キャッシュ方式の動作を模擬した時のヒット率を出力する．

シミュレータでは与えるローカルメモリ容量を変えることができる為，本研究の手法を適用するにあたり，機能の適用規模の最大量を SM アクセストレースログの採取時の機能適用規模で置き換えて見積もった．こうすることで，ヒット率 95%以上を達成する為の見積もり値が十分であるか，そして過剰でないかを検証した．

機能の例として，ミラーリング機能運用時の I/O 3 パターンでのシミュレーション結果を図 5 に示す．

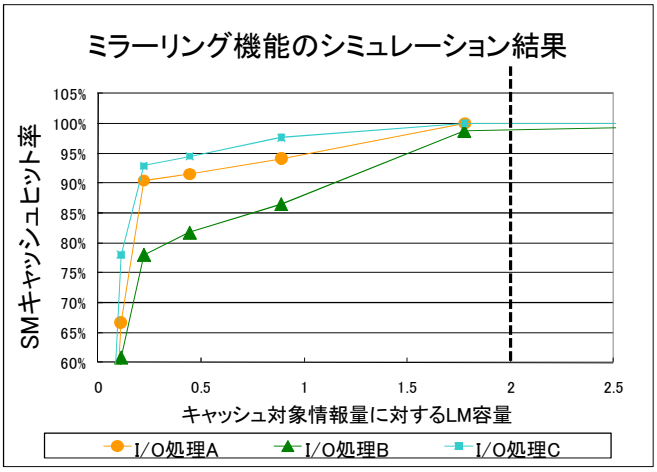

図 5 ミラーリング機能のシミュレーション結果

キャッシュ対象情報量に対して 2 倍を与えた時に全 I/O 処理でヒット率が 95%以上となった．I/O パターンによってはより小さい容量でヒット率 95%を達成するのはアクセス先の局所性によると考えられるが，全ての I/O で性能を保証するには，主要処理のアクセスをランダムと仮定した本手法のモデル化が妥当であると考える．

4.3. プロトタイプ測定によるヒット率評価

プロトタイプシステムにおいて実際に SM キャッシュ方式を適用し，本手法に基づいて見積もった容量をもとにローカルメモリ上に SM キャッシュ要の容量を確保して，機能を適用したボリュームに対し I/O を発行し，SM キャッシュヒット率の測定を行った．実際に機能を適用した状況でのヒット率が，機能のサポートする最大規模まで目標ヒット率 95%以上となることを検証する．

プロトタイプ測定(1)：適用規模に応じたヒット率評価

ミラーリング機能での測定結果を示す(図 6)．プロトタイプシステムにおいてミラーリング機能を適用したボリュームを多数構成し，通常運用を想定した検証項目として，リード I/O，ライト I/O の発行時とボリュームのコピーを行った時の SM キャッシュヒット率を測定した．ミラーリング機能適用規模を変えて測定した．

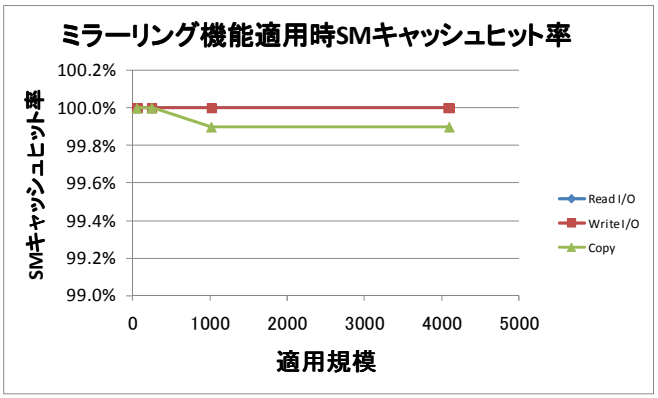

図 6 ミラーリング機能適用規模に応じたヒット率

結果，最大規模まで増やした場合でもヒット率は目標の 95%を上回った．

プロトタイプ測定(2)：容量依存情報のヒット率評価

スナップショット機能を適用して作成したボリュームのスナップショットイメージに対するリードアクセスを行った時の SM キャッシュヒット率を測定した．この I/O パターンでは，容量依存情報へのアクセスを伴い，容量依存情報についてヒット率を保証する最大規模の I/O 範囲にアクセスをかけた時のヒット率の確認と，構成依存情報と格納領域をパーティショニングした効果（構成依存情報のヒット率が落ちない）の確認が目的である．

ボリュームへの I/O 範囲を増やして測定した結果を図 7 に示す．実用性能上ヒット率保証が必要としたアクセス範囲においては目標ヒット率 95%を下回らないことを確認した．

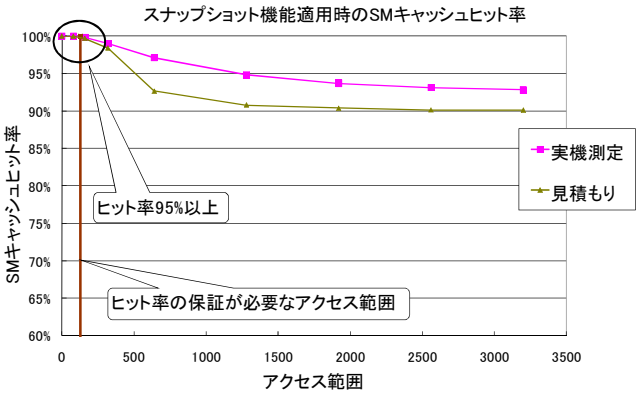

図 7 スナップショット機能適用時のヒット率

また，それ以上の規模の I/O 範囲については，ヒット率が下がっているものの，90%程度のヒット率より下回らなかった．この I/O 処理における SM アクセス内容を分析すると，構成依存情報へのアクセスが 9 割，容量依存情報へのアクセスが 1 割であり，容量依存情報へのアクセスについては SM キャッシュヒット率が低下しているが，ボリューム制御情報等の構成依存情報はヒット率が低下しないことが確認できた．

また，実測結果ではモデルに比べてヒット率の下がり方が緩やかであるが，この時アクセスされる容量依存情報がツリー構造を成しており，上位テーブルのアクセス量の増加が下位テーブルのアクセス量より小さいという傾斜が存在した．本手法の適用にあたり，この傾斜を考慮しなかった為，上位テーブルを下位テーブル同等に増加すると見積もっていた為と考える．

5. 結論

本研究では，ローカルメモリに制御情報をキャッシングしてアクセスするようなストレージを考えた時に，制御情報のキャッシングのヒット率を保証する手法を提案した．

提案手法では，主要な処理のアクセスする制御情報をその性質で分類，運用規模に応じたアクセス量の変化をモデル化し，キャッシュヒットの対象とするべき情報の最大量を明らかにした．そしてキャッシュ対象情報量とローカルメモリ容量，およびヒット率の関係をモデル化した．

これによりストレージの運用状況によらず常に制御情報のキャッシング効果を保証する為のローカルメモリ容量を見積もることができる．

プロトタイプストレージシステムでのシミュレーション及び実機測定にて検証し，本手法によって見積ったローカルメモリ容量で，目標とした SM キャッシュヒット率が得られることが確認できた．

参 考 文 献

[1] http://www.hds.com/products/storage-software/shadowimage-in-system-replication.html

[2] http://www.emc.com/products/detail/software/timefinder.htm

[3] http://www.hds.com/products/storage-software/copy-on-write-snapshot.html

[4] http://www.redbooks.ibm.com/redpapers/pdfs/redp4368.pdf