

ストレージ階層仮想化機能におけるアクセス頻度の低 OVH な採取方法 Low Overhead Monitoring Method for Virtual Storage Tiering Functions

今崎 美保[†] 坏 弘明[†] 大平 良徳[†] 須藤 梓[†] 江口 賢哲[†]
Miho Imazaki Hiroaki Akutsu Yoshinori Ohira Azusa Sudo Yoshiaki Eguchi

1. はじめに

年々増加するデジタルデータ量に対し、IT への投資額は増えていない。このため、ペタバイト(PB)クラスの大容量ストレージを運用する企業は、IT システムの性能要件とコスト要件の両立を実現するために、アクセス頻度に応じて適切なストレージ階層(高速なメディア(記憶媒体)や安価で低速なメディア)にデータを格納し、データ保持コストを削減している。しかし、年々増え続け大量になったデータをアクセス頻度に応じたストレージ階層に格納することが困難になってきているため、解決手段として、ストレージ領域(ページ)毎のアクセス頻度の偏りに着目した、ストレージでのデータアクセス頻度の監視と適切なデータ格納先階層判定および格納を行う、ストレージ階層仮想化機能がある[1]。

本研究では、大量のアクセス頻度情報をストレージ内で I/O 処理性能に影響を与えずに、低 OVH で採取する方法に関する検討内容について述べる。

2. 背景と目的

ストレージ階層仮想化機能を運用する場合、ページ毎に適切なディスク階層を判定するため、ページのアクセス頻度をモニタリングする。モニタ情報はページ毎にカウンタを用意しメモリに格納する。モニタ処理では、ページに I/O が発生する度、該当ページのカウンタをアップし、モニタ情報を更新する。ストレージ階層仮想化機能のメリットの1つは I/O 処理時間の短縮であるが、モニタ処理がストレージ装置の I/O 処理に影響し、全体で見ると I/O 処理時間の短縮を実現できない可能性がある。

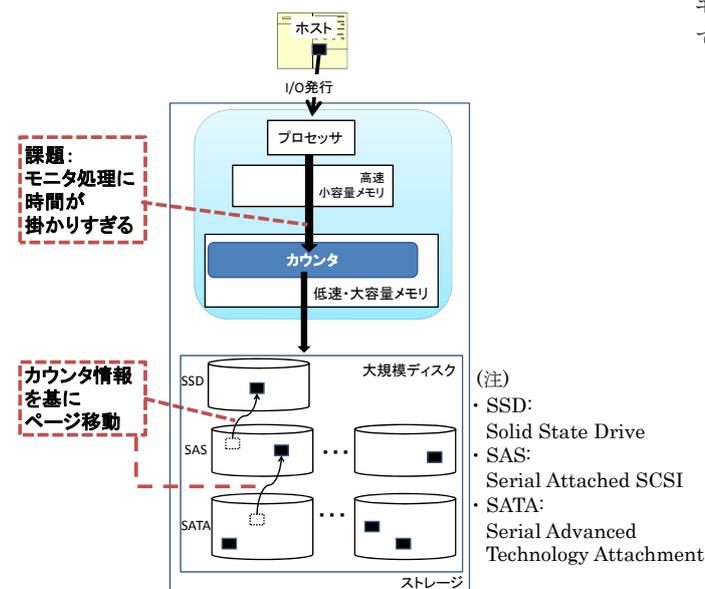


図1 大規模ストレージにおけるモニタ情報(改善前)

本研究では、ページアクセス頻度をストレージ装置が低 OVH でモニタ処理する技術の確立を目的とした。

3. 大規模ストレージ・モニタ処理の課題と目標

図1は、ストレージ階層仮想化機能を大規模ストレージに適応した場合である。モニタ処理の低 OVH 実現のため、プロセッサに近い高速なメモリにモニタ情報を格納したい。しかし、大規模ストレージでは機能対象のページ数が大量であり、モニタ情報が膨大な容量となるため、小容量の高速なメモリに全てを格納出来ない。従って、大容量メモリにモニタ情報を格納する必要があるが、大容量メモリはプロセッサの近くに配置することができないため低速処理となる。よって、モニタ処理が I/O 処理へ影響を与えてしまうという課題があった。I/O 処理への影響をなくすためにはモニタ処理時間を 98%削減する必要があったため、これを目標とした。

4. 低 OVH なアクセス頻度の採取方法

3章で述べた通り、モニタ処理時間は使用メモリの種別による。よって、モニタ情報格納を工夫した方式を提案し、図2に示す。まず、高速・小容量メモリと、低速・大容量メモリの2種類を使用する。後者は大容量であるため大規模ストレージ装置のディスクの全モニタ情報を格納可能であるが、前者は小容量であるため一部のモニタ情報のみ格納出来る。そこで、高速・小容量メモリは I/O 発生時リアルタイムでモニタ処理を実施する一方、低速・大容量メモリには高速・小容量メモリのモニタ情報を非同期で反映する。つまり I/O 発生時には、通常、高速・小容量メモリのモニタ情報のみアクセスするため、モニタ処理時間を削減できる方法を考案する。

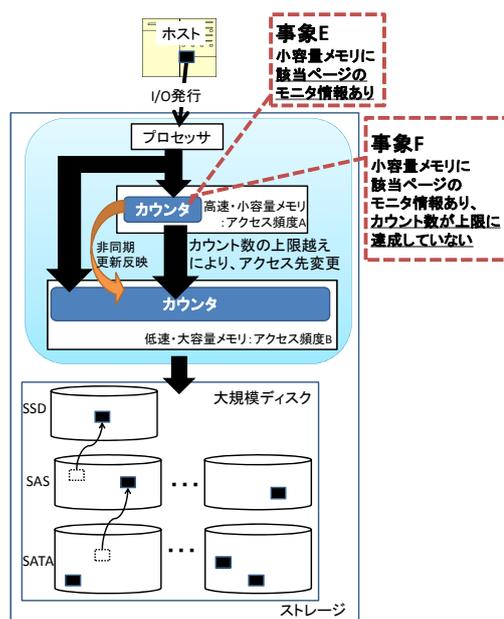


図2 低 OVH なアクセス頻度の採取方法(改善後)

[†](株)日立製作所 横浜研究所
Hitachi, LTD., Yokohama Research Laboratory

4.1 条件定義

本方法の OVH 量を数式で示すため、大規模ストレージ装置における I/O は高負荷な状況であるランダム系とし、簡易化のため 1 回の I/O 発行時の振る舞いを前提とする。表 1 に使用する変数をまとめる。

表 1 変数定義

| # | 項目 | 変数定義 | |
|----|---------------|---------------|-------|
| 1 | アクセス頻度 | A | |
| 2 | 高速・小容量メモリ | 処理時間 | s_m |
| 3 | | カウント可能なページ数 | i |
| 4 | | カウント可能な I/O 数 | n |
| 5 | | アクセス頻度 | B |
| 6 | 低速・大容量メモリ | 処理時間 | s_l |
| 7 | | カウント可能なページ数 | k |
| 8 | | カウント可能な I/O 数 | m |
| 9 | 事象 E の発生頻度 | P | |
| 10 | 事象 F の発生頻度 | Q | |
| 11 | 改善前：平均モニタ処理時間 | T_0 | |
| 12 | 改善後：平均モニタ処理時間 | T | |

(注) 変数は全て 0 以上であり、#3 と #4 は自然数とする。

以下、変数の説明及び関係について説明する。

I/O 発生時、高速・小容量メモリにアクセスする頻度を A、低速・大容量メモリにアクセスする頻度を B とする。ここで、A と B を足し合わせた全アクセス頻度を 1 とする。

次に、アクセス先メモリがどれかを判定する事象として 2 通り考えられる。

1 つ目は、高速・小容量メモリに該当するモニタ情報が格納されている場合である。この事象を E とし、発生頻度を P とする。一方、高速・小容量メモリにモニタ情報が格納なしで、低速・大容量メモリのみに格納されている場合は、発生頻度は 1-P である。

2 つ目は、事象 E において、カウント可能な I/O 数が上限を超えず更新可能な場合である。この事象を F とし、発生頻度を Q とする。一方、カウント可能な I/O 数の上限を超えている場合のアクセス頻度は 1-Q となる。

4.2 方式の数式化

4.1 節をもとに、提案方式の OVH 量 T と、方式改善前の OVH 量 T_0 は以下のように示せる。

$$T_0 = s_l \dots 1$$

$$T = s_m A + s_l B \dots 2$$

またアクセス頻度 A と B は、P と Q で以下のように示せる。

$$A = PQ \dots 3$$

$$B = 1 - PQ \dots 4$$

$$0 < P < 1, 0 < Q < 1 \dots 5$$

よって、式 2、式 3 と式 4 より、以下が示せる。

$$T = s_m PQ + s_l(1 - PQ) \dots 6$$

4.3 数式によるモニタ処理時間短縮の証明

次に、提案した採取方法は、改善前に比べモニタ処理時間を短縮できたかを数式より検証する。式 1 と式 5 は以下のように変形できる。

$$T_0/s_l = 1 \dots 7$$

$$T/s_l = 1 + (s_m/s_l - 1)PQ \dots 8$$

前提条件より $s_m < s_l$ であり、さらに式 5 より、式 7 と 8 は以下のように示せる。

$$T/s_l < 1 \dots 9$$

$$\therefore T < T_0 \dots 10$$

以上より、本方式によるモニタ処理時間の短縮を示せた。

5. 実運用を想定した効果検証

4.章で示した T の振る舞いをグラフ化し、実運用における効果を検証する。表 1 より、以下の関係を示す。また、前提条件は表 2 の通りである。

$$P = i/k \dots 11$$

$$Q = 1 - (1/n) \dots 12$$

表 2 前提条件[2]

| # | メモリ種別 | 性能比 | 容量比 |
|---|--------|-----|-----|
| 1 | 高速・小容量 | 6 | 1 |
| 2 | 低速・大容量 | 1 | 25 |

ただし、表 2 の容量比は同一コストと仮定した場合とする。以上より、以下の関係を示すことができる。

$$s_m : s_l = 1 : 6 \dots 13$$

$$i \cdot \ln(n^{-1}) : k \cdot \ln(m^{-1}) = 1 : 25 \dots 14$$

ここで、 $\ln(n^{-1})$ 及び $\ln(m^{-1})$ は、各メモリのカウント I/O 数の情報量を指す。また、SSD の IOPS 性能[3]より、1 日稼働させた場合に必要のカウント数を割り出した値 3.5G を変数 m の固定値とし、モニタ処理相対時間 T を縦軸、高速・小容量メモリのカウント情報量 $G(n)=\ln(n^{-1})$ を横軸としてグラフを作成し、図 3 に示す。

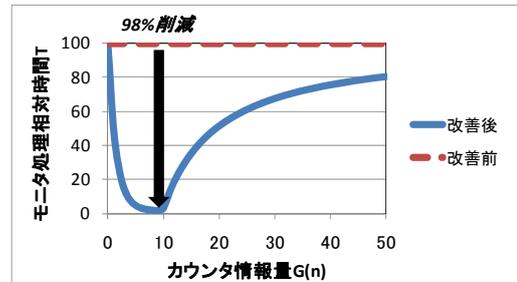


図 3 モニタ処理相対時間の変化

図 3 より、改善前に比べ、本方式によりモニタ処理時間を最大 98%削減可能であることが分かった。

6. まとめ

本研究では、ストレージ階層仮想化機能におけるページのアクセス頻度の採取について、低 OVH な方法を提案した。また、本方式の効果を検証した結果、目標を達成したことを確認した。

参考文献

[1] 須藤 梓ら, “ストレージ階層仮想化機能の実現方式検討”, 第 73 回情報処理学会全国大会講演論文集, Vol.1, No.1.31-32 (2011)
 [2] David A.Patterson and John L.Hennessy, “コンピュータの構成と設計”, 第 2 版, P.52 (2001)
 [3] Hitachi Global Storage Technologies, “UltraStar™ SSD400s”, Users Catalog, (2011)