情報処理学会論文誌 Vol.54 No.4 1592–1608 (Apr. 2013)

ストレージ自動階層配置機能におけるデータ再配置の最適化

坏 弘明[1,a)] 山本 彰[2] 大平 良徳[1] 江口 賢哲[3]

受付日 2012年7月30日, 採録日 2013年1月11日

概要：記憶媒体の多様化とデータ量の増大化，データ種類の多様化により，従来のストレージ階層を意識した性能設計や手動による適切な階層へのデータ配置が困難となってきている．この課題に応える一手段として，アクセス頻度が高いデータは高速な記憶媒体に，そうでないデータは大容量の記憶媒体に，自動的に格納するストレージ自動階層配置機能の普及が進んでいる．本研究はストレージ自動階層配置機能において，データ再配置の最適化を検討する．具体的には，階層の容量を考慮しつつ，アクセス頻度の高いデータを優先して移動するアルゴリズムを提案し，その最適性を証明する．また，データコピー処理とホスト I/O 処理のスケジューリング方式として，FCFS 方式と，優先度を考慮した HOL 方式を比較する．両者のホスト I/O レスポンスの振舞いを待ち行列による解析モデルで示し，HOL 方式が優位な方式であることを示す．さらに，実環境 I/O トレースによるシミュレーションを実施し，データ再配置により平均レスポンス時間を削減できる効果を定量的に確認した．

キーワード：ストレージ，自動階層化，データ再配置，SSD，スケジューリング，最適化

Optimization of Data Relocation Process in Automated Storage Tiering Functions

Hiroaki Akutsu[1,a)] Akira Yamamoto[2] Yoshinori Ohira[1] Yoshiaki Eguchi[3]

Received: July 30, 2012, Accepted: January 11, 2013

Abstract: By increase amount of data, it becomes to have difficulty in performance design and data placement to the appropriate tier by the hand operation. This problem begins to be solved by Automated Storage Tiering Functions. As for Automated Storage Tiering Functions, the small pieces of data are automatically relocating among the tier by access frequency. This study examines optimization of the data relocation in Automated Storage Tiering Functions. Specifically, we suggest an algorithm to give priority to data showing frequent access, and to move while considering the capacity of the tier and prove the most fitness. In addition, we compare the HOL method in consideration of FCFS method and priority with the data copy process as scheduling method of the host I/O process. We show the behavior of the host I/O response of both by the queuing analytic model and show that HOL method is a dominant method. Furthermore, we carried out simulation by the real environment I/O trace and confirmed the effect that could reduce mean response time by data relocation quantitatively.

Keywords: storage system, auto tiering, data relocation, SSD (solid state drive), scheduling, optimization

1. はじめに

多数の記憶デバイスを並列に束ねて大容量の記憶領域の一元管理を可能とする装置をストレージと呼び，IT システムにおいて非常に重要な役割を担っている．

ストレージでは，記憶媒体が多様化している．フラッシュメモリを活用し高速データアクセスが可能な SSD（Solid State Drive）や，高性能な SAS（Serial Attached SCSI）

[1] 株式会社日立製作所横浜研究所
Yokohama Research Laboratory, Hitachi Ltd., Yokohama, Kanagawa 244–0817, Japan
[2] 株式会社日立製作所研究開発本部
Research & Development Group, Hitachi Ltd., Yokohama, Kanagawa 244–0817, Japan
[3] 株式会社日立製作所 IT プラットフォーム事業本部
IT Platform Division Group, Hitachi Ltd., Odawara, Kanagawa 250–0872, Japan
[a)] hiroaki.akutsu.cs@hitachi.com

ハードディスクドライブ，大容量なSATA（Serial Advanced Technology Attachment）ハードディスクドライブ等，容量や価格，性能の異なるメディアが，ストレージに混在する環境となってきている．

また，他種多様な情報の電子化が進展し，企業内で管理するデータ量が爆発的に増加している．そのため，ストレージの管理するデータ量が増大している．現在ではペタバイトクラスの大容量ストレージを運用する企業も少なくない．さらに，非構造化データの増加や，サーバ仮想化の普及により，ストレージのボリューム内でアクセス頻度の異なるデータが混在している．

以上に述べたように，記憶媒体の多様化と，データ量の増大，データ種類の多様化により，従来のストレージ階層を意識した性能設計や，手動による適切な階層へのデータ配置が困難となってきている．

この課題に応える一手段として，ストレージ自動階層配置機能の普及が進んでいる．ストレージ自動階層配置機能は，アクセス頻度が高いデータは高速な記憶媒体に，そうでないデータは大容量の記憶媒体に，自動的に格納する．これにより，ストレージ階層を意識した性能設計が不要となるとともにアクセス頻度に応じた効率的なストレージ階層の利用が可能となるため，ストレージの性能向上とコスト低減の両立を実現することができる．

ストレージ自動階層配置機能は，データのアクセス頻度を小領域単位でモニタリングする機能と，移動対象の領域を決定して，階層間でのデータのコピー処理を周期的に実行するデータ再配置機能により実現する．特にデータ再配置機能では，アクセス頻度の高いデータを可能な限り高速に移動することを実現し，ホストI/O負荷への追従性を高めることが重要となる一方で，ホストI/Oへの性能影響を最小限とする必要がある．

そこで本研究では，（1）データ再配置における最適なページ選択アルゴリズムと，（2）データのコピー処理の最適化の2点を検討した．（1）のページ選択アルゴリズムについては，階層の容量を考慮しつつ，アクセス頻度の高いページを優先して選択するアルゴリズムを提案し，その最適性を証明する．（2）のデータのコピー処理においては，スケジューリング方式としてFCFS方式とHOL方式について，データのコピー処理によるI/Oとホスト I/Oが混在する場合のホストI/Oレスポンスの振舞いを待ち行列理論による解析モデルを示し，両者のどちらが優位な方法かを示す．

また，実環境（金融系のアプリケーション，サーチエンジン系のアプリケーション）から採取された複数のI/Oトレースを用いた，待ち行例シミュレーションを実施し，データ再配置によるホストI/Oの平均レスポンスの短縮効果を定量的に検証する．

2. 関連研究

本研究が対象とする，ストレージシステムにおけるデータ再配置に関する先行研究は，大きくボリュームベースの研究と，近年研究されてきているボリューム内の小領域単位（ページと呼ぶ）でデータ再配置を実施する，ページベースの研究に大別できる．

ボリュームベースの先行研究として，文献 [1], [2] 等があげられる．これらの研究では，ユーザの指定したQoS（Quality of Service）を維持しつつデータ再配置を高速に動作させることを目的としている．これらは，QoSの情報として，データ再配置による効果を表す関数やホストワークロードごとの優先度を指定する必要がある．しかし我々の対象としているデータ再配置の目的は，ストレージシステムにおけるプール全体の平均レスポンスの最小化であり，ストレージシステム内で閉じた，完全に自動的な最適化を目標としている点が異なる．また，これらの先行研究においては，ドライブは1種類を想定しており，レスポンス特性の異なる複数の階層の間でデータ再配置を実施する点が，本研究と異なる．

また，ページベースのデータ再配置を実施する先行研究として，文献 [3], [4] 等があげられる．これらの研究においては，複数の階層間でデータ再配置をしつつストレージシステムの平均レスポンスの最小化するという点を目標としており，その点に関して本研究と同じである．ただし，モニタリングと再配置を一定周期のサイクルで自動実行する方式における，ページの選択方法の最適性や，ページのコピー処理自体の最適性が考慮されていなかった．たとえば，文献 [3] においては，移動速度を一定の範囲に制限しており，文献 [4] においては，再配置によるドライブあたりのI/O実行多重度を1重に制限していると考えられる．この理由は，ホストI/Oに与える性能影響を削減するためと考えられる．本研究では，これらの再配置の効果をより最大限に引き出すための最適化を狙う．

また，近年のストレージシステムにおいて，SSDとHDDを自動的に使い分けるアプローチとして，SSDキャッシュと，本論文の対象とする自動階層管理の大きく2個のアプローチがある．SSDキャッシュの長所は，データの管理単位が小さいことによる負荷変動への高い追随性であり，一方，短所はデータの管理単位が小さいため，スケーラビリティ上の制約がある．本論文の対象としている自動階層管理の長所は，容量のスケーラビリティが高く，ペタバイトスケールのシステムにも適用可能であるという点である．一方，短所はページサイズがSSDキャッシュと比較して大きいため，負荷変動への追随性がSSDキャッシュと比較して低いという点があげられる．実際のシステムでは，両方の方法をうまく使い分けて使用する必要がある．本論文では，自動階層管理の短所である負荷変動への追随性を

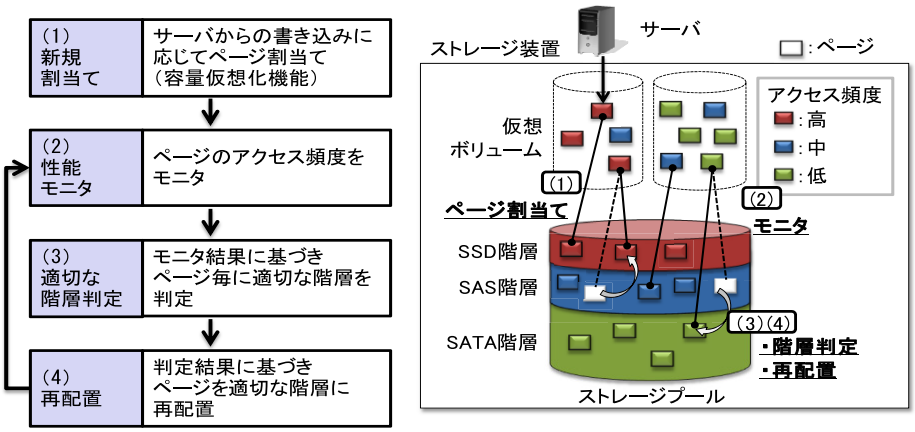

図 1 ストレージ自動階層配置機能の概要
Fig. 1 Overview of automated storage tiering functions.

向上させることを狙っている．

また，本研究では，再配置の最適性を議論するため，ストレージにおけるドライブのレスポンスを解析可能な待ち行列モデルにより定式化する．このアプローチの先行研究として文献 [5] 等があげられる．文献 [5] では非優先のバッチジョブと高優先の I/O 処理の振舞いを優先度付き M/G/1 待ち行列モデルにより定式化している．また，文献 [6] では，稼働率の均等化による負荷分散方法を提案している．本検討の対象とするシステムでは，サービス時間特性の異なる階層間で平均レスポンスを最小化することを目的としている点で，文献 [5], [6] とは異なる．そのため，本研究で対象とするシステムにおいては，文献 [6] のような階層間で稼働率均衡化を実施することが最適とは限らない．

3. ストレージ自動階層配置機能

本章では，ストレージ自動階層配置機能について説明する．

3.1 機能概要

ストレージ自動階層配置機能は，ストレージ領域のアクセス頻度の偏りに着目し，データのアクセス頻度に応じて適切な階層に自動的に再配置する機能である．ストレージ自動階層配置機能は容量仮想化機能（一般に Thin-Provisioning 機能とも呼ばれる）をベースとしている．容量仮想化機能とは，ストレージのボリューム内の小領域単位（以降ページと呼ぶ）で I/O アクセスに応じて割当て管理を実施する機能であり，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，近年のストレージでは一般的な機能となりつつある．

ストレージ自動階層配置機能の動作概要を図 1 に示す．ストレージ自動階層配置機能は，図に示すとおり，プール内で複数の階層を管理し，ページを管理単位として，アクセス頻度の採取と，アクセス頻度に応じた適切な階層への再配置を周期的に実行する機能を持つ．

本機能は，さまざまな用途に適用できる．たとえば，近年ビックデータと呼ばれ注目されている大量データの分析である．企業では，日々ビックデータが発生しているが，その分析処理，分析結果のビジネスへの反映等のプロセスを迅速化するため，フラッシュメモリを活用した SSD を採用することが考えられ，コストパフォーマンスの高い小容量の SSD と，ビットコストが低い大容量 HDD とをうまく使い分けることによりシステムのコストを削減することができる．また，多数のクライアントからのアクセスを支える基幹業務システムのストレージに適用することも考えられる．一般的な基幹業務を支えるデータベースシステムにおいて，小容量の高アクセス頻度のインデックス領域と，大容量の低アクセス頻度のテーブル領域があることが知られており，これらの格納先を効率的に管理するために，本機能は有効であると考えられる．本機能の対象とするストレージは，数十テラバイトからペタバイト規模クラスのスケーラビリティを持ったストレージシステムを想定しており，ターゲットとなるアプリケーションの対象とするデータセットが非常に大きい場合を想定している．

3.2 ストレージアーキテクチャ

以下に，本論文で説明するストレージ自動階層配置機能の適用対象の一例である，ストレージアーキテクチャを紹介する．

図 2 における「ポート」は，サーバ等の上位装置がストレージに接続する際の I/F を表している．具体的な実装形態は，FC（Fiber Channel）や，SAS，SATA，iSCSI（Internet Small Computer System Interface）等があげられる．

また，「キャッシュ」は，データのキャッシング用途のための DRAM（Dynamic Random Access Memory）等の高速な揮発性メモリを表す．また，「ドライブ」は，HDD（Hard Disk Drive）や，高速で容量の少ない SSD（Solid State Drive）等の記憶媒体を表す．一般に大規模なスト

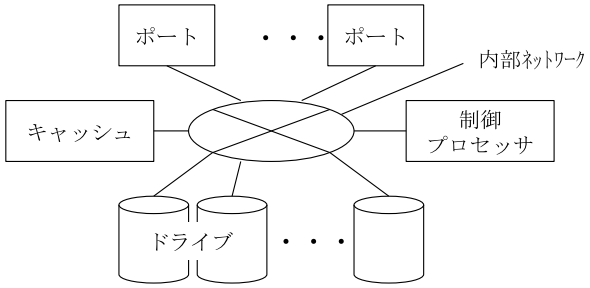

図 2 ストレージアーキテクチャの例
Fig. 2 An example of storage architecture.

レージアーキテクチャでは，HDD や SSD のような異なる性能特性を持つ記憶媒体（以降その分類を「階層」と呼ぶ）が混在している場合がある．

本論文の対象とするストレージアーキテクチャでは，複数のポートと複数のドライブ，キャッシュが内部ネットワークによりつながっており，それらの間でデータの転送を実施することができる．その転送処理を制御するのが，「制御プロセッサ」である．制御プロセッサも，内部ネットワークに接続される．

ストレージは，サーバからの I/O 要求（主に Read/Write）に応じて，I/O 処理を実行する．次に，I/O 処理の概要を説明する．まず，ポートから受信した I/O 要求を制御プロセッサが分析する．その結果，サーバからの I/O 要求が Read だった場合は，制御プロセッサがドライブに I/O 要求を発行することにより，ドライブ上の対象領域からデータを読み出し，内部ネットワークを経由してキャッシュにコピーする．その後，キャッシュからポートを経由してデータをホストに転送する．ここで，制御プロセッサは，データのキャッシュ格納の有無を管理し，ホストからの Read 対象のデータがキャッシュに格納されている場合は，ドライブから読み出すことなく，高速にホストへデータを転送することが可能である．また，サーバからの I/O 要求が Write だった場合は，制御プロセッサがホストから転送されたデータを内部ネットワークを経由してキャッシュにコピーし，ホストに Write 処理の完了を通知する．その後，非同期的にドライブに I/O 要求を発行することにより，キャッシュからドライブ上の対象領域へダーティデータを書き出す．以上のようにして，キャッシュ上にデータがある場合は，高速な Read/Write 処理が可能となる．また，キャッシュは，ストレージ内のドライブ間のデータコピー処理におけるバッファとしても用いられる．

次に，前述した制御プロセッサが発行するドライブへの I/O 要求に対する，ドライブのサービス時間について述べる．一般的に，HDD のサービス時間は，以下の式により与えられる．

HDD の I/O サービス時間 (S)
= シーク時間 + 回転待ち時間 + 転送時間

上式の「シーク時間」とは，HDD のヘッドをアクセス対象のトラック位置に移動させるための時間である．このシーク時間は，ドライブのベンダやモデルにより異なる．このため，平均シーク時間や，最大シーク時間が各ドライブのカタログ等に記載されている場合が多い．

「回転待ち時間」とは，読み出し対象のレコードの先頭が HDD のヘッドに回転してくるまでの時間を表す．この回転待ち時間は，HDD の回転速度により平均値が定まる．たとえば，HDD の回転速度が 15000 rpm であった場合は，最大の回転待ち時間は $60\,\text{sec} \div 15000\,\text{rpm} = 4\,\text{ms}$ となる．平均の回転待ち時間は，$4\,\text{ms} \div 2 = 2\,\text{ms}$ となる．

「転送時間」とは，HDD の円盤からデータを読み出す時間である．この転送時間は，ドライブのベンダやモデルにより異なる．このため，転送速度 [MB/s] がドライブのカタログ等に記載されている場合が多い．一般に記録密度が高いドライブほど，高速に転送することができるため，転送時間が短くなる．ドライブに I/O を発行する際に，転送長を指定する．この転送長が長いほど，1 回の I/O で転送できるデータ量が増加し，転送時間が長くなる．また，円盤の特性上，その外周と内周とで，同一の I/O 転送長でも，転送時間が異なる．

HDD のサービス時間（S）は，上述した，シーク時間，回転待ち時間，転送時間の総和により求まる．

ただし，HDD の物理的な特性として，上記のいずれの時間についても，Read/Write の場合で，それほど処理時間の差は発生しないことが多いといえる．

一方，SSD は HDD とは異なり，記憶媒体が半導体メモリであるため，ヘッドや円盤等，メカニカルな構造が削減されている．SSD のサービス時間は，一般的に以下の式で与えられる．

SSD の I/O サービス時間 (S)
= I/O 処理 OVH 時間 + 転送時間

上式の「I/O 処理 OVH 時間」とは，SSD 内部で 1 回の I/O を実行するのにかかる最短の時間である．この時間は，SSD を構成するチップのサイズにより異なる．たとえば，文献 [10] によると，8 GBit SLC（Single Level Cell）チップのリードレイテンシは実測値で約 40 μs，ライトレイテンシは約 300 μs となっている．

「転送時間」とは，SSD からのデータ読み出しまたは SSD へのデータ書き出し時の転送にかかる時間である．近年では，Flash チップに対して 200 MB/s〜400 MB/s のスループットの転送速度が一般的である．一般に，SSD には複数の Flash チップにより実装されるが，本論文では問題を単純化するため，SSD は 1 個の Flash チップにより実装されるものとする．また，上記以外にも，I/F のレイテンシや転送速度の OVH 等が加算されるが，これらの値は一般にドライブのサービス時間と比較して非常に小さいため，本

表 1 サービス時間を決定する要素
Table 1 Elements to decide a service time.

#	ドライブ種別	平均シーク時間	平均回転待ち時間	転送速度
1	SSD	0.17 ms		200.0 MB/s
2	SAS 15Krpm	3.00 ms	2.00 ms	137.5 MB/s

論文では問題を単純化するため，これらについても無視するものとする．

SSD のサービス時間（S）は，上述した，I/O 処理 OVH 時間，転送時間の総和により求まる．ただし，SSD の物理的な特性として，HDD とは異なり，Read/Write の場合で，処理時間の差が発生する．一般的には，より複雑な処理を必要とする Write のほうが Read よりも時間がかかることが多い．

文献 [8], [10] より算出した，ドライブ種別ごとのサービス時間を決定するパラメータを**表 1** に記載する．

前述したとおり，Read/Write や外周/内周で性能が異なるため，上述の表では，平均ケースの値を示している．表から分かるように，SSD は HDD と比較して大幅にサービス時間が短いため，ストレージアーキテクチャを前提としたホスト I/O のレスポンスも短い（すなわち性能が良い）．しかし，SSD は容量に対する価格が HDD よりも高く，大容量を使用できない．そこで，ストレージ自動階層配置機能は，I/O 頻度の高いデータをできるだけ SSD に配置し，I/O 頻度の低いデータはできるだけ HDD に配置することにより，ストレージ全体の価格を抑えつつ，ストレージ全体のレスポンスを短くすることを狙っている．

また，SSD で用いられるフラッシュメモリには大きく 2 種類があり，MLC 型（Multi Level Cell）と SLC 型（Single Level Cell）がある．MLC 型は許容される書き換え回数が小さい分価格が安いというメリットがある．SLC 型は MLC 型よりも高性能であり，許容される書き換え回数が多く，MLC 型よりも高価である．ストレージ自動階層配置機能は，I/O 頻度の高いデータをできるだけ SSD に配置する．したがって，アプリケーションやシステムの性能特性に応じて，これらをうまく使い分けて利用する必要がある．たとえば，基幹業務（後述する Financial トレースのような特性）向けには SLC 型を用い，データ分析やサーチエンジン等のリードメインの用途には（後述する Websearch トレースのような特性）MLC 型を用いること等が考えられる．

3.3 ページ管理

容量仮想化機能（Thin-provisioning）を実現するためのページ管理の概要について説明する．

容量仮想化機能の概要を**図 3** に示す．容量仮想化機能では，ホストに仮想的なボリューム（以降仮想ボリュームと呼ぶ）を認識させる．仮想ボリュームのサイズは，物理容量に

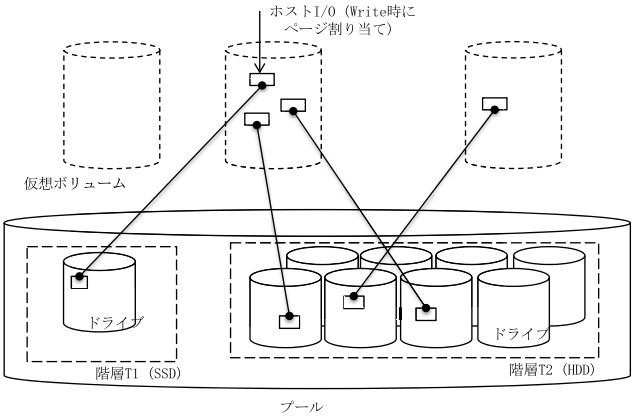

図 3 容量仮想化機能
Fig. 3 Thin-provisioning function.

依存せず，ユーザが自由に設定することができる．仮想ボリュームは，ページ単位でその物理領域との対応関係を制御プロセッサがダイナミックに管理する．仮想ボリュームに対して，ホストが初回の Write I/O を実行したときに，未使用の物理領域を新たに割り当てる（以降新規割当てと呼ぶ）．一方，物理領域は，多数のドライブをプールとして一元管理し，I/O 負荷を自動的にプール内で分散（ワイドストライピングと呼ぶ）しつつ，容量管理を一元化する．ページのサイズは小さいほど，物理容量の使用効率が向上するが，その分ダイナミックに管理するためのメタデータのサイズが増大化するため，適度なサイズ（数十 MB 単位）とする必要がある [9]．

また，プールを構成する階層が複数ある場合（たとえば SSD と HDD），ワイドストライピングは階層ごとに実施し，初回の Write I/O は，上位の階層（SSD）から優先的に割り当てる．

3.4 性能モニタ

ストレージ自動階層配置機能の一部の機能である，性能モニタについて説明する．性能モニタは，各ページごとのアクセス頻度を採取する．ページのアクセス頻度は，単位時間あたりの，制御プロセッサによるドライブへの I/O 要求発行回数を採取する．ホスト I/O によるポートの I/O 要求発行回数をとる方法もあるが，本機能の目的は，ドライブ負荷の最適化であるため，ドライブへの I/O 要求発行回数を採取する．ただし，後述するデータ再配置によるドライブへの I/O 要求発行回数は除外する．ストレージ自動階層配置機能は，ペタバイトクラスまでの容量を対象としているため，大量のページごとのモニタ情報を管理する必要がある．さらに，後述するようにそのモニタ情報の中から，高い負荷のページを効率的に発見する必要がある．これを実現する方法として，アクセス負荷量をレベル分けし，レベルごとのページ数のヒストグラムを生成し，レベルを閾値としてページを判定していく方法等が考えられる [11]．

3.5 データ再配置

ストレージ自動階層配置機能の一部の機能である，データ再配置について説明する．データ再配置は，プール全体の仮想ボリュームについて，ホストI/Oの平均レスポンスを短くすることを目的としている．プール全体の仮想ボリュームの平均レスポンス（$Resp_{pool}$）は以下の式で与えられる．

$$Resp_{pool} = \frac{\sum_{T \in Tier} Resp_T IO_T}{\sum_{T \in Tier} IO_T} \cdot (1 - CHitRate) \quad (1)$$

$Resp_T$ は各階層のI/Oレスポンス，IO_T は各階層におけるI/O頻度，$CHitRate$ はキャッシュのヒット率である．一般に，T1はSSD等のレスポンスの短いドライブを使用し，T2以降に比較的にレスポンスの長いHDDドライブを使用する．そのため，高いI/O頻度のページをT1に移動することにより，レスポンス時間の短いT1にホストI/O負荷を集中させることにより，プール全体の平均レスポンスを短くすることができる．また，キャッシュヒット時のレスポンスは非常に小さいため，0秒としている．

T1に最大限ホストI/O負荷を集中させるためには，プール全体でページをI/O頻度の高い順番でT1に格納すればよい．

ただし，T1にホストI/O負荷を詰め込みすぎると，逆にT2のレスポンスよりもT1のレスポンスが悪くなるため，一定負荷量以上詰め込まないように制限する必要がある．

以上のように，本論文では，T1の負荷量を一定の範囲を超えない範囲で最大化することを目標とするため，I/O負荷の偏りが大きく，T1にホストI/O負荷を詰め込みすぎる可能性のあるケースにおいては，必ずしもT1に負荷の高い順番で格納されていなくても，再配置を完了してよいものとする．

データ再配置は，性能モニタが採取したアクセス頻度に応じた適切な階層へのデータの再配置を周期的に実行する．モニタ・再配置の周期動作の概要を**図4**に示す．前節で述べた性能モニタは，一定の周期（30分〜1日程度）で採取し，前周期のページごとの性能モニタの情報を基に，再配置処理を動作させる．

再配置処理は，**図5**に示すように，（1）再配置を実行対象のページ選択処理と，（2）そのページの再配置先の階層へのコピー処理，（3）ページマッピングの変更と再配置元のページの解放処理によって成り立つ．（1）については，次に説明する制御目標を満たすように，ページ選択アルゴリズムを後ほど述べる．（2）については，再配置元のページが割り当てられた階層の対象ドライブに，制御プロセッサがI/O要求を発行し，キャッシュにデータを読み出す．その後，制御プロセッサが再配置先の階層の空きページを割り当て，対象ドライブにI/O要求を発行し，キャッシュ上のデータをドライブへ書き込む．これらのデータコピー処理は，制御プロセッサ上ではI/O処理とは別の非同期処理のジョブとして実行される．また，上位階層への移動はPromotion，下位階層への再配置はDemotionと呼ぶ．（3）については，再配置元のページを再利用するために仮想ボリューム上から割当てを解放する処理であるが，本論文では特に取り扱わないため，詳細は割愛する．

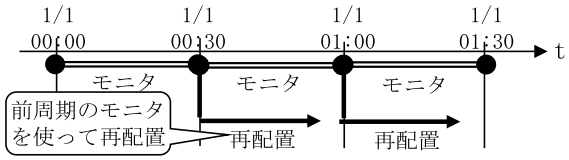

図4 モニタ・再配置の周期動作
Fig. 4 Cycle process of monitoring and relocation.

データ再配置の最終的な目的は，ホストI/Oの平均レスポンスをできるだけ短くすることである．これを実現すべく，制御目標は，以下の4点とした．

［再配置対象ページ選択の最適化］

制御目標1：データ再配置完了時にホストI/OのT1（最上位の階層）稼働率を最大化すること（ただし稼働率は閾値 UF_{T1} 以内）

制御目標2：再配置中のホストI/Oの平均T1稼働率を最大化すること（ただし再配置中のホストI/Oの稼働率は閾値 UF_{T1} 以内）

［データコピー処理の最適化］

制御目標3：コピー速度を最大化すること（特定の階層のホストI/Oを除いた余剰分のドライブ性能を使いきること）

制御目標4：上記を満たしたうえでコピー処理のホストI/Oペナルティを可能な限り低減させること

制御目標1について，アクセス頻度の高いページをT1に配置したほうがプール全体のホストI/Oレスポンスを短縮できる．ただし，T1の稼働率が一定以上となると，T1が高負荷状態となり，逆にレスポンスが悪化することとなるため，T1のホストI/Oの稼働率の閾値 UF_{T1} を設け，この閾値を超えない範囲で最大とすることを目標としている．

制御目標2について，再配置中のホストI/Oの平均T1稼働率を最大化するとは，再配置完了時点のT1稼働率（最大）への収束を最速にするという意味である．これは，ホストI/Oの平均レスポンスを最小化するために必要な条件である．

制御目標3について，コピー速度を最大化するとは，データ再配置時間を最短にすることを目的としている．この最短化により，ホストI/Oの負荷変化に対して，追従性を高め，ホストI/Oの平均レスポンスを最短とすることができる．コピー速度の最大化を実現するためには，ホストI/O処理を除いたドライブの余剰分を使いきってコピーによるドライブ稼働率を最大化することにより実現できる．階層ごとにドライブのスループット性能は異なるため，ある特

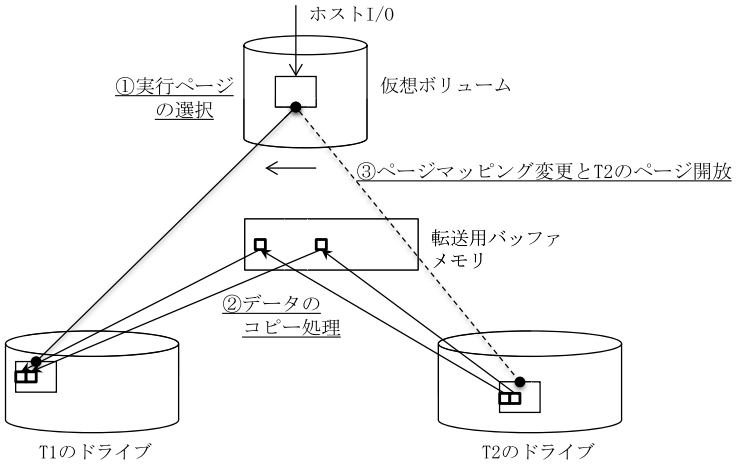

図 5 再配置処理の動作（T2 から T1 への Promotion）
Fig. 5 Relocation process (Promotion from T2 to T1).

定の階層のドライブ稼働率が 100%となると，それ以上は他の階層のコピー処理を高速化できない．したがって，コピー速度の最大化とは，特定の階層のホスト I/O を除いた余剰分のドライブ性能を使いきることである．

制御目標 4 について，制御目標 3 でコピー速度を最短としたが，コピーを高速に動作させることによりホスト I/O のペナルティが多くなる懸念がある．そこで，ホスト I/O とコピーによる I/O スケジューリング（FCFS または HOL）を検討することにより，制御目標 3 を満たしつつ，ホスト I/O のペナルティを低減させることを目標とする．

これらの制御目標を可能な限り達成すべく，本論文ではデータ再配置における実行ページ選択の順序最適化（制御目標 1，2 の実現）とコピー処理の最適化（制御目標 3，4 の実現）について述べる．

4. データ再配置における実行ページ選択の順序最適化

本章では，データ再配置における最適なページ選択アルゴリズムを提案し，その正当性（制御目標 1 および 2 を満たす）について証明する．

4.1 実行ページ選択の順序最適化アルゴリズム

本アルゴリズムは，以下の仮定に基づいている．

［仮定 1］ 2 階層（T1，T2）のドライブ構成を仮定する．

［仮定 2］ 再配置中にホスト I/O の特性は変わらないと仮定する．

［仮定 3］ 再配置開始時の T1 のホスト I/O による稼働率は閾値 UF_{T1} を超えないと仮定する．

［仮定 4］ T2 の容量は十分あるものとし，容量の枯渇は発生しないと仮定する．

［仮定 5］ 1 ページあたりの負荷は全ページの負荷に対して十分少ないと仮定する．

［仮定 1］について，階層の数を 3 以上とすると，アルゴリズムが複雑となり，説明が煩雑となるため，本論文では単純な 2 階層のケースを前提とする．

［仮定 2］について，モデル上でアルゴリズムの純粋な最適性を評価するためである．ただし，6 章で，実 I/O トレースを用いたシミュレーションを実施し，一般的な特性の変化のケースについても効果があることを示す．

［仮定 3］について，T1 稼働率が閾値 UF_{T1} を超えている場合（T1 がネックのとき）は T1 の高負荷ページを T2 へ Demotion するのが最善であるというケースが発生し，アルゴリズムが複雑となる．ただし，T1 が十分性能の良いドライブ（SSD 等）であれば，T1 がネックとなるケースは一般に少ないため，説明が煩雑となることを防ぐため，T1 の稼働率は閾値 UF_{T1} を超えないと仮定した．

［仮定 4］について，一般に T1 の容量と比較し T2 の容量は数倍～数十倍となる．また，T2 の容量枯渇となる前にプールを増設するのが一般的であるため，説明が煩雑となることを防ぐ目的で，T2 の容量は十分あり，容量の枯渇は発生しないと仮定した．

［仮定 5］について，閾値 UF_{T1} の近傍を狙う真に最適なページ配置の組合せパターンを算出しようとすると，明らかに計算コストが非常に高い．しかしながら，現実的な運用の範囲では，使用しているページ数が十分多く，最低でも 1 ページ分の負荷は全体の負荷量に対して誤差の範囲である程度のページ数をプール内で使用していることが多いため，この条件を仮定した．

図 6 に，実行ページ選択アルゴリズムを示す．図 6 の左下は，T1 容量の空きがある（T1 のページ使用率 C_{T1} が 1 より少ない）場合で，T1 の稼働率にゆとりがあれば（T1 のページごとの I/O 頻度 F の累計値 F_{T1} が許容累計値 UF_{T1} 以上ではない場合），T2 内でページごとの I/O 頻度の最も高いページを T1 に Promotion することを示している．また図 6 の右下は，T1 容量の空きがない場合の動作をあらわし，T2 の最大よりも負荷の低い T1 のページが

```
LOOP {
 IF (F_T1 + max(Fj∈T2) < UF_T1) ∧ (C_T1 < 1) THEN promotion(j)
 ELSE IF (min(Fi∈T1) < max(Fj∈T2)) ∧ (C_T1 == 1)
   THEN demotion(i)
 ELSE END
}
```

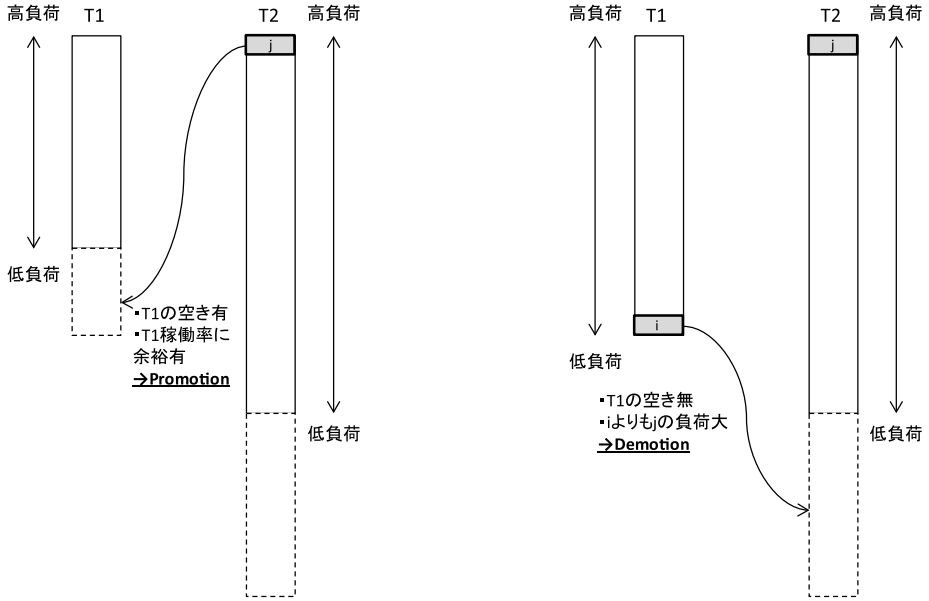

図 6 実行ページ選択アルゴリズム
Fig. 6 Execution page selection algorithm.

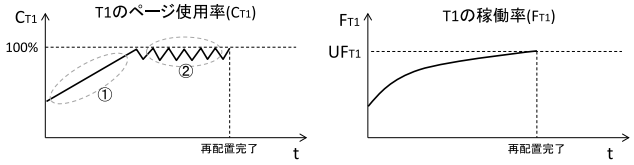

図 7 実行ページ選択アルゴリズムの動作
Fig. 7 Action of the execution page selection algorithm.

あれば，T1 から Demotion し，その他の場合は，完了する．また，F_{T1} や UF_{T1} はホスト I/O による I/O 頻度の累計値を表し，再配置による I/O は比較の対象に含まないものとする．

実行ページ選択アルゴリズムの動作イメージをタイムチャートで図 **7** に示す．

図 7 の ① の部分は，T1 容量に空きがあるため，空きがなくなるまで，プロモーションのみを実行している．図の ② の部分で，T1 容量の空きがなくなったら，デモーションとプロモーションを繰り返すことでページの再配置を継続的に実行する．T1 の稼働率が一定の限界に到達したら，再配置が完了する．

4.2 アルゴリズム正当性の証明

実行ページ選択アルゴリズムの正当性の証明として，停止性と最適性を証明する．

4.2.1 停止性の証明

本アルゴリズムでは，Promotion を実施していないページ数が単調減少する．つまり，1 度 Promotion したページを再度 Demotion しなければ，有限ステップで停止することになる．本アルゴリズムでは，ステップが経過するたびに，$\min(Fi \in T1)$ は増加し，$\max(Fj \in T2)$ は減少する方向にしか進展しないため，$\min(Fi \in T1) \geqq \max(Fj \in T2)$ が成立しない限り，1 度 Promotion したものを再度 Demotion することはない．しかしながら，$\min(Fi \in T1) \geqq \max(Fj \in T2)$ が成立した時点で，このアルゴリズムは終了するため，本アルゴリズムは必ず終了するといえる．□

4.2.2 最適性の証明

コピー処理自体の負荷影響については，5 章で論じるため，本章では 1 ページのコピー時間が一定でホスト I/O への性能影響がないものと仮定した場合に，3 章で定義した制御目標 1 および 2 を達成できることを証明する．具体的には，制御目標 1 で最終状態の T1（最上位の階層）の稼働率が最大（ただし閾値 UF_{T1} を超えない）になること，制御目標 2 で再配置中のホスト I/O の平均 T1 稼働率を最大化すること，を示すことにより最適性を証明する．

制御目標 1 については，アルゴリズムの終了時点で，条件 1 $(\min(Fi \in T1) \geqq \max(Fj \in T2)) \vee (C_{T1} < 1)$ と，条件 2 $((F_{T1} + \max(Fj \in T2) \geqq UF_{T1}) \vee (C_{T1} == 1))$

が成立していることになる．まず，条件 1 の $\min(Fi \in T1) \geqq \max(Fj \in T2)$ と条件 2 の $(C_{T1} == 1)$ が両方成立しているのであれば，完全なページ上詰状態（T1 に存在するすべてのページの負荷が T2 のページの負荷を超えていて，T1 の容量使用率が 100%）なので，最適状態である．条件 1 の $\min(Fi \in T1) \geqq \max(Fj \in T2)$ と条件 2 の $F_{T1} + \max(Fj \in T2) \geqq UF_{T1}$ が両方成立している状態も，T1 が性能飽和状態（次にページ移動したら閾値 UF_{T1} を超える）なので，最適状態である（前提（5）で，1 ページあたりの負荷量は無視できるほどの量であると仮定しているため）．条件 1 の $C_{T1} < 1$ と条件 2 の $F_{T1} + \max(Fj \in T2) \geqq UF_{T1}$ が両方成立している状態も，T1 が性能飽和状態なので，最適状態である．条件 1 の $(C_{T1} < 1)$ と条件 2 の $(C_{T1} == 1)$ は同時に成立しえない．以上ですべての停止状態の条件において，ホスト I/O の T1 稼働率を最大化（ただし閾値 UF_{T1} を超えない）できる．

制御目標 2 について，本アルゴリズムでは，T1 の稼働率が最も最大化されるページ移動（Promotion）から先に実施し，T1 の稼働率が減るページ移動（Demotion）はできるだけ後で（T1 容量の空きがないときに限り実施）かつ T1 の稼働率の減少量が少ないページから実施するため，T1 の稼働率の時間的な積分量を最大化することができる．したがって，再配置中のホスト I/O の平均 T1 稼働率を最大化することができる． □

5. データ再配置におけるコピー処理の最適化

本章では，データ再配置におけるコピー処理の動作を，待ち行列モデルを用いて定式化し，制御目標 3 および 4 を満たす，最適な制御方法について論じる．

5.1 コピー処理における課題

データ再配置にはデータのコピーが発生するので，ホスト I/O レスポンスのペナルティがともなう．したがって，最適なコピー処理動作（ホスト I/O とコピー処理のスケジューリング方法，コピー処理の最適な I/O 転送長）の検討が必要である．

コピー処理最適化の検討における前提条件は，4 章で説明した前提条件に加えて，以下である．

[仮定 1] ポートやキャッシュのアクセス，内部ネットワーク間の通信，プロセッサ処理のレイテンシはないものとし，またキャッシュのヒット率は 0 と仮定する．

[仮定 2] ドライブによる I/O 処理のサービス時間は，一定分布と仮定する．

[仮定 3] コピー処理のためのバッファ量は制限なしと仮定する．

仮定 1 については，一般に，ポートやキャッシュのアクセス，内部ネットワーク間の通信，プロセッサ処理のレイテンシは，ドライブのレスポンス時間と比較すると極小であるため，無視できるものとし，0 とした．また，キャッシュのヒット率にかかわらず，ドライブレスポンスが短縮されたら，ホストレスポンスもその分改善されることが期待できるため，本論文での方式比較の目的の範囲では，キャッシュのヒット率を 0 と考えても，有効と考える．この前提条件により，ドライブのレスポンス時間 = ホスト I/O のポートにおけるレスポンス時間と単純化して考えることができる．

仮定 2 については，3.2 節で説明したサービス時間の式について，HDD の転送時間は一定分布の特性を持つが，シーク時間と回転待ち時間の分布は厳密には一定分布ではない．ただし，最大振れ幅（最大シーク時間，最大回転待ち時間）が決まっており，指数分布よりは一定分布に近いと考えられる．また SSD には 3.2 節で説明したとおり，メカニカルな部分がないため，一般的に一定分布に近いと考えられる．そこで，本論文では，ドライブによる I/O 処理のサービス時間は，すべて一定分布と仮定し，単純化した．

仮定 3 については，バッファ量がコピー処理のスループット性能に影響を与える [12] が，本論文では純粋なスケジューリング方式の比較という目的のため，解析が容易となるようにバッファ量に制限を持たないものと仮定して，単純化した．ただし，後ほどのシミュレーションによる評価で，この影響について考察する．

5.2 待ち行列によるモデル化

データ再配置によって，ドライブに対するコピー I/O とホスト I/O のアクセス要求が混在する．複数の種類の I/O が混在する場合のスケジューリング方式として，よく知られた方法は FCFS（first-come first-served）である．また，その他の方式としてホスト I/O を Non-preemptive に優先処理する HOL（Head of Line）が考えられる．本節では，この 2 種類の I/O スケジューリング方式のうち，どちらがコピー処理に最適かを検討する．

5.2.1 FCFS モデル

FCFS による I/O スケジューリング方式を**図 8** に示す．各記号の定義を**表 2** に示す．

FCFS による I/O スケジューリング方式では，ドライブごとに 1 個のキューを持ち，ホスト I/O によるドライブへの I/O と，データ再配置のコピー処理によるドライブへの I/O を混在させる．ただし，各キューで同時に混在できるコピー処理による I/O は 1 個であるとする（コピーがドライブごとに 1 多重で動作）．1 個とした理由は，この個数を増やせば増やすほどホスト I/O 側のペナルティとしての待ち時間が増大していくためである．

一般に，キューにおける平均待ち時間は，キューにおいて待っているエントリの平均サービス時間の総和と，サービス中のエントリのサービス時間の平均残余時間の総和の合計で算出できる．キューにおける平均待ち時間につい

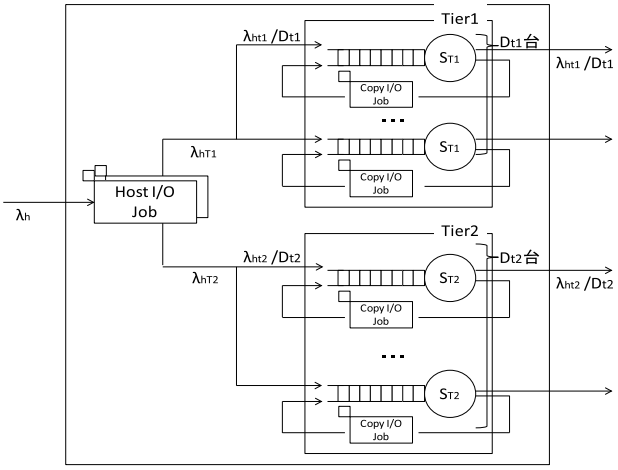

図 8 FCFS による I/O スケジューリング方式
Fig. 8 I/O scheduling method by FCFS.

表 2 記号の定義
Table 2 Definition of symbols.

記号	意味
W_0	平均残余時間
$\frac{W_p}{S_i^2}$	優先度 p の平均待ち時間
	サービス時間の 2 次モーメント
ρ_r	データ再配置の稼働率
S_r	データ再配置の平均サービス時間
ρ_h	ホスト I/O の稼働率
S_h	ホスト I/O の平均サービス時間
λ_h	ホスト I/O の平均到着率
Z_r	データ再配置のスループット
$Resp_{Tx}$	階層毎の平均レスポンス
d_{Tx}	階層毎のドライブ台数
ρ_{hTx}	階層毎のホスト I/O の稼働率
S_{rTx}	階層毎のデータ再配置の平均サービス時間

て，以下に式を示す．

$$W = XS_h + YS_r + \frac{\overline{S_h{}^2}}{2S_h}\rho_h S_h + \frac{\overline{S_r{}^2}}{2S_r}\rho_r S_r$$

X はホスト I/O のキュー上の平均リクエスト数を表す．また，Y は再配置の I/O のキュー上の平均リクエスト数を表す．ここで，Y は，コピー I/O ジョブの多重度を 1 としているため，最大でも 1 である．さらに，再配置がサービス実行中である確率は ρ_r であるため，再配置の I/O がキュー上にある確率，つまり平均リクエスト数は $Y = 1 - \rho_r$ である．さらに，リトルの公式より，$X = W\frac{\rho_h}{S_h}$ であるため，これらの式を代入すると，以下の式を得る．

$$W = \frac{(1-\rho_r)S_r + \rho_r\frac{\overline{S_r{}^2}}{2S_r} + \rho_h\frac{\overline{S_h{}^2}}{2S_h}}{(1-\rho_h)} \quad (2)$$

本モデルにおいて，仮定 2 のストレージシステムの特性上，サービス時間を一定分布（$\frac{\overline{S_x{}^2}}{S_x} = S_x$）と仮定し，仮定 3 によりドライブの稼働率の余剰を再配置の I/O により完全に使いきる（$1 = \rho_h + \rho_r$）と仮定すると，各階層のドライブ I/O の平均レスポンス時間は，以下の式で算出できる．

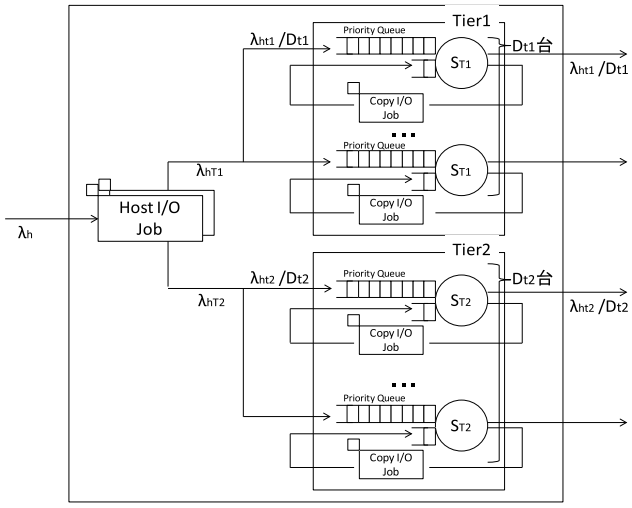

図 9 HOL による I/O スケジューリング方式
Fig. 9 I/O scheduling method by HOL.

$$Resp_{Tx} = \frac{(1+\rho_{hTx})S_{rTx} + \rho_{hTx}S_{hTx}}{2(1-\rho_{hTx})} + S_{hTx} \quad (3)$$

仮定 1 より，各階層のドライブ I/O の平均レスポンス時間の，各階層への I/O 到着頻度との加重平均によって，ホスト I/O の平均レスポンス時間を算出できる．

また，再配置のスループット（Z_r）は，仮定 3 より，以下で表される．

$$Z_r = \min\left(\frac{(1-\rho_{hT1})d_{T1}}{S_{rT1}}, \frac{(1-\rho_{hT2})d_{T2}}{S_{rT2}}\right) \quad (4)$$

5.2.2 HOL モデル

次に，HOL による I/O スケジューリング方式を図 **9** に示す．HOL による I/O スケジューリング方式では，ドライブごとにキューを 2 個持ち，1 個のキューをホスト I/O によるドライブ I/O を Non-preemptive に優先処理を実施するために用いる．もう 1 つのキューは，データ再配置のコピー処理によるドライブへの I/O を低優先で実施するために用いる．各キューにおいて，低優先のキューの要求は，高優先のキューが空の場合のみ実行されるようにすることで，高優先の処理を優先で実行する．

異なるサービス時間の特性を持つジョブの混在時のレスポンスは，M/G/1 待ち行列モデルによりモデル化できる．したがって本モデルを用い，ドライブの平均レスポンス時間をモデル化する．一般的なドライブの I/O アクセスによるデータ転送処理は，定まった転送単位以外での点での実行中に中断することが困難である．その理由は，データ転送自体が異常終了してしまうためである [5]．したがって，定まった転送単位により再配置 I/O を実行させるものとし，その実行中は割込みが発生しないものとする．

一般に，このような単位ジョブの実行中に割込みが発生しない優先度モデルは，HOL non-preemptive priorities の公式により [6]，キューにおける平均待ち時間を算出でき

る．以下に式を示す．

$$W_p = \frac{W_0}{(1-\sigma_p)(1-\sigma_{p+1})} \qquad p = 1, 2, \cdots, P$$

ここで，p の値が大きいほうが優先度大である．また，σ_p および W_0 は以下の式で表される．

$$\sigma_p = \sum_{i=p}^{P} \rho_i \qquad W_0 = \sum_{i=1}^{P} \rho_i \frac{\overline{S_i^2}}{2S_i}$$

ここで，W_0 は平均残余時間，W_p は優先度 p の平均待ち時間を表す．

上式によると，優先（$p = 2$）であるホスト I/O によるキューの平均待ち時間は，以下の式で算出できる．

$$W_2 = \frac{\rho_r \frac{\overline{S_r^2}}{2S_r} + \rho_h \frac{\overline{S_h^2}}{2S_h}}{(1-\rho_h)} \tag{5}$$

仮定 2 のストレージシステムの特性上，サービス時間を一定分布（$\frac{\overline{S_x^2}}{S_x} = S_x$）と仮定すると，以下の式を得る．

$$Resp_{Tx} = \frac{\rho_{rTx} S_{rTx} + \rho_{hTx} S_{hTx}}{2(1-\rho_{hTx})} + S_{hTx} \tag{6}$$

仮定 1 より，各階層のドライブ I/O の平均レスポンス時間の，各階層への I/O 到着頻度との加重平均によって，ホスト I/O の平均レスポンス時間を算出できる．

また，再配置のスループット（Z_r）は，以下で表される．

$$Z_r = \min\left(\frac{(1-\rho_{hT1}) d_{T1}}{S_{rT1}}, \frac{(1-\rho_{hT2}) d_{T2}}{S_{rT2}} \right) \tag{7}$$

5.3 I/O スケジューリングの比較検討

前節で，I/O スケジューリング方式として，FCFS と HOL について，ホスト I/O の平均レスポンス時間と再配置スループットを待ち行列モデルを使って定式化した．本定式を用いて，I/O スケジューリングを比較検討する．

図 10 に，T1（SSD）の再配置スループットの比較を例として示す．SSD のサービス時間として，$S_h = 0.19\,\text{ms}$，$S_r = 2.67\,\text{ms}$ を仮定している．式 (4) および式 (7) は同一であり，かつドライブの稼働率を最大限に引き出すことができているため，両方式は制御目標 3 を満たすことができる．

また，**図 11** に，T1（SSD）ネック時の T1 のドライブレスポンスの比較を例として示す．SSD のサービス時間は図 10 と同様の仮定である．式 (3) および式 (6) を比較すると，HOL のほうがつねに階層ごとのドライブ I/O の平均レスポンス時間レスポンスが小さいことが分かる．また，式 (1) から，プール全体のホスト I/O レスポンスは，各階層のドライブ I/O の平均レスポンスの I/O 頻度による加重平均によって算出される．式 (3) および式 (6) から，いかなるホスト I/O の稼働率の場合も HOL のほうが FCFS よりもレスポンスが短いことが分かっているので，式 (1) で同じ階層間の I/O 比率で加重平均をとった場合も，HOL のほうがレスポンスは短くなる．したがって，制御目標 4 については，より HOL が適していると結論付ける．

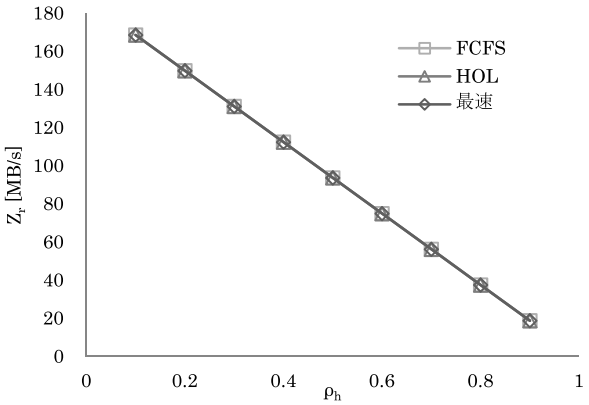

図 10 再配置スループット比較（T1）
Fig. 10 Relocation throughput (T1).

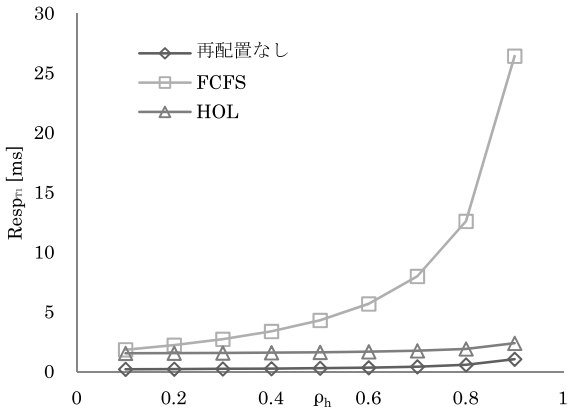

図 11 ドライブレスポンス比較（T1）
Fig. 11 Drive response time (T1).

6. シミュレーション評価

本章ではシミュレーションによる評価について，条件と得られた結果を記載する．

6.1 シミュレーション条件

実環境（金融系のアプリケーション，サーチエンジン系のアプリケーション）から採取された複数の I/O トレースを用いた，待ち行例キューのシミュレーションを実施し，ホスト I/O の平均レスポンスの短縮効果を検証する．シミュレーションの条件を**表 3** に示す．

また，本シミュレーションでは，ページ移動のための転送用バッファメモリの使用量は最大で 1 ページ分（32 MB）とする．よって，ページの Promotion/Demotion は，1 ページごとに順番に動作するものとして，並行には動作しないものとする．ただし，1 ページ内のデータコピー処理は多重に動作する．この並列動作の概要を**図 12** に示す．

表 3 で示しているが，1 ページは 32 MB とし，再配置

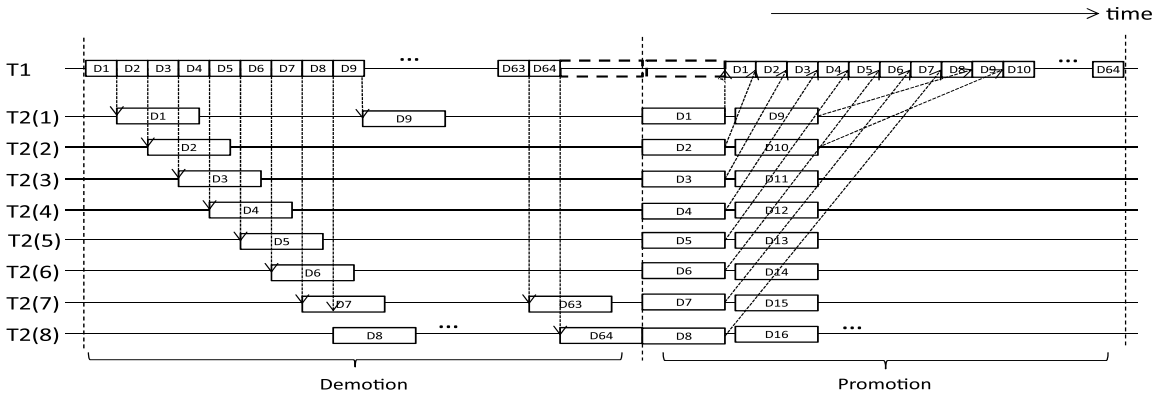

図 12 データコピー処理のタイムチャート
Fig. 12 Timing diagram of the data copy process.

表 3 シミュレーション条件
Table 3 Simulation conditions.

#	大項目	項目	条件
1	ドライブ構成	T1: SSD	S_h：0. 190 ms S_r：2.670 ms d_{T1}：1 台 （論文 10）より算出，再配置は 512KB で計算，ホスト I/O は 4KB で計算）
2		T2: SAS 15Krpm	S_h：5.028 ms S_r：8.636 ms d_{T2}：8 台 （ドライブスペック 8）より算出，再配置は 512KB で計算，ホスト I/O は 4KB で計算）
3	ストレージ設定	RAID レベル	0（ストライピングのみ）
4		ストライプ	512KB 単位(SAS15K は 8 ドライブでストライプを構成，SSD はそのまま使用)
5		ページサイズ	32MB
6		キャッシュ	キャッシュメモリ量は容量の 0.1% とする（8 多重時で，Financial は 156MB，Websearch は 590MB とする）．キャッシュアルゴリズムは LRU とする．
7		モニタ周期	30 min
8	ホスト I/O トレース	全般	別々の領域に 8 多重で負荷を掛ける
9		Financial1	金融系 I/O のトレース，平均 1012 IOPS(8 多重時)，約 12 時間，T1 容量は 5GB/25GB で分析
10		Financial2	金融系 I/O のトレース，平均 722IOPS(8 多重時)，約 11 時間，T1 容量は 5GB/25GB で分析
11		Websearch2	サーチエンジン系 I/O のトレース，平均 2377IOPS(8 多重時)，約 4 時間，T1 容量は 60GB/100GB で分析
12		Websearch3	サーチエンジン系 I/O のトレース，平均 1504IOPS(8 多重時)，約 6 時間，T1 容量は 60GB/100GB で分析

によるコピー処理は 512 KB 単位で実行するため，64 回の I/O（対応する I/O 処理を D1～D64 で図示）をそれぞれの階層のドライブに発行することになる．また，512 KB 単位の転送バッファへのリード処理と転送バッファからのライト処理は，図に示すように，原理的に逐次的な実行となる．以上のような前提の場合，図に示すように，全体の再配置スループットとして T1 ネックであるが，T1 が稼働しない時間が発生する．この時間の影響のため，コピー処理によるドライブ稼働率が完全に 100%とはならないことになる．この影響を含め，本シミュレーションで効果を分析する．また本シミュレーションでは，UF_{T1} を 60%と仮定した．

また本シミュレーションでは，RAID0（ストライピング）を仮定した．ストライピングの場合，論理記憶領域を 512 KB 単位で分割し，Tier 内の各ドライブに対して分散配置する．分散配置した各領域に対して負荷が完全に分散した場合，各ドライブへの I/O 到着頻度は，$\frac{\lambda_{hTx}}{d_{Tx}}$ となる．

6.2 トレースデータの特性

本シミュレーションで用いる，実環境（金融系のアプリケーション，サーチエンジン系のアプリケーション）から採取された複数の I/O トレース [7] について，その特性を示す．本トレースデータは，ホスト側で採取されたものであるため，ストレージシステム側のキャッシュの影響を考慮していない．そこで，より現実的な結果を得るため，ストレージで一般的に用いられるキャッシュアルゴリズムである，LRU（Least Recently Used）を模擬するシミュレーションを実施し，キャッシュ動作を介したドライブアクセスのトレースを用いてシミュレーションを実施する．一般的な商用のストレージシステム [13] では数 PB のサポート容量に対して，数百 GB～数 TB のキャッシュメモリを搭載できるものが多い．したがって，本論文ではストレージ容量に対して 0.1%のキャッシュ量を想定し，シミュレーションを実施した．**表 4** に，トレースデータのキャッシュヒット率を示す．一般的なストレージシステムでは，ライト I/O 処理は内部の 2 重化されたキャッシュメモリにデータを格納した時点でホストにライト完了を通知するため，ドライブのスループットが限界となるアクセス量でない限り，つねにキャッシュヒットする．そのため通常はライトのキャッシュヒット率は 100%となるため，表 4 には記載していない．また，キャッシュ上のライトデータが LRU

表 4 トレースデータのキャッシュヒット率
Table 4 Cache hit ratio of trace data.

#	トレース名	Read 率	ReadHit 率	実 Write 率
1	Financial1	23.09%	33.07%	27.47%
2	Financial2	82.29%	20.95%	60.61%
3	Websearch2	99.98%	0.05%	99.00%
4	Websearch3	99.97%	0.07%	98.73%

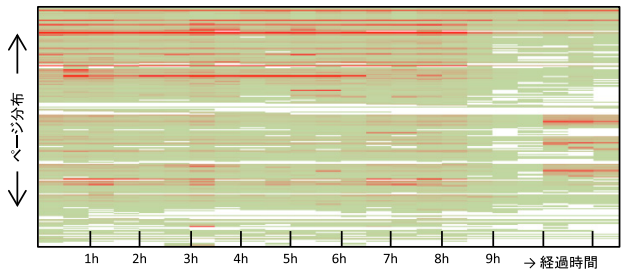

図 13 アクセスヒートマップ（Financial2）
Fig. 13 Access heat map of trace data (Financial2).

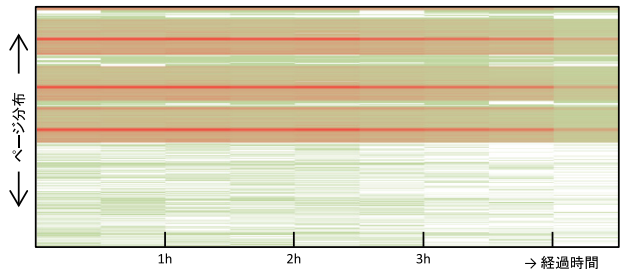

図 14 アクセスヒートマップ（Websearch2）
Fig. 14 Access heat map of trace data (Websearch2).

に遷移し，ドライブに書き込まれる前に，再度同じ領域にライト I/O が発生すると，キャッシュ上でのみライトデータが上書きされるため，ドライブへのライト I/O 発行回数を削減することができる．表 4 の “実ライト率” はこの特性を示しており，ホストからのライト I/O 発行回数に対する実際のドライブ I/O 発生回数の割合である．

表 4 に示すように，Websearch のトレースデータではライトがほとんど発生していない．この理由は，ストレージへの I/O は検索のための索引やテーブル参照がメインであり，更新はあまりないためと考えられる．さらに，Websearch のトレースではキャッシュヒット率が非常に低かった．この理由は，ホスト側のメモリにデータがキャッシュされており，ストレージ側へのアクセスは結果としてランダムアクセスがメインであるためと考えられる．一方で，Financial のトレースでは 20%～33%の割合でリード I/O がキャッシュヒットした．また，Financial のトレースでは実ライト率が 27%～60%であった．

さらに，キャッシュを介した場合のアクセスヒートマップを**図 13**，**図 14** に示す．図 13，図 14 では，縦軸を各ページとし，横軸を時間経過とした各周期（30 分）のアクセス頻度量の変化を示している．1 周期あたり 6000 IO 以上の I/O が発生している領域を赤色で示し，1 IO～6000 IO の I/O が発生している領域を緑色から赤色で段階的に示している．両者を比較すると Websearch のトレースよりも，Financial のトレースのほうが，高負荷ページ（赤色）の負荷変動が大きいことが分かる．

ただし，本トレースはシステム規模が小さいためアクセス対象容量が小さすぎ，シミュレーションの前提としているドライブの構成では十分なシミュレーション結果が得られない．一般に，システムの規模が大きくなった場合にも，アクセスローカリティ（容量に対する負荷分布）が同一であると考えられる．そこで本論文では，本トレースを 8 多重で別々の領域に負荷をかけることとした．その際に，同時に多重でトレースを開始すると，不当にアクセスが重なりレスポンスが増加してしまうため，本シミュレーションでは各トレースを 1 秒ずつずらして開始する．

6.3 シミュレーション結果

本節では，シミュレーション結果について，シミュレーションおよびモデルの精度について考察し，実行ページ選択アルゴリズムの効果，HOL によるコピー処理の効果をそれぞれ示し，それらの全体としての効果を示す．

6.3.1 シミュレーションの有効性

本シミュレーションにおいて，想定した実環境（ストレージシステム）と比較して違いがある要因として，3 点を示し，それぞれの要因における本評価の妥当性を示す．

［要因 1］サービス時間の変動

本シミュレーションの前提では，サービス時間が一定分布で固定時間としている．しかし，実環境における HDD ドライブでは，表 1 に示したように，サービス時間はシーク時間と，回転待ち時間，転送時間の合計により決まる．これらの要素は，ヘッドの状態や I/O 転送長の違いにより一定の範囲でのばらつきが発生する．また，SSD についても，I/O 転送長の違いによりばらつきが発生する．本シミュレーションが前提としている一定分布の場合は，待ち行列式のモデル (3)，(6) で示したように，$\frac{\overline{S_x{}^2}}{S_x} = S_x$ であるため，最もドライブレスポンスが短くなる．サービス時間の分散値が大きいほどドライブレスポンスが大きくなるが，FCFS のケースと HOL のケースのモデル式 (2)，(5) で示したとおり，両方のケースでレスポンスが同じ時間だけ増加する．レスポンスの絶対的な値に差異が出る可能性はあるが，差分としては同等値（式 (2) – 式 (5) $= \frac{(1-\rho_r)S_r}{(1-\rho_h)}$）であると考えられる．したがって，サービス時間の変動は本論文のレスポンス削減効果に本質的な影響を与えるものではないため，本評価は妥当であると考える．

［要因 2］キャッシュの影響

待ち行列シミュレーションは，ドライブ負荷についてのシミュレーションのみを実施している．しかし，一般的

#	トレース名	T1 容量	(1)LBA (HOL)	(2) Prio (HOL)	(3) 最適時 (再配置が即時完了)	比較 (2)/(1)-1
1	Financial1	5GB	51.9%	54.5%	58.6%	5.0% 増加
2		25GB	93.0%	95.9%	97.3%	3.1% 増加
3	Financial2	5GB	52.0%	54.3%	55.4%	4.5% 増加
4		25GB	90.4%	91.8%	92.8%	1.6% 増加
5	Websearch2	60GB	67.2%	68.4%	68.4%	1.8% 増加
6		100GB	90.3%	91.1%	91.1%	0.9% 増加
7	Websearch3	60GB	68.5%	69.1%	69.2%	0.9% 増加
8		100GB	91.0%	92.0%	91.8%	1.1% 増加

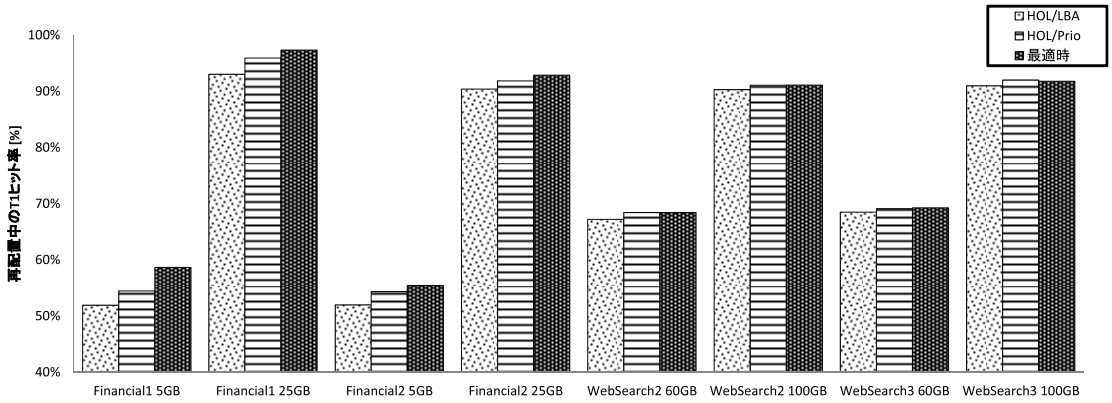

図 15 再配置中のホスト I/O による T1 ヒット率の比較
Fig. 15 Comparison of the T1 hit ratio by host I/O in relocating.

なストレージシステムでは，I/O の高速化のためにキャッシュを持つ．キャッシュを介する場合，ページの I/O 負荷特性が変わる．また，キャッシュヒット時はミス時よりも大幅にホスト I/O レスポンスが小さくなるため，キャッシュヒット率がホスト I/O レスポンスに大きな影響を与える．そこで，キャッシュ動作についてのシミュレーションを実施し，そのトレースを用いて待ち行列シミュレーションを実施することにより，キャッシュの影響を考慮した（6.2 節に記載）．よって，本評価は妥当であると考える．

［要因 3］RAID レベルの影響

実際のストレージシステムでは，データ保護のために RAID1 や 5 等を適用していることが多い．その場合，ホストの Write I/O 時に RAID1 の場合 2 回のドライブ I/O となる．また，RAID5 の場合，4 回のドライブ I/O となる．本シミュレーションでは，RAID0 のため，Write I/O 1 回につき 1 回のドライブ I/O と仮定しているが，RAID1 や 5 の場合，Write 量に応じてドライブ稼働率が全体的に増加することになる．しかし，図 11 に示すように，ドライブ稼働率（ρ_h）が高いほど FCFS に対して HOL の効果は高くなる．したがって，本シミュレーションの結果よりも効果が高くなる可能性は考えられるが，本論文のレスポンス削減効果を否定するものではないため，本評価は妥当であると考える．

6.3.2 モデルの有効性

図 16 に，再配置中のドライブ I/O 平均レスポンスについて，モデルとシミュレーションの比較データを示している．ほとんどのケースでは，90%～140%の範囲でシミュレーションとモデルとの一致を確認できる．ただし，1 ケース（Websearch2 100 GB）について，220%程度の誤差があることが分かる．誤差が発生している理由は，トレースデータのホスト I/O の到着頻度が厳密にはモデルの前提であるポアソン分布となっていないことと，T2 の 8 台のドライブの負荷に偏りが発生しているためである．つまり，平均稼働率よりも，実際のドライブ稼働率が高くなっている時間帯にレスポンスが高くなり，全体の平均レスポンスとしてモデルよりも高くなる場合がある．したがって，図 16 に示すように，ホスト I/O によるドライブ平均稼働率が高い場合に，誤差が発生しやすい傾向がある．ただし，一般的なシステムではホスト I/O の平均稼働率を低く抑えることが多いと考えられるため，多くのケースでは有用であると考えられる．

6.3.3 実行ページ選択アルゴリズムの効果

ページの移動順序を特に工夫しない単純な実装方法として，ボリュームの論理アドレスの順番に Demotion/Promotion ページを選択する方法（以降 LBA 方式と呼ぶ）が考えられる．本項では，この論理アドレス順の方法と，本論文で述べたページの I/O 頻度によって最適化した実行ページ選択アルゴリズム（以降 PRIO 方式と呼ぶ）の 2 個の方式の比較を実施する．

図 15 に再配置中のホスト I/O による平均 T1 ヒット率の比較を示す．ページコピー方式は HOL を前提とする．

図 15 に示すとおり，本論文の PRIO 方式が最大で 5%程度，T1 にヒットさせることができ，優位であることが分かった．本論文が前提としているシステムでは，ホストからのアクセス範囲に対して小容量の SSD を用いることを前提としている．また図 4 に示したように，負荷のモニタ

#	トレース名	T1 容量	再配置中のドライブ I/O 平均レスポンス（平均再配置時間） (1) FCFS	(2) HOL	比較 1−(2)/(1)	モデルとの比較 (1) FCFS (モデル)	(2) HOL (モデル)
1	Financial1	5GB	5.90ms (14s)	5.45ms (14s)	7.6% 削減	5.74ms (102.8%)	4.01ms (135.9%)
2		25GB	2.14ms (47s)	1.96ms (47s)	8.4% 削減	2.07ms (103.5%)	1.68ms (116.4%)
3	Financial2	5GB	6.66ms (18s)	4.79ms (18s)	28.1% 削減	6.12ms (108.8%)	4.08ms (117.3%)
4		25GB	2.32ms (51s)	2.04ms (51s)	12.4% 削減	2.48ms (93.7%)	1.90ms (107.2%)
5	Websearch2	60GB	11.91ms (161s)	8.24ms (177s)	30.8% 削減	8.42ms (141.4%)	3.76ms (219.2%)
6		100GB	4.11ms (127s)	2.15ms (128s)	47.6% 削減	4.26ms (96.4%)	2.01ms (107.3%)
7	Websearch3	60GB	6.04ms (115s)	4.28ms (119s)	29.0% 削減	6.17ms (97.8%)	3.45ms (124.2%)
8		100GB	3.19ms (113s)	2.02ms (113s)	36.4% 削減	3.27ms (97.4%)	1.92ms (105.4%)

#	トレース名	T1 容量	再配置中の T1 平均稼働率 (1) FCFS	(2) HOL	再配置中の T2 平均稼働率 (3) FCFS	(4) HOL	再配置中のホスト I/O T1 平均稼働率 (1) FCFS	(2) HOL	再配置中のホスト I/O T2 平均稼働率 (3) FCFS	(4) HOL
1	Financial1	5GB	94.40%	94.10%	47.13%	47.04%	3.78%	3.80%	10.48%	10.53%
2		25GB	95.86%	95.84%	36.79%	36.79%	7.55%	7.55%	1.07%	1.07%
3	Financial2	5GB	92.15%	91.67%	50.43%	50.24%	5.51%	5.33%	15.41%	15.41%
4		25GB	96.06%	96.02%	37.73%	37.72%	10.20%	10.20%	3.02%	3.02%
5	Websearch2	60GB	85.91%	81.18%	71.45%	70.06%	32.95%	33.00%	50.04%	50.60%
6		100GB	98.63%	98.40%	36.52%	36.44%	42.65%	42.65%	13.95%	13.89%
7	Websearch3	60GB	93.13%	90.92%	61.54%	60.56%	22.11%	22.04%	32.82%	32.71%
8		100GB	97.68%	97.54%	36.15%	36.08%	28.98%	28.99%	8.36%	8.36%

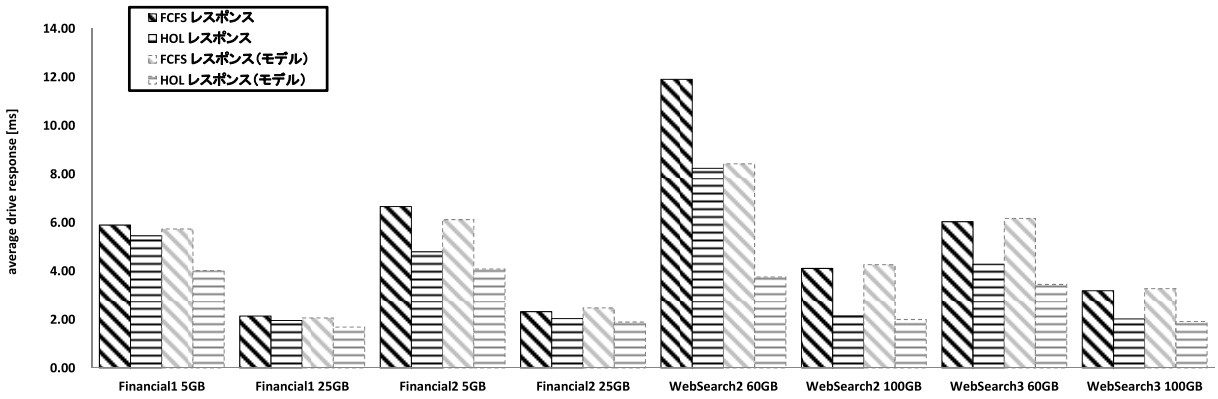

図 16 再配置中のドライブ I/O 平均レスポンスの比較
Fig. 16 Comparison of a drive I/O average response in relocating.

リングと再配置を周期的に実行しており，モニタリングの結果として，再配置を実行している．したがって負荷が変化した後，モニタリングが完了して再配置が始まるまでは，T2 にアクセスされる．以上の理由から，T1 に 100%ヒットさせることは現実的には困難であるため，ページ再配置が周期の開始時にペナルティなしで，ただちに完了する場合を，提案手法が最適な場合（図 15 の (3)）と定義した．この最適時と比較して，Websearch のトレースについては，本提案手法（図 15 の (2)）は，最適時とほぼ一致した．#8 の WebSearch3 100 GB のケースで最適時のほうがわずかに（0.2%）本提案手法よりも低くなっているが，モニタリングによる予測が稀に外れるケースがあるためである（モニタリングの的中率については本論文では扱わないが，今後の課題である）．Financial トレースについては，LBA 方式と最適時の中間程度のヒット率に近づけることができた．以上の効果の違いの理由は，負荷変動して移動対象となったページ内での負荷の偏りの違いと考えられる．負荷が偏っているほうが PRIO 方式の効果が高くなる．

6.3.4 HOL によるコピー処理の効果

本項では，単純な実装方式として，再配置の I/O をホスト I/O を同一の FIFO キューにおいて 1 多重で処理するコピー処理（FCFS）と，本論文で述べたホスト I/O 用の優先キューを持つコピー処理（HOL）を比較する．図 16 に再配置中のホスト I/O レスポンスと再配置時間の比較を示す．

FCFS と HOL ともに，再配置中のネックとなる T1 平均稼働率を 80%以上引き出しているため，再配置時間はそれほど差異がない．一方で再配置中のホスト I/O 平均レスポンスは，HOL のほうが FCFS と比較して 7%〜47%削減された．図 11 に示すように，ドライブ稼働率が高ければ高いほど FCFS に対して HOL の効果は高くなる．図 16 に示すように，Financial よりも Websearch のほうが HOL によるドライブ I/O 平均レスポンスの削減効果が高くなっている．これは，再配置中のホスト I/O によるドライブ稼働率が Websearch のほうが高いためである．

6.3.5 全体としての効果

全体のホスト I/O レスポンスのシミュレーション結果を**図 17** に示す．本結果は，シミュレーションで得られたドライブレスポンスと，表 4 に示したキャッシュヒット率から，キャッシュヒット時のレスポンスを 0 ms として算出した．シミュレーションで得られたドライブレスポンスは，初回の 1 周期目についてはモニタリングデータがなく再配

#	トレース名	T1 容量	(1) 再配置無	(2) 再配置有 (FCFS/LBA)	(3) 再配置有 (HOL/Prio)	(4) 最適時	比較 1-(3)/(2)	比較 1-(3)/(1)
1	Financial1	5GB	4.20ms	0.63ms	0.63ms	0.63ms	0.1% 削減	85.0% 削減
2		25GB	2.18ms	0.15ms	0.15ms	0.14ms	2.3% 削減	93.2% 削減
3	Financial2	5GB	12.08ms	2.60ms	2.58ms	2.56ms	0.8% 削減	78.7% 削減
4		25GB	2.55ms	0.57ms	0.56ms	0.53ms	2.2% 削減	78.0% 削減
5	Websearch2	60GB	26.88ms	6.87ms	6.23ms	5.67ms	9.4% 削減	76.8% 削減
6		100GB	1.22ms	1.10ms	0.95ms	0.83ms	13.7% 削減	22.4% 削減
7	Websearch3	60GB	4.40ms	2.79ms	2.63ms	2.48ms	5.8% 削減	40.3% 削減
8		100GB	1.06ms	0.92ms	0.83ms	0.73ms	9.6% 削減	21.4% 削減

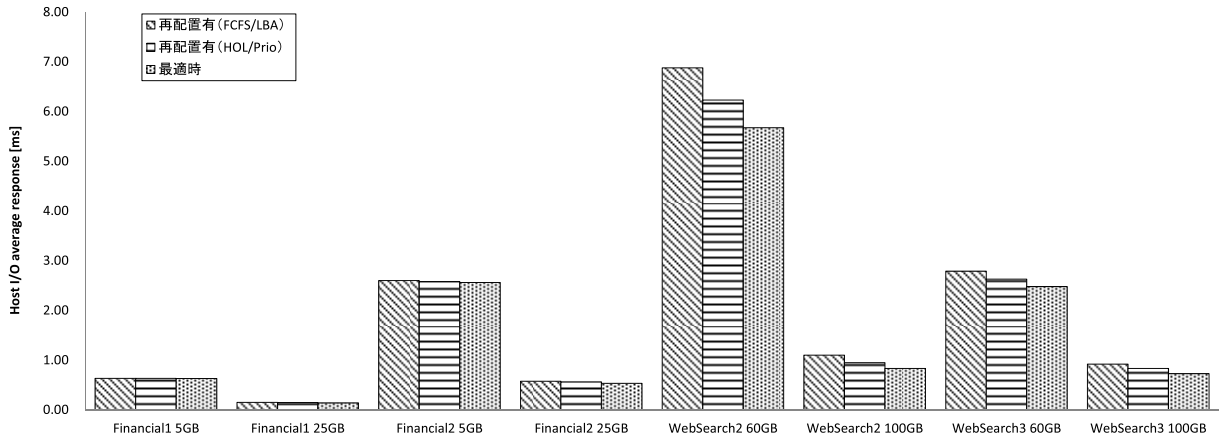

図 17 ホスト I/O 平均レスポンス比較グラフ
Fig. 17 Comparison of a host I/O average response.

置が動作していないため，2 周期目以降（30 分経過後）のドライブレスポンスの平均値を用いた．本論文の方式は，再配置を実行しない場合（初回ライト I/O の到着順に T1 に配置したまま動かさない）と比較し，21%～93%のホスト I/O 平均レスポンス削減効果を確認した．また，単純な再配置方式（ボリュームのアドレス順にページ移動し，FCFS スケジューリングでデータコピーを実行）と比較し，最大で 13%のホスト I/O 平均レスポンス削減効果を確認した．また，本結果を見ると，再配置の時間，つまり移動対象のページ数が多いほど全体の効果が高くなることが分かる．アクセス対象の容量が大きいほど，移動対象のページ数が多くなる傾向にある．図 17 に示した，ホスト I/O 平均レスポンス削減効果が Websearch のほうが高い傾向にあるのは，図 16 に示した，再配置時間（1 周期あたりの平均時間）が Websearch のほうが長いためと考えられる．

7. まとめ

本論文において，ストレージ自動階層配置機能において，データ再配置における実行ページ選択の順序最適化と，コピー処理の I/O スケジューリング方式を検討した．

（1）実行ページ選択の順序最適化
階層の容量を考慮しつつ，アクセス頻度の高いデータを優先して移動するアルゴリズムを提案し，一定の条件下における最適性を証明した．

（2）コピー処理の I/O スケジューリング方式の検討
データ再配置におけるコピー処理を効率化するべく，コピー I/O スケジューリングの方式として，単純な FCFS 方式と，優先度を考慮した HOL 方式を比較した．両者のホスト I/O レスポンスの振舞いを待ち行列による解析モデルで示し，一定の条件下において，HOL 方式が優位な方式であることを示した．

実環境 I/O トレース（4 種類）を使った待ち行列シミュレーションの検証の結果，上記のデータ再配置方式により，単純な再配置方式と比較し再配置中のレスポンス性能を 7%～47%削減でき，全体として最大で 13%のホスト I/O 平均レスポンス時間の削減効果を確認した．

今後の課題は，より T1 へのヒット率を高めるため，モニタリング手法や，再配置の周期実行を工夫することである．

参考文献

[1] Dasgupta, K., Ghosal, S., Jain, R., Sharma, U. and Verma, A.: QoSMig: Adaptive rate-controlled migration of bulk data in storage systems, *Proc. 21st Int'l Conf. on Data Engineering*, pp.816–827, IEEE (2005).
[2] Verma, A., Sharma, U., Jain, R. and Dasgupta, K.: Compass: optimizing the migration cost vs. application performance tradeoff, *Proc. IEEE Trans. Network and Service Management*, Vol.5, No.2, pp.118–131 (2008).
[3] Zhang, G., Chiu, L. and Liu, L.: Adaptive Data Migration in Multi-tiered Storage Based Cloud Environment, *Proc. IEEE CLOUD 2010*, pp.148–155 (2010).
[4] Guerra, J., Pucha, H., Glider, J., Belluomini, W. and Rangaswami, R.: Cost Effective Storage using Extent Based Dynamic Tiering, *Proc. FAST11* (2011).
[5] 山本 彰：割り込み許可点を有する待ち行列モデルとディスク装置の中断制御方式への適用，電子情報通信学会論文誌，Vol.J82-S-I, No.4 (1999).
[6] Kleinrock, L.: Queueing Systems, Vol.11, Hohn Wiely

(1976).
[7] Bates, K. and McNutt, B.: Storage – Umass Trace Repository, available from ⟨http://traces.cs.umass.edu/index.php/Storage/Storage⟩ (accessed 2012-04-01).
[8] HGST: Ultrastar, available from ⟨http://www.hitachigst.com/portal/site/jp/products/ultrastar/⟩ (accessed 2012-04-01).
[9] 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，FIT2008，C-019 (2008).
[10] Desnoyers, P.: Empirical Evaluation of NAND Flash Memory Performance, *Proc. ACM SIGOPS Operating Systems Review*, pp.50–54 (2010).
[11] 坏 弘明，今崎美保，須藤 梓，大平良徳，江口賢哲：ストレージ階層仮想化機能における高速な移動先階層の判定方法，FIT2011, D-007 (2011).
[12] 山本 彰，坪井俊明，北嶋弘行：連続転送方式に基づくカートリッジ型 MT の先読み/まとめ書きスケジューリングアルゴリズムとその性能解析，情報処理学会論文誌，Vol.30, No.1 (1989).
[13] IBM: IBM System Storage DS8000 シリーズ，available from ⟨http://www-06.ibm.com/systems/jp/storage/products/disk/ds8000/pdf/ds8000.pdf⟩ (accessed 2012-11-26).

坏 弘明 （正会員）

平成 15 年早稲田大学理工学部情報工学科卒業．平成 17 年同大学大学院修士課程修了．同年日立製作所入社．同年システム開発研究所（現：横浜研究所）に入所．以来，横浜研究所にてストレージシステムの研究に従事．

山本 彰 （正会員）

昭和 52 年京都大学工学部情報工学科卒業．昭和 54 年同大学大学院修士課程修了．同年日立製作所入社．同年システム開発研究所（現：横浜研究所）に入所．性能評価手法，ストレージシステムの研究に従事．平成 19 年より同社研究開発本部所属．

大平 良徳

平成 16 年東京工業大学工学部情報工学科卒業．平成 18 年同大学大学院修士課程修了．同年日立製作所入社．同年システム開発研究所（現：横浜研究所）に入所．以来，横浜研究所にてストレージシステムの研究に従事．

江口 賢哲

平成 5 年九州大学理学部物理学科卒業．平成 7 年同大学大学院修士課程修了．同年日立製作所入社．同年システム開発研究所（現：横浜研究所）に入所．横浜研究所にてストレージシステムの研究に従事．平成 24 年より同社 IT プラットフォーム事業本部所属．