並列ストレージにおけるデータ再配置による長期的負荷均衡化と短期的応答性能の両立

小林 大[†1,†2] 横田治夫[†1]

並列ストレージシステムにおける性能維持では，アクセス負荷の時間的変化に対するストレージノード間のアクセス負荷均衡化を迅速に行うことが肝要である．しかし動的負荷均衡化をデータマイグレーションによるデータ再配置によって行う手法は，システム処理能力の一部を使用するため短期的に応答性能保証が困難となる．本論文では，データマイグレーションによる負荷均衡化と応答性能維持の両立を目指す．提案する手法では，他のノード中に格納される複製データを用いたデータ移動経路選択およびアクセス回送を，各ノードの許容最大負荷を基準として制御する．これにより応答性能要件を維持しつつ迅速にマイグレーションを行う処理能力を確保する．シミュレーションによる実験により，読み出しアクセス中心のワークロード下における提案手法の有効性を検証した．その結果，提案手法を用いることで，単純な人工的ワークロード下では 33%〜38%の負荷耐性向上，実利用を想定したワークロード下でもマイグレーションに起因する超過レスポンスの 93%削減を達成し，応答性能を維持しつつ迅速なマイグレーションが可能となることが示された．

Simultaneous Pursuit of Load Balancing by Data Migration and Quality of Response Performance on Parallel Storage Systems

Dai Kobayashi[†1,†2] and Haruo Yokota[†1]

Load balancing between storage nodes is important to keep performance of parallel storage systems above required level. However, online load balancing with data migration causes a temporary quality violation of system services. To solve the problem, we propose a method using replicated data to keep the service quality even during the data migration. It forwards overloaded accesses to the replica data, and selects an appropriate source node of the migration from its original or replica to moderate the load and to shorten the migration process. We show its efficiency with simulation results.

1. はじめに

爆発的に増大するデータを可用性，柔軟性，拡張性といった技術項目を満たしつつ低い管理コストで格納するストレージシステムの実現が求められている．このため，HDD に加え CPU やメモリ等を組み込んだ高機能なストレージノードを LAN により多数結合したシステムである高機能並列ストレージシステムが注目されている[1)–3)]．

管理コスト増加の一因として負荷均衡化があげられる．システムを構成する一部のストレージノードへのアクセス集中はストレージノードのレスポンスタイムを著しく増加させる[4)]．そのため，ノード間での大きなアクセス負荷偏りは負荷均衡化により除去すべきである．データマイグレーションは並列構成のデータベースやストレージシステムにおけるアクセス負荷均衡化手法である．データマイグレーションでは，ネットワーク結合されたストレージノード間で各データのアクセス負荷傾向を考慮しデータを移動することでデータ配置を変化させる．これにより，一部ノードへのアクセス集中を回避し，システムに要請されるサービス品質を維持することができる．高機能並列ストレージシステムではストレージノード上の計算資源を用いてストレージシステムの自律管理を行うことが可能である．動的に採取される現在の負荷情報をもとにシステムが自律的に行う動的データマイグレーションは，セルフパフォーマンスチューニング手法として提案され効果を得ている[5),6)]．

しかし，動的データマイグレーションによるアクセ

†1 東京工業大学
Tokyo Institute of Technology
†2 日本学術振興会特別研究員 DC
Research fellow, Japan Society for the Promotion of Science

ス負荷均衡化は，短期的にはノード性能を低下させることが問題となる．データマイグレーションがディスクアクセスやネットワーク転送等，ストレージノードが通常のデータ提供サービスに用いる資源の一部を利用するためである．アクセス傾向の変化が激しい場合や，負荷の評価精度が低い場合，処理能力がすでに飽和しているストレージノードからのデータ転送が求められる．その結果，ユーザにより要求される応答性能が満たされなくなる可能性がある．

データマイグレーション中の応答性能維持に関する既存研究として，速度制御がある[7)–9)]．これらの既存研究はいずれも，主に階層構成のストレージシステムにおける階層間のデータ移動を目的としている．上記の既存研究はいずれも，階層構成におけるデータ移動は使われていないデータに対する移動であり，データマイグレーションの速度を遅らせてもシステムレスポンスタイムやスループットへの影響が小さいことを仮定し，マイグレーション速度を調節するアプローチをとっている．一方，階層構成におけるデータマイグレーションと異なり，負荷均衡化のためのデータマイグレーションは迅速な実行が必要とされる．データ移動元，移動先がともにサービス提供中であり，かつ大きな負荷のアクセス下であることが多いためである．速度調整による応答性能保証は負荷均衡化のためのマイグレーションでは難しい．

このような問題を解決するため，本論文では，Replica-assisted Migration（以下，RM 手法）を提案する．RM 手法により，マイグレーション速度を減少させることなく，マイグレーション中の応答性能維持を達成する．RM 手法では複製データを用いる．並列ストレージシステムでは，ノード故障によるデータ喪失を防ぐため，他のノードのバックアップのため複製データを保持する構成をとるものが多い．RM 手法では，他のノードに配置された複製データを利用することで負荷集中ノードから一時的に処理能力を捻出する．よって，ノードに負荷が集中した後でも，データマイグレーションを行うことが可能となる．我々の先行研究[10)]では，複製データへのアクセス回送による応答性能維持効果について論じた．本論文の提案では，複製データを用いたマイグレーション経路選択，および複製へのアクセス回送制御を行い，処理能力を確保する．提案する RM 手法では，この経路選択，アクセス回送の 2 種類を，ストレージノードの許容最大性能を基に制御する．RM 手法によって，負荷集中ノード，マイグレーション対象ノード，そしてそれらの複製保持ノードそれぞれが性能要件を維持可能な許容最大負荷以下の負荷で動作可能となることを目指す．なお，本論文においてはデータマイグレーション発生時の特殊な状況のみ複製の利用を考慮し，つねにアクセスを複製へ回送する負荷均衡化手法を併用するアクセス回送常用システムへの拡張は今後の課題とする．

本論文の後半では，並列ストレージシミュレーションプログラムによる実験により，提案手法の有効性を確認する．実験では，Zipf 分布に基づく読み出しリクエスト中心の人工的な負荷，および大規模 WEB サーバから採取されたアクセス傾向に基づくワークロードを用いる．それぞれにおいて応答性能，マイグレーション時間の観点から評価を行う．その結果，提案手法を用いることで，単純な人工的ワークロード下では 33%〜38%の負荷耐性向上，WEB ワークロード下でもマイグレーションに起因する超過レスポンスの 93%削減を達成し，応答性能を維持しつつ迅速なマイグレーションが可能となることが示された．

本論文の構成を以下に示す．つづく 2 章では，データマイグレーション，応答性能保証，複製の利用に関する関連研究について論ずる．3 章において前提とするシステムおよびデータマイグレーション戦略について述べる．4 章では RM 手法で用いる 2 種の複製利用法を紹介し，5 章において，提案する RM 手法を構成する複製制御アルゴリズムについて述べる．6 章において，本論文における実験環境の概要を述べる．7 章では人工的なワークロード，8 章では実利用を想定したワークロードを用いて，提案手法の有効性をシミュレーションにより確認する．最後に 9 章でまとめと今後の課題について述べる．

2. 関 連 研 究

アクセス負荷均衡化は主に無共有並列データベースの分野における主要な課題であった．Scheuermann ら[11)]は，分割されたテーブルを格納するシステムにおいて，各データ断片のアクセス負荷をもとにした動的データマイグレーションによる負荷均衡化を達成した．そして彼らは動的マイグレーションと性能保証の両立が次の重要な課題であると述べていた[5)]．また，Feelifl らは並列 B-tree 索引構造の負荷分散をデータマイグレーションにより解決している[6)]．このとき彼らは速度制御による性能保証を導入している．これはデータ量が比較的少ない索引構造のマイグレーションとして効果が得られている．データマイグレーションと応答性能保証両立に対する速度制御によるアプローチは階層構成ストレージシステムにおいて多く提案されている．階層構成では上位層と下位層の間でデータ

を移動する必要がある．これまでに，現在のレスポンスタイムを基にした適応的速度制御[8]，負荷変化の予想による速度制御[7]，ストレージノードの許容最大性能見積り値を基にした速度制御[9] 等が提案されている．しかし，負荷均衡化のためのデータマイグレーションでは速度制御の効果が低い．これは本論文における実験で明らかにする．

自律データ管理と応答性能保証の問題は，ストレージアレイにおける障害回復時においても問題となっている．ディスク障害にともなう突発的な負荷不均衡や，データ冗長度回復のためのアクセスが応答性能を低下させるためである．Chained Declustering[12] は，前者の突発的な不均衡を複製データへのアクセス回送により解決している．一方，Façade[13] は突発的な不均衡や冗長度回復と応答性能保証を，クライアントとストレージの間のフローコントロールにより解決している．

複製データを障害回復だけでなくアクセスにも利用する手法は，WAN における長いネットワークレイテンシの隠蔽[14)–16)]，ビデオ配信サーバの負荷均衡化[17)] 等で長く利用されてきた．しかし，高機能並列ストレージにおいては，単純なアクセス回送は各ストレージノード上のキャッシュメモリ利用率を低下させるという問題がある[18),19)]．一方データマイグレーションによるアクセスはキャッシュヒット率が低いアクセスである．よって，キャッシュミスアクセスを優先的に回送する手法[19)] により，マイグレーションによる負荷上昇は他ノードに回送できる．

3. 前提とするシステムおよびデータマイグレーション

3.1 システム

本論文では，Gigabit Ethernet 等の，高速な LAN 上に並列接続された高機能ストレージノードにより構成されたシステムを想定する．データは，ファイル，エクステント，ページ，ブロック等の粒度で分割を行い，各ストレージノードに分散格納する．分割されたデータ断片と格納されるストレージノードの対応は，メタデータサーバや分散ディレクトリを利用したストレージ仮想化機構により利用者から隠蔽される．そのため，システムは運用中の動的データ再配置を低い管理コストで行うことが可能となる．また，各ノードはアクセスのためのデータ断片（プライマリデータ）に加え，他のある 1 つのノードに格納されたプライマリデータの複製データを障害復旧用に保持する．

データの読み出しは基本的にプライマリデータから

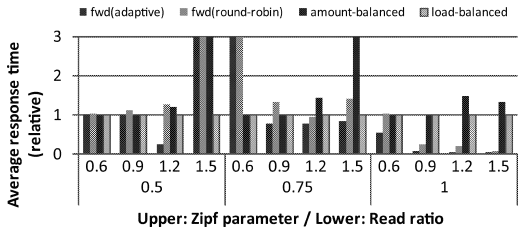

図 1 アクセス回送と応答性能（データ配置による負荷均衡時の値を 1 とした相対応答性能）

Fig. 1 Access forwarding and response time (Normarized based on load balanced with suitable data placement).

行うことを仮定する．データの更新はプライマリと同時にすべての複製データにも同期的に行われる．以下では簡単な実験により，複製間ラウンドロビンによるアクセス回送常用，負荷が少ないノード上の複製へアクセスを回送する適応的アクセス回送常用，アクセス回送なし（データ配置は負荷均衡化），アクセス回送なし（データ配置は格納量均衡）の 4 種の平均応答性能を測定・比較する．図中ではそれぞれ，fwd(round-robin)，fwd(adaptive)，load-balanced，amount-balanced と表記する．

6 章で述べるシステムでノード 4 台，後述の Zipf 分布，ポアソン到着で到着率 350 [req/sec]，傾向変化なしのワークロードを発生させた．また，キャッシュミス時の性能ペナルティを大きくするため，SYNC 書き込みを 25%，50%混ぜたアクセスでも実験を行った．

図 1 に，データ配置格納量均衡時の値を 1 とした相対応答性能を示す．図より，アクセス回送により著しく応答性能が改善される例がある反面，ラウンドロビンや適応的回送においても応答性能が悪化する例も見られる．たとえば，Read 比率 0.75，Zipf 係数 0.6 の例では，ノード間アクセス量の標本分散が適応的回送により 1/4 倍に改善されているにもかかわらず，キャッシュヒット率が 37%から 31%へ低下したため許容負荷量が低下し，応答性能が悪化している．

このため，本論文においてはデータマイグレーション発生時の特殊な状況のみ複製の利用を考慮し，アクセス回送常用への拡張は今後の課題とする．

3.2 データマイグレーション

時間的に変化するワークロード下でシステム全体の性能を維持するため，システムはデータ動的再配置（データマイグレーション）による負荷均衡化を自律的に行う．データマイグレーションは，各ノードごとのアクセス負荷が要求性能を維持可能な量を超えないデータ配置を目標とする．本論文では集約評価型の負荷均衡化アルゴリズムを前提とする．すべてのノード

は，自ノードに格納されるデータの現在の負荷を表す負荷値を記録している．ここで負荷とはデータ提供のための資源利用量を示す度合いである．READ によるプライマリデータアクセス，WRITE によるプライマリ，バックアップ両データに対するアクセスの量で決まる．負荷値は負荷を定量的に表現する値である．例として，最近一定時間のアクセス量の平均値や，最近一定回数のアクセスの到達時間間隔，あるいはそれらの値を各ノードの性能を表す値によって正規化した値が負荷値として利用される．

集約評価型の負荷均衡化では，システムはある定められたインターバル時間ごとに以下の作業を行う．まず全ノードは，ある 1 つのノード（以下では coordinator と呼ぶ）を選定する．そして coordinator に対し直接，あるいは段階的[20] に自ノードの負荷値を送信する．全ノードの負荷値を受け取った coordinator は，すべてのノード間で負荷値が均衡化するデータ配置とそのためのデータ移動戦略を計算する．移動戦略はマイグレーションタスクの集合とする．ここでマイグレーションタスクは（移動元ノード，移動先ノード，移動負荷値）とする．そして各ノードに移動戦略を送信する．

coordinator では集約負荷値から移動する負荷値のみを決定し，具体的な移動データの特定は各ノード内で行う．これは個々の格納データごとの負荷情報を coordinator ノードへ集約することはコストが高いためである．各ノードは，与えられた戦略に含まれる移動負荷値がある閾値を越えた場合，自ノードの個々の格納データに関する負荷値を走査し，与えられた移動負荷値を満たすデータセットを特定する．各ノードは指定された移動先ノードに必要なデータを送信する．

ここで，データの移動は Silvering[8] と呼ばれる操作により行われることを前提する．Silvering ではまず移動するデータの複製を移動先ノードに作成する．次に複製作成開始時から終了時までの更新を複製に適用する．最後に元のデータを削除する．これにより移動中データへの排他ロック取得期間を短く抑えることができる．よって，以下の議論ではデータマイグレーションのための排他制御による応答性能への影響は考慮しないこととする．

4. 複製データの併用

本章ではまず RM 手法で用いる複製の 2 つの利用法である，マイグレーション経路選択とアクセス回送制御についてそれぞれ説明を行う．続いて次章において，2 つの利用法を制御しデータマイグレーション中

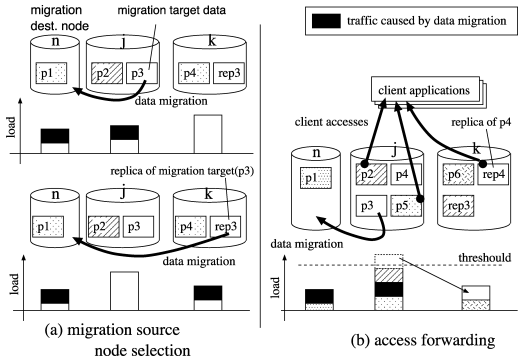

図 2 RM 手法における 2 種類の複製利用法．(a) マイグレーション経路選択，(b) アクセス回送
Fig. 2 Relica usage in RM method. (a) Migration source node selection, (b) access forwarding.

の応答性能維持を達成する RM 手法のアルゴリズムを紹介する．

マイグレーション経路選択では，移動対象データの複製を保持するノードのうち最も負荷の低いノードをデータ移動元ノードとして選択する．**図 2** (a) にその様子を示す．データ p3 をノード n へ移動する．データ移動元ノード j の負荷が複製保持ノード k よりも低ければ，ノード j のプライマリデータ（p3）を移動する．そうでなければ，ノード k 上に格納された p3 の複製データ（rep3）を移動する．

アクセス回送では，あるノードに格納されたデータへのアクセスの一部を，当該データの複製データを保持するノードに回送する．データマイグレーション対象ノードが負荷集中ノードとなった場合，このアクセス回送を用いマイグレーション対象ノード上にマイグレーション処理能力を確保することができる．アクセス回送は各ノード上のキャッシュヒット率の低下を招く可能性がある．アクセス頻度の高い少数の同じデータが複数のノード上のキャッシュに格納されるためである．そこで，我々が以前に提案した，キャッシュヒット・ミスを観測しつつ回送対象アクセスリクエストを選出する手法[19] により，この低下を抑えることとする．図 2 (b) では，ノード j 上のデータ p3 をノード n へ移動する．ノード j の処理能力がデータマイグレーションにより飽和するため，データ p4 へのアクセスをノード k 上にある複製データ rep4 へ回送している．

アクセス回送は次のように行う．あらかじめ，各ノード i ごとにアクセス回送率 r_i を算出する．回送率 r_i は $0 \leq r_i \leq 1$ であり，そのノード上データに要求される全アクセス量（アクセス数とアクセスサイズの積）のうち，複製側へ回送するアクセス量の割合を示す．

ノード j が移動元の場合，移動元ノード j に対する

アクセス要求を，与えられた回送率 r_j の割合で，複製の存在するノード k へと回送するようにノード j 上のコントローラをセットする．つづいてマイグレーション戦略に基づきデータを移動する．その間のノード j へのアクセス要求の一部はセットされた回送率 r_j とアクセス回送判定手法[19] に基づき複製ノード k へ回送される．データ移動完了後，複製への要求回送率 r_j を 0 にする．

5. Replica-assisted Migration

データマイグレーションによる応答性能の低下は，データマイグレーション処理を行うストレージノードのアクセス処理能力飽和により起こる．クライアントへの応答性能を維持しつつマイグレーションへも処理能力を確保するため，マイグレーション経路の選択とアクセス回送率の計算が重要となる．ラウンドロビン等の単純な戦略では，複製利用により負荷集中ノードの負荷が複製保持ノードに移譲されることで，複製保持ノードの許容最大負荷を超えてしまうためである．加えて，単純な負荷分散やスループット向上ではなく，各ノードの性能要件保持を考慮する必要がある．

本章では，5.2 節で述べる我々が以前から提案している自律ディスク[1] のデータ配置を前提に，マイグレーション経路選択とアクセス回送を制御する，RM 手法を提案する．移動元ノードの負荷が最大許容量に近い場合でも，複製データ保持ノードの処理能力を用い，データマイグレーションによる負荷均衡化を行うことが可能となる．さらにマイグレーション用処理能力の確保も重視するためマイグレーション速度低下が少ない．

RM 手法では性能要件として与えられるレスポンスタイムを基に集約評価型のアルゴリズムにより coordinator ノード上で，2 つの複製データ利用戦略を決定する．これは，負荷集中ノード，マイグレーション対象ノード，そしてそれらの複製保持ノードそれぞれが性能要件を維持可能な許容最大負荷以下の負荷で動作可能となる戦略を得るためである．実際の制御アルゴリズム構築・記述については，データ配置，複製配置の制約による影響が大きい．そこで本章では，まず RM 手法による複製制御の概要を説明する．つづいて自律ディスクへの適用を考慮し，アルゴリズムを構築する．

5.1 概　　要

RM 手法では，まずマイグレーション経路の選択によりマイグレーションタスクの移動元をより負荷の低いノードに割り当てる．その後，各ノードの許容最大負荷値を維持するためのアクセス回送率の計算を行う．

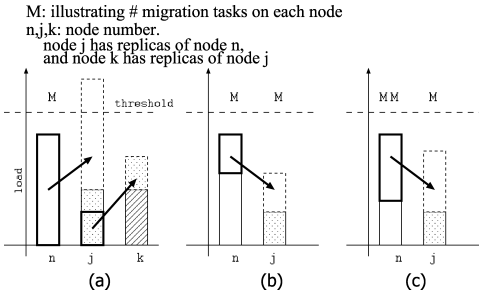

図 3 アクセス回送率計算アルゴリズムの概要
Fig. 3 Concepts of algorithm for calcuration of access forwarding ratio.

各データに対する性能要件は主にアクセスに対するレスポンスタイムで表される．ストレージノードのレスポンスタイムは負荷の増加に対して巨視的には単調増加する[4]．よって，レスポンスタイムとしての性能要件は各ノードの許容最大負荷値へ変換できる．許容最大負荷値は，ストレージノードのディスク，ネットワーク等の性能緒元，用途における不変のアクセス傾向（キャッシュヒット率，アクセスサイズ分布等）を考慮のうえ，事前に各ノードの性能測定を行った結果を用いて算出する．

提案手法では 3 章で述べた，coordinator によるマイグレーション戦略立案後，得られた戦略を利用して次のことを行う．まずマイグレーション経路選択アルゴリズムにより，複製保持ノードのうちより負荷の低いノードにマイグレーションタスクの移動元ノードを割り当てる．その結果として，各ノード i における割当てマイグレーションタスク数 s_i（移動先ノードまたは移動元ノードとなったタスクの個数）が得られる．マイグレーション経路選択アルゴリズムでは，まずすべてのマイグレーションタスクからデータ移動先ノードのリストを作成し，それをもとに各ノードの負荷を再計算する．そして戦略から得られるマイグレーションタスク 1 つごとに，そのデータ移動元ノードを最も負荷が低い複製ノードに割り当てる．

つづいて提案手法のうち回送率計算アルゴリズムにより，アクセス回送率を算出する．その結果として各ノード i におけるアクセス回送率 r_i が得られる．回送率計算アルゴリズムでは，クライアントへのアクセスへの処理能力を確保しつつ，マイグレーション対象タスクにより多くの処理能力を確保する．**図 3** にアクセス回送率計算アルゴリズムの概要を示す．ここで，図中 M はマイグレーション経路選択アルゴリズムによりデータ移動元もしくはデータ移動先として割り当てら

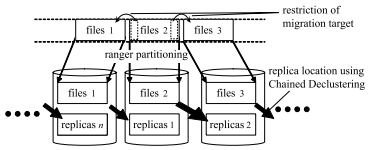

図 4 Chained Declustering と値域分割データ配置を適用したストレージシステムの例
Fig. 4 Example of Chained Declustering and range partitioning.

れたタスク数を表す．マイグレーション対象ノード n の複製保持ノード j へタスクが割り当てられていない場合（図 3 (a)），ノード n へマイグレーション用処理能力を確保するため，ノード n のすべての読み出しアクセスを j へ回送する．ここでノード j の許容最大負荷値を超えた場合，超過分に相当する j へのアクセスをさらにその複製保持ノード k へと回送する．ノード n，j ともに 1 つのマイグレーションタスクの対象の場合（図 3 (b)），両ノードで空き負荷値（許容最大負荷値とノード負荷値の差）が均等になるように回送負荷値を決定する．ノード間で異なる数のマイグレーションタスクが割り当てられていた場合（図 3 (c)），空き負荷値の比とタスク数の比が一致するよう回送負荷値を決定する．以上を各ノードに対して行い，現在負荷値に対する回送負荷値の割合を回送率 r_i とする．

5.2 自律ディスク

我々はこれまで可用性やスケーラビリティに優れた高機能並列ストレージシステムとして自律ディスクを提案している．自律ディスクでは，Primary-Backup の One-copy [21] 複製により障害復旧用バックアップ用複製データを管理する．このバックアップ用複製データを，Chained Declustering [12] 複製配置戦略に基づきすべてのストレージノード間で鎖状に保持しあうことで信頼性を高めている．

また，データ配置管理に分散 B$^+$-tree による分散仮想化機構を有する．各ノードは B$^+$-tree 索引構造のある値域の部分木を保持し，データ位置検索に利用する．そのため格納されるデータ断片についても，一意に定められた識別子を与え，対応する索引構造と同じノードへ格納する，値域分割を採用している．このような値域分割と複製配置戦略に対する制約から，あるノード n からのマイグレーションは，n に与えられた値域において最大のデータ識別子を持つデータをデータ識別子値正方向（$n+1$），もしくは最小識別子のデータを同負方向（$n-1$）へ移動する 2 方向のマイグレーションのみとなる（**図 4**）．

5.3 アルゴリズム

本節では，自律ディスクのデータ配置制約上で 5.1 節で述べた RM 手法を実現するためのアルゴリズムを記述する．

以下では各ノード i の複製データはノード $(i+1)$ に保持されているとする．また明記のない場合ノード番号 i に対する加減算 $i \pm p$ は $i \pm p \bmod N$ とする．ただし N は全ノード数である．ここで $L_i(t)$ は採集された時刻 t におけるノード i の格納データ（プライマリおよび他のノードの複製）に対するアクセス負荷値とする．同様に $L_i^{\mathrm{pri}}(t)$ をプライマリデータのみに対するアクセス負荷値とする．$T_i(t)$ と $T_i^{\mathrm{pri}}(t)$ は一時変数で，それぞれ $L_i(t)$，$L_i^{\mathrm{pri}}(t)$ で初期化される．L_i^{MAX} は事前に測定等から算出された，各ノード i の許容最大負荷値とする．

マイグレーション経路選択アルゴリズム

マイグレーションタスクのうちデータ移動元ノードを均等に割り当て，各ノード i における割当てマイグレーションタスク数 s_i を出力する．

Step 1 すべての s_i を 0 で初期化する．

Step 2 ノード i がマイグレーション戦略においてデータ移動先ノードであった場合，インクリメントする（$s_i = s_i + 1$）．

Step 3 $s_i > 0$ であるすべてのノード i について次のことを行う．
- 該当マイグレーションタスクの 2 つのデータ移動元候補ノードのうち，空き負荷値（$L_{k_i}^{\mathrm{MAX}} - T_{k_i}(t)$）が大きい方を選ぶ．ノード k_i とする．
- ノード k_i の割当てタスク数を 1 増やす：$(s_{k_i} \leftarrow s_{k_i} + 1)$．
- ノード k_i の負荷値を増やす：($T_{k_i}(t)$ に $(L_{k_i}^{\mathrm{MAX}} - T_{k_i}(t)) \times (1/\{s_{k_i}(s_{k_i}+1)\})$ を加える)．

Step 4 s_i を出力する．

本アルゴリズムの計算量は $O(N)$ となる．

アクセス回送率計算アルゴリズム

マイグレーション経路選択アルゴリズムで得られた s_i をもとに，クライアントへのアクセスへの処理能力を確保しつつ，マイグレーション対象タスクにより多くの処理能力を確保するため，各ノード i におけるアクセス回送率 r_i を出力する．

Step 1 （変数初期化）ノード i から $i+1$ への回送負荷値 f_i を 0 で初期化する．アルゴリズム内で使用する一時変数 $T_i(t)$ と $T_i^{\mathrm{pri}}(t)$ を，それぞれ $L_i(t)$，$L_i^{\mathrm{pri}}(t)$ で初期化する．

Step 2 ノード番号降順に s_i を走査し，$s_j > 0$ なる j を見つけるたびに次の処理をする．変数

$p \leftarrow j+1$ とする．

(2a) もし $s_p > 0$ なら（図 3 (b)，(c) のパターン），回送により j と p の空き負荷値比を $s_j : s_p$ とする：
- $f_j = \max\Big(0, \frac{1}{s_j+s_p}\{(L_p^{\mathrm{MAX}} - T_p(t))s_j - (L_j^{\mathrm{MAX}} - T_j^{\mathrm{pri}}(t))s_p\}\Big)$
- $T_p(t)$ に f_j を加える．
- $T_j^{\mathrm{pri}}(t)$ から f_j を引く．

Step 2 へ戻る．

(2b) もし $s_p = 0$ なら（図 3 (a) のパターン），(2b) と (2c) でノード j のアクセス負荷を，マイグレーション非対象ノード $j+1, j+2, \ldots, p$ にすべて回送する．その後，途中のノードが溢れたら，(2d) と (2e) で回送し過ぎた負荷を戻す：
- f_{p-1} と $T_p(t)$ に $T_{p-1}^{\mathrm{pri}}(t)$ を加える．
- $T_{p-1}(t)$ から $T_{p-1}^{\mathrm{pri}}(t)$ を引く．
- $T_{p-1}^{\mathrm{pri}}(t) \leftarrow 0$.

(2c) もし $T_p(t) > L_p^{\mathrm{MAX}}$ かつ $s_{(p+1)} = 0$ なら（図 3 (a) の j → k のパターン），$p \leftarrow p+1$ とする．Step 2b へ戻る．

(2d) ノード p の許容最大性能を超えている，もしくは回送設定量がノード p のアクセス量を超えているなら，余剰分 δ_p をノード $(p-1)$ に戻す作業：もし $p > j$ かつ，($s_{p+1} > 0$ または $L_p^{\mathrm{pri}}(t) < f_p$ または $T_p(t) > L_p^{\mathrm{MAX}}$) なら，
- δ_p を $\max(f_p - L_p^{\mathrm{pri}}(t), T_p(t) - L_p^{\mathrm{MAX}})$ とする．
- $T_{p-1}^{\mathrm{pri}}(t)$ と $T_{p-1}(t)$ に δ_p を加える．
- f_p と $T_p(t)$ から δ_p を引く．

(2e) もし $p > j$ なら，$p \leftarrow p-1$ とし，Step 2c へ戻る．

Step 3 $(f_i / L_i^{\mathrm{pri}}(t))$ を回送率 r_i として出力する．

ただし，本アルゴリズムは許容最大負荷値を超える負荷値を持つノードが回送対象に含まれる場合うまく機能しない．その場合，各ノードの負荷値を均衡化するようなアルゴリズムに切り替える必要がある．本アルゴリズムの計算量は最悪の場合 $O(N^2)$ となる．Step 2b 以下が $O(1)$ となれば $O(N)$ となる．これは負荷分布に依存する．

6. 実験概要

本章以降では，本論文で提案した RM 手法の有効性を検証するために行ったシミュレーションによる実験結果を示す．まず本章で，実験設定の概要を説明する．

待ち行列ベースの並列ストレージシミュレーションプログラムを構築し，その上で実験を行う．シミュレーション内のシステムに対し時間的に変化するワークロードを発生させ，ある時点でデータマイグレーションによる負荷均衡化処理を発生させた．そのうえで，各手法の応答性能を観測することで，提案手法とその他の手法の応答性能保持能力を比較する．

本実験ではまず人工的なワークロードを用いて，異なるパラメータ下での挙動を観察し，性能維持効果とマイグレーション速度減少率について検証する．実験では，Zipf 分布に従うアクセス分布とポワソン到着に従うアクセス到着率を持つワークロードを発生させマイグレーションの挙動と応答性能の変化を観測する．2 つ目の実験として，実利用を想定したワークロード下での挙動を観察することで手法の現実システムへの適用可能性を考察する．ここでは，大規模な WEB サーバから採取されたアクセスログを基にしたワークロードを用いる．

6.1 システム構成

待ち行列を利用したイベント駆動型のストレージシミュレーションプログラムを構築し，実験に用いた．シミュレーションはイベントタイマ，イベント，モジュール，ジョブで構成される．イベントタイマにより起動されたイベントがジョブをモジュールに投入することでシミュレーション時間が進む，イベント駆動型をとる．

シミュレーションのモジュール構成を**図 5** に示す．シミュレーション内におけるディスクアクセス時間は，HGST Deskstar T7K250 [22] を基にした**表 1** に示すパラメータと前回アクセス時のヘッド位置から計算する．各ストレージノードのネットワークは 800 Mbps

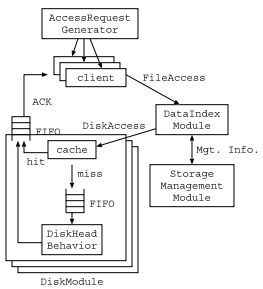

図 5 ストレージシミュレーションのモジュール構成
Fig. 5 Module structure of a system on the simulation.

表 1 シミュレーション構成設定
Table 1 Configuration of storage simulation.

simulation parameter	value
ディスク回転速度 (RPM)	7,200
ディスクサーフェイス数	4
シリンダあたりのセクタ数	2,520 to 5,184
ゾーン数	29
シークタイム（最長，最短）(msec)	14.7, 0.8
ヘッドスイッチ時間 (msec)	1.4
バッファサイズ (KB)	8,196
ディスクノード数	4
ノードキャッシュサイズ (MB)	64
ネットワーク速度 (Mbps)	800

で処理可能な FIFO で表現した．ディスクへのアクセスは buffer size 単位に分割して行う．キャッシュの追い出しアルゴリズムは LRU とした．

6.2 手法の実装

本実験では提案手法である RM 手法を用いたマイグレーション（RM），適応的な速度制御を用いたマイグレーション（speed adjust），および応答性能維持制御を行わない通常のマイグレーション（normal）の 3 つを比較する．また，8 章では参考値として，ディスクやネットワーク処理等をともなわず一瞬でデータ配置を変更した，アクセスのないマイグレーション Zero access migration（za-mig）の結果を用いる．RM 手法および比較対象とする適応的速度制御手法について述べる．

6.2.1 マイグレーション戦略立案

負荷均衡化のためのデータマイグレーション戦略については 3 章に示した戦略を実装した．負荷値については，各データごとに過去 600 秒間のアクセス時間とサイズを保持する．サイズは 6.2.2 項において記す式 (1) により負荷量に変換する．そして，計測期間で割った単位時間あたりの負荷量を負荷値とし戦略立案に用いる．

6.2.2 ノード性能

提案手法のアルゴリズムで必要とする，シミュレーション内のディスク 1 台の許容負荷値 L_i^{MAX} を算出するため，予備実験を行った．ディスク 1 台に対し，異なるサイズごとの読み出しリクエストを発生させ応答性能を測定した．今回目標とするレスポンスタイムを低負荷時のレスポンスタイムの 10 倍と設定した．そこで，低負荷時の 10 倍を超えるレスポンスタイムを示す平均到着率を用いて，本実験に用いる設定のディスクに対するデータサイズ a [KByte] からアクセス負荷量 $l(a)$ を得る次の式を導出した：

$$l(a) = \begin{cases} \frac{a \times 3.4}{10^9} + \frac{5.5}{10^3} & (a \leq 800\,\mathrm{KB}) \\ a/(96 \times 10^6) & (a > 800\,\mathrm{KB}) \end{cases} \quad (1)$$

ただし，ディスクの許容最大負荷量を 1 とした．

また，WEB ワークロードにおける実験では，ワークロード特性からよりキャッシュヒット率が高くネットワークがボトルネックとなった．よって，アクセスサイズによらず上記下側の式を用いた．

6.2.3 適応的速度制御手法

提案手法に対する比較対象として，適応的速度制御手法である Aqueduct [8] を実装した．Aqueduct は階層型ストレージにおいてデータマイグレーションと応答性能保証の両立を達成する適応的速度制御手法である．Aqueduct では各ノードのレスポンスタイムを常時監視し，許容応答性能を超えた場合にマイグレーション速度を動的に減少させることで応答性能維持を達成する．

Aqueduct では，時間間隔 W ごとに時間ウインドウ（$(k-1)W$，kW）における各ノード i の平均レスポンスタイム $L_i(k)$ を取得する．そして応答性能破綻指標 $E_{\min}(k)$ を求める：

$$E_{\min}(k) = \min\{P \times LC_i - L_i(k) \,|\, 0 \leq i < N\} \quad (2)$$

ここで，LC_i は求める応答性能であり，P はマージンを表す制御基準係数である．$E_{\min}(k)$ を用いて，次の時間ウインドウ（kW，$(k+1)W$）におけるマイグレーション速度比 $R_m(k)$ を更新する：

$$R_m(k) = R_m(k-1) + K \times E_{\min}(k) \quad (3)$$

ここで K は安定化係数である．

以上により得られた $R_m(k)$ を用いて，マイグレーション速度を遅延させる．速度遅延はデータ断片 j の転送ごとに，$\max(0, W/R_m(k) - T_j)$ で表される時間インターバルを挟むことにより実現する．ここで T_j はデータ断片 j の転送時間である．

本実験では，$W = 2$ 秒，$P = 0.9$ を用いた．K については，次章では $K = 0.1$，1，5，10 の 4 種類を用いた．その結果，8 章では速度制御の影響が大きい $K = 5$，10 を用いた．$R_m(k)$ の最小値を 0.1 とし，20 秒以上のインターバルが入らないようにしている．マイグレーション単位となるデータ断片として，1 ファイルを用いた．8 章では性能要件を 1 ブロックあたりとしたため，転送ごとのインターバル時間を，$\max\left(0, \frac{WB_j}{R_m(k)} - T_j\right)$ とした．ただし B_j はファイル j の構成ブロック数である．

7. 人工的ワークロードによる実験

実験ではまず人工的なワークロード下での挙動を観察する．

表 2 人工的ワークロード実験緒元
Table 2 Specification of experiments with synthetic workload.

workload parameter	value
時間 (hour)	2
read:write	10:0
格納ファイルサイズ (GB)	100×2 (Primary & Backup)
格納ファイル数	100,000 (1 MB each)
アクセス分布	Zipf $\left(f = \frac{1/k^s}{\sum_N 1/n^s} \right)$
Zipf 係数 s	1.5, 1.2
アクセス到着分布	ポワソン到着
experiment configuration	**value**
アクセス傾向変化時点 (sec)	600
負荷均衡化開始時点 (sec)	900
許容最大応答時間 (msec)	200

7.1 環　　境

表 2 にワークロード緒元を示す．

実験では，Zipf 分布に従うアクセス分布とポワソン到着に従うアクセス到着率を持つアクセスをシャッフル関数を用いて格納ファイルに分散する．Zipf 分布は一部のファイルに多くのアクセスが集中するモデルである．表中の式 f は，N 個のファイルのうち k 番目に多くアクセスされるファイルへのアクセス割合を示し，パラメータ s の値が大きいほど偏りが大きい．そして実験開始 600 秒後にシャッフル関数の乱数種を変化させ，アクセス傾向を変化させる．実験開始 900 秒後にマイグレーションによる負荷均衡化を実行する．その後実験終了までの応答性能およびマイグレーション速度の変化を観察する．以上の実験を，アクセス偏りの度合いを表す Zipf 係数 s が 1.5 の場合と 1.2 の場合について，アクセス分布を決定するシャッフル関数の乱数種およびポワソン到着のアクセス到着率 [req/sec] を変化させる．そのうえで，各マイグレーション性能保持手法使用時のアクセス応答時間を測定する．

今回の実験における必要性能としては 1 リクエストあたりの目標レスポンスタイムを 0.2 秒以下と設定した．6.2.2 項で述べたように，この値は低負荷時の平均レスポンスタイムのおよそ 10 倍の値を目安とし設定した．

7.2 実験結果 1：応答性能保証

まず，**図 6** に大きな偏りのワークロード（Zipf 係数 s=1.5）での各実験における，目標レスポンスタイム超過アクセス数（以下，超過アクセス数）を示す．グラフでは横軸にワークロード負荷の大きさであるアクセス到着率を表し，縦軸に実験におけるレスポンスタイムが目標値である 0.2 秒を超えたリクエスト数の全体に占める割合を表す．各点は，異なる 3 つのシャッフル関数を用いたアクセス分布における実験結果の平均値である．図 6 より，提案手法により超過アクセス数が大きく抑えられているのが分かる．仮に 5%の超過アクセス数を許容できると仮定した場合，提案手法の導入によりこのアクセス偏り度合いのもとでシステムの負荷耐性が 38%上昇したといえる．一方，適応的速度制御ではいずれの K に対しても到着率が 120 [req/sec] から 140 [req/sec] を超えた近辺でシステム性能が維持できなくなっている．さらに，速度制御の度合い K の値が大きくなるほど性能が劣る．この理由については次のマイグレーション時間に関する評価とともに考察する．

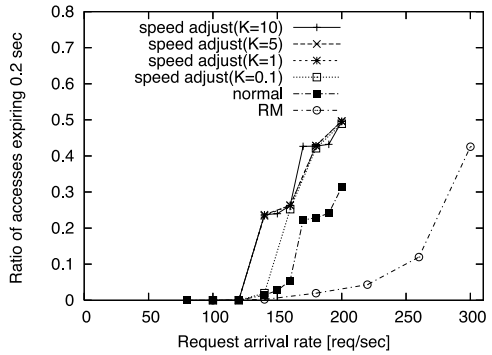

図 6 目標レスポンスタイム超過アクセス数の割合（$s = 1.5$）
Fig. 6 Ratio of violating accesses ($s = 1.5$).

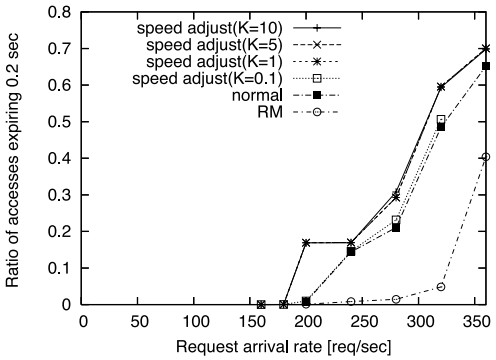

図 7 目標レスポンスタイム超過アクセス数の割合（$s = 1.2$）
Fig. 7 Ratio of violating accesses ($s = 1.2$).

つづいて，アクセス偏りの度合いがより低い Zipf 係数 $s = 1.2$ とした実験の結果を示す．**図 7** は超過アクセス数の割合である．超過アクセス数の割合が急激に上昇する点におけるアクセス到着率を見ると，どの手法も図 6 より 30%から 40%程度大きい．よって，どの手法もより大きな負荷に対応できていることが分かる．しかしながら，手法間の差異に関する傾向は変

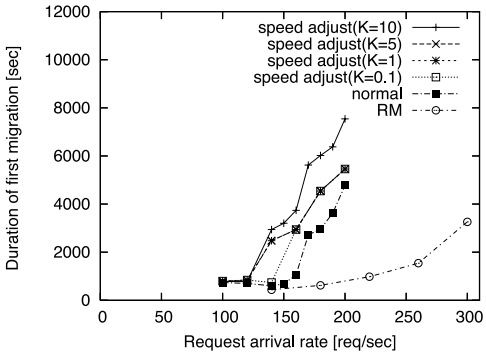

図 8 1 回目のマイグレーションに要した時間（$s = 1.5$）
Fig. 8 Duration of first data migration in each experiments ($s = 1.5$).

わらず，RM 手法を導入することで，システムが要求性能を維持可能なワークロードの最大値が大きく増加している．しかし，この増加率は 240 [req/sec] から 320 [req/sec] と 33%程度であり，$s = 1.5$ 時よりも低い向上率を示した．偏り度合いが低いアクセス負荷では，負荷集中ノード上のアクセス負荷が低くかつ他のノードの負荷が相対的に高いためである．そのためマイグレーションによる性能低下が少なく，アクセス回送率を高く設定できない提案手法の効果が低下したためである．

7.3 実験結果 2：マイグレーション時間

$s = 1.5$ の実験において，実験開始 900 秒後に発生する 1 回目のマイグレーションがデータ転送に要した時間を**図 8** に示す．横軸はアクセス到着率を，縦軸にマイグレーションに要した時間を表す．ただし計測値は 60 秒単位である．なお，各実験では同じ量のデータがマイグレーションされている．図より，適応速度制御では $K = 1$，5，10 の場合 120 [req/sec]，$K = 0.1$ の場合 140 [req/sec] を超えたところでマイグレーションが機能しなくなることが分かる．これは，超過アクセスが発生した場合にマイグレーション速度を下げる処理が働き，マイグレーション処理が止まってしまうためである．速度制御の度合い K の値が大きくなるほど早く速度を下げる．超過アクセスはマイグレーションだけでなくアクセス偏りによっても引き起こされているが，アクセス偏りを是正するはずのマイグレーションを適応的速度制御が止めてしまう．その結果，アクセス偏りが是正されず超過アクセス数が増加している．

図 8 の傾向が図 6 の傾向と酷似していることからもデータ移動速度と応答性能維持の正の相関が見て取れる．RM 手法では応答性能維持だけでなく，マイグレーション時間の削減も同様に達成している．

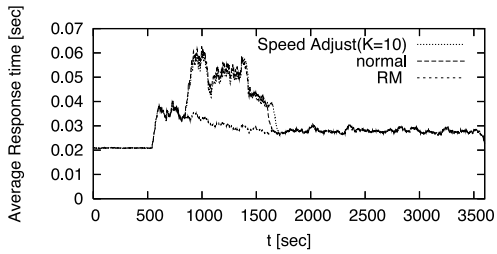

図 9 平均レスポンスタイムの時間推移（緩やかな変化の場合）．アクセス傾向変化後，各ディスクの負荷がそれまでの（2.2 倍，1.5 倍，0.5 倍，0 倍）となった場合．負荷集中ノードではアクセス到着率 < アクセス処理率
Fig. 9 Transitional average respons time when workload change was mild. Load on each node are increased (2.2, 1.5, 0.5, 0) times. Access arrival rate are smaller than service rate.

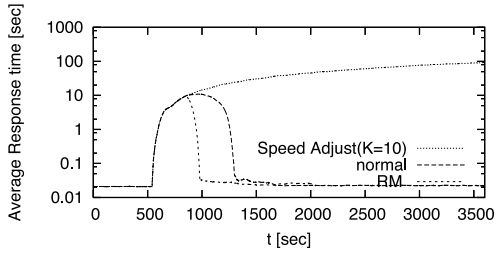

図 10 平均レスポンスタイムの時間推移（急激な変化の場合）．アクセス傾向変化後，各ディスクの負荷がそれまでの（2.7 倍，1.1 倍，0.2 倍，0 倍）となった場合．負荷集中ノードではアクセス到着率 > アクセス処理率となり平均 5 アクセス/秒が当該ノードのキューに溜まっていく
Fig. 10 Transitional average respons time when workload change was rapid and extensive. Load on each node are increased (2.7, 1.1, 0.2, 0) times. Access arrival rate became larger than service rate, so that five accesses per second are enqueued the queue in the node.

7.4 実験結果 3：応答性能の推移

実際の実験中の応答性能の推移を観察する．ここでは特徴的な 2 つの実験について，各手法の応答性能の変化をみる．グラフ中の応答性能はすべて 1 分間の平均である．

まず，**図 9** に変化が緩やかな場合の応答性能の変化をみる．$s = 1.5$，アクセス到着率 140 [req/sec] である．図より，まず $t = 600$ においてアクセス傾向の変化により応答性能が若干悪化しているのが分かる．そして，$t = 900$ において，RM を除く 2 手法が大きく応答性能を悪化させているのが分かる．これはマイグレーション開始時点である．このように，アクセス傾向の変化が緩やかであれば，RM 手法により応答性能を維持しつつマイグレーションを行うことができる．

また，**図 10** はより大きな変化が起きた場合である．$t = 1.5$，アクセス到着率は 140 [req/sec] と同じであ

るが，シャッフル関数に用いた値が異なる．$t = 600$ のアクセス傾向変化ののち，負荷集中ノードのアクセス到着率がアクセス処理率を上回り，未処理リクエストがキューに溜まり始める．その結果応答性能の悪化が進む．normal では $t = 900$ から，データマイグレーションとクライアントアクセスが資源を取り合いながらゆっくりデータ再配置が進行し，応答性能が回復している．一方 RM 手法では，複製保持ノードの処理能力を使い，応答性能が速やかに回復しているのが分かる．適応的速度制御では，$t = 900$ 時点の応答性能が性能要件を著しく超過しているため，応答性能が要件値を下回るまでマイグレーションを遅延させる．しかし，マイグレーションが行われないことで負荷集中ノード上のキュー長は伸び続け，応答性能は悪化し続ける．本実験におけるワークロードはポアソン到着に従っており，平均到着率は時間を通して一定としている．よって，負荷偏りが除去されない限りはキュー中のリクエスト数は減少せず，よって，応答性能は改善せずマイグレーションも行われない．平均到着率が時間とともに変化し，負荷集中ノードにおいてもアクセス到着率が処理率を下回る状態であれば，状況は変化する．そのようなワークロード下での振舞いについては 8 章で紹介する．

このように，緩やかな変化，急な変化であっても，RM 手法ではシステムを応答性能を維持しつつ迅速にマイグレーションを行うことが可能となる．

8. WEB サーバワークロードによる実験

本章では，より実際のシステムへ適用した場合に近いワークロードを用いることで，提案手法の性能保持手法の適用可能性を検証する．

実際のシステムでは前章の実験のような大きなアクセス傾向変化に加え，短期的で突発的なアクセス傾向の変化も含まれる．またアクセスサイズも多様化するため，各ノードの振舞いも複雑化する．よって前章の実験に比べアクセス負荷値の見積り精度，ノードごとの許容最大負荷値精度が低下する．このような条件を含む，現実に近いワークロードによる実験より，提案手法の性能維持能力への影響を考察する．

8.1 環　　境

シミュレーションで用いるシステム負荷は，FIFA WorldCup98 Official WEB サイトから採取されたアクセスログ[23] から抽出した．実験ではこのうち 3 時間分のワークロードを切り出して基本的なワークロードとした．詳細を**表 3** に示す．ワークロード傾向の詳細については文献 24) を参考にされたい．この基本的なワークロードを，ファイルサイズおよびアクセスリクエストサイズを h 倍した複数の設定により実験を行った．これは WEB アクセスログから得られるアクセス分布，アクセスサイズの相対分布，およびそれら分布の変化速度を維持しつつ，異なる複数のワークロードを実現するためである．

表 3 WEB サーバワークロード構成緒元
Table 3 Specification of experiments with real server workload.

workload parameter	value
time span	3 hours
# of requests	約 20,000,000
read:write	10:0
total file size	430 MB × h × 2 (Primary & Backup)
# of total files	21,000

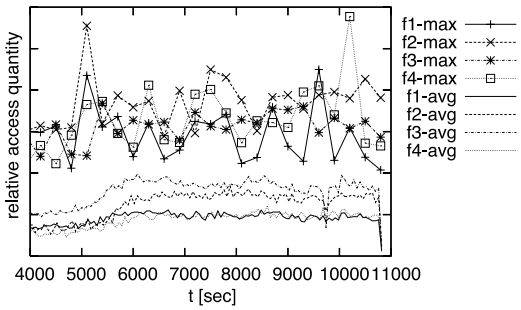

図 11 使用したワークロードの傾向．ファイル数を基準にファイル群を 4 分割し，それぞれに対する 1 分ごとの平均単位時間あたりアクセス要求量（avg）と，5 分ごとの最大単位時間あたりアクセス要求量（max）の時間変化を示した
Fig. 11 Average quantity of requests per a minute as avg and max quantity of requests per 5 minutes as max as a Trend of used workload.

本実験で用いた部分の傾向変化を**図 11** に示す．この図は，格納ファイルをファイル数が 4 等分となるよう f1〜f4 のグループに分けたとき，それぞれのグループに属するファイルへのアクセス要求量の時間変化を表したものである．それぞれ平均と最大を示した．図中の 4 本の avg 曲線より，このワークロードでは t が 5000 から 6000 の間で f2，f3 に大きなアクセス量・比の変化があり，また $t = 8000$ 以降においても f2，f3 間の比に緩やかな変化があることが分かる．また，同図の max 曲線において f2，f4 が高い値を示していることから，f2，f4 に属するファイル中に突発的にアクセスされるファイルが含まれることが分かる．とくに $t = 10000$ での f4-max に注目されたい．

本実験では 60 秒ごとにノード負荷を評価し，ノード間平均負荷が最大許容負荷の 20%以上で，かつあるノードの負荷が平均負荷を 5%以上上回っていたとき

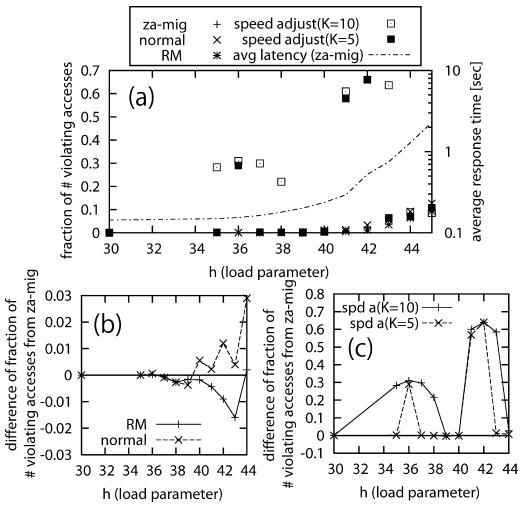

図 12 目標レスポンスタイム超過アクセス数の割合と za-mig との差（WEB ワークロード）．(a) 超過アクセス数絶対値と za-mig の平均応答時間，(b) normal，RM における za-mig との超過アクセス数差，(c) speed adjust（$K = 5$，10）における za-mig との超過アクセス数差

Fig. 12 Ratio of violating accesses (WEB server workload). (a) Absolute ratio, (b) difference from za-mig (RM, normal), (c) diffrence from za-mig (speed adjust).

に，ノード間負荷を平準化する戦略でマイグレーションが行われることとした．

なお，適応的速度制御については，この実験に限り実験開始後 1,500 秒後から適応制御を行うこととした．本実験では実験開始時の負荷均衡状態を，データマイグレーション機能を流用して実現している．しかし適応的速度制御を初期状態より有効にし続けると，極度に偏った状態である初期状態から通常状態に遷移するために必要なマイグレーション処理を，適応的速度制御が止め続けてしまうためである．なお，本実験における応答性能等の合算値は，実験開始後 3,600 秒後からの値を用いた．初期状態へのデータマイグレーションによる影響を排除するためである．

本実験における目標レスポンスタイムは，1 ブロックあたり 1 秒と設定した．低負荷時のレスポンスタイムのおよそ 10 倍の値を目安とし設定している（6.2.2 項参照）．

8.2 実験結果 1：応答性能保証

図 12 (a) にアクセスサイズ倍率 h ごとの目標レスポンスタイム超過アクセス数の割合を示す．本論文における着目点はデータマイグレーション処理による超過レスポンス増加への対処である．しかし図 12 (a) 中の値は，ワークロードの突発的変化等に起因するそれ以外の超過レスポンスも含まれる．そこで，ディスクやネットワーク処理等をともなわず一瞬でデータ配置を変更する，アクセスのないマイグレーション Zero access migration（za-mig）の結果を併記した．za-mig はシミュレーションプログラム上で動作しているストレージシステムにおいて，ある瞬間にデータ配置情報（どのディスクのどのセクタが該当データを格納しているかという情報）をシミュレーション時間を進ませずに書き換えることで実現した．za-mig の結果により，データマイグレーション処理に起因するアクセスによらない超過レスポンス数のおおよその値が確認できる．ただし，マイグレーション処理のアクセスを除去することにより，負荷評価タイミング，データ配置変更タイミングが変化する等の理由により，アクセスのあるマイグレーションよりも低い性能を示す可能性があることに注意されたい．za-mig は参考値であり，za-mig との差はデータマイグレーション処理のアクセスに起因する超過レスポンス数ではない．図 12 (a) に za-mig における超過レスポンス数と za-mig における平均レイテンシを併記した．また図 12 (b)，(c) には，za-mig の平均レイテンシが性能要件を超える $h = 44$ までの各手法における超過レスポンス数割合の za-mig との差を示した．

図 12 (a) より，$h \leq 30$ までは，いずれの手法も同様の応答性能を保持していることが分かる．これは $h \leq 30$ ではノード性能を超えるようなワークロード集中が散発的であり，各性能維持手法の発動回数が低いためである．

図 12 (b) より，normal では za-mig よりも最大で 3%程度，全体に占める超過リクエストの割合が増加している．一方提案する RM 手法では za-mig と同等以下の超過リクエスト数割合を維持している．図 12 (b) 中のそれぞれの正の値の合計を比較すると，RM は normal のおよそ 7%であり，データマイグレーションに起因する超過リクエスト数を大きく削減していることが分かる．

図 12 (c) より，speed adjust では $h = 36$ や $h = 43$ では大きく超過アクセス数を増加させてしまっていることが分かる．これは負荷均衡化に対して重要なタイミングでのマイグレーションを止めてしまったためである．**図 13** に $h = 35$ における応答性能とインターバルの関係を示す．図 11 で示したとおりアクセス量が増加し応答性能が悪化する．$h = 35$ ではマイグレーション発動時点が遅く，$t = 5500$ で性能要件値を超えたため適応的にインターバルを増加させる．インターバルの増加が負荷不均衡を助長するため，応答性能は悪化し続ける．本実験では h の値ごとにマイグレーション発動時点が異なる．上記のようなマイ

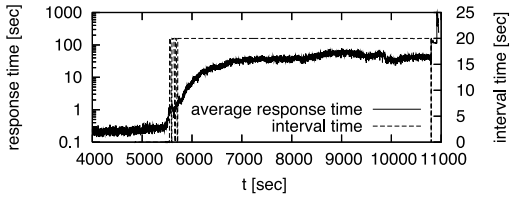

図 13 $h = 35$, $K = 10$ における平均応答時間とマイグレーション間インターバル時間の時間推移.

Fig. 13 Transitional average response time and interval between migration ($h = 35$, $K = 10$).

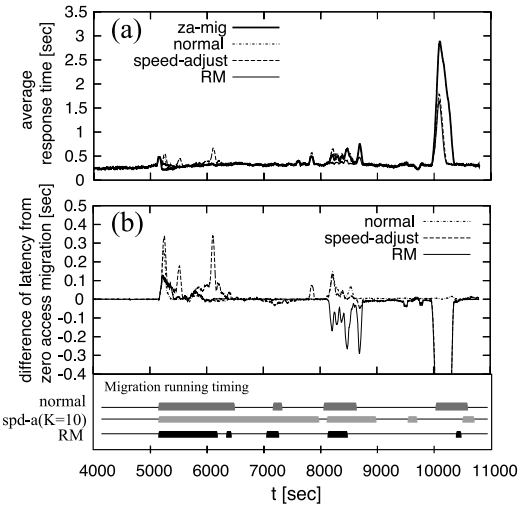

図 14 $h = 40$ における各手法の時間推移. (a) 絶対的平均応答時間の時間推移, (b) za-mig との平均応答時間差の時間推移と各手法のマイグレーション実行タイミング

Fig. 14 Transitional average response time (WEB server workload). (a) response time (absolute), (b) difference of response time from za-mig and migration execution timing.

グレーション中の負荷不均衡が 1 度でも起きた場合，適応的速度制御がマイグレーション機能をずっと停止させ続けてしまう．しかし，人工的ワークロードによる実験と異なり，$h = 39$，$h = 40$ では speed-adjust でも超過アクセス数を削減できている．これは，人工的ワークロードと異なり WEB ワークロードはアクセス量が時間とともに変化するためである．マイグレーションを遅延させることで，ワークロードの小さくなるタイミングのノード資源を効率良く利用できている場合もある．

8.3 実験結果 2：応答性能の時間変化

マイグレーション発生時の応答性能の変化を観測する．**図 14** (a) は $h = 40$ における，それぞれの手法使用下の応答性能変化の推移を表している．また，図 14 (b) は za-mig との差を，図 14 (c) はマイグレーションが実行されたタイミングを表す．

この実験では $t = 5100$，8100，9600 付近で大きなデータマイグレーションが発生している．なお，この実験における 3 手法の平均スループットは毎秒 228 MByte で同等であり，いずれの手法もスループットへの影響はない．

まず $t = 5100$ 付近に注目する．図 14 (b) より，どの手法でも za-mig と比較して応答時間が上昇しているが，RM 手法が 0.1 [sec] と最も小さいことが分かる．適応的速度制御では，マイグレーションを遅延させた結果，$t = 8000$ 付近までマイグレーションが継続し，$t = 6000$ 付近で応答性能の悪化を引き起こしている．$t = 8100$ 付近では，RM 手法以外では za-mig と比較して応答性能を悪化させている一方，RM 手法では複製ノードを用いて応答時間が削減されている．

なお，$t = 10000$ 付近の急激な応答時間上昇は，図 11 の f4-max より，アクセス負荷の突発的な変化に起因するもので，本論文の対象範囲外の挙動であることに注意されたい．また，RM や speed-adjust の手法がマイグレーションを行っていない（速度制御や複製利用を行っていない）にもかかわらず，za-mig が相対的に悪くなっている．これは，この直前までのアクセス傾向に za-mig でのデータ配置がよりよく合致していたため，突発的で過去の傾向から逸脱したアクセスによる応答時間の上昇が顕著だったためと考えられる．このように，za-mig はマイグレーション処理に起因するアクセスを除去してあるが，必ずしも理想的性能を示すものではない．

このように，データマイグレーションによる一時的な応答性能悪化は通常の運用で起こりうることである．提案する RM 手法を用いることで，データマイグレーションに起因する応答性能悪化を大幅に削減することが可能となることが示された．

9. ま と め

高機能並列ストレージシステムでは，負荷均衡化のためのデータマイグレーションに起因する一時的な応答性能低下が存在する．本論文ではデータマイグレーション時においても要求される応答性能を達成するための手法 Replica-assisted Migration（RM 手法）を提案した．本手法では，マイグレーション経路選択とアクセス回送の 2 手法に複製を利用する．提案する RM 手法では 2 手法を，ストレージノードの最大許容負荷情報をもとに，負荷集中ノード，マイグレーション対象ノード，そしてそれらの複製保持ノードそれぞれが性能要件を維持可能な許容最大負荷以下の負荷で

動作可能となることを目指し複製制御を行う．そのため極端な負荷集中後においても性能要件を維持しつつ迅速なデータマイグレーションを実現する．

人工的なワークロードを用いた実験により RM 手法と適応的速度制御手法を比較した．その結果，読み出しアクセス中心のワークロード下においては，これまでのシステムでは対応できなかった高負荷・高負荷偏り環境下においても高い性能維持能力を確認できた．また，現実に近いワークロードとして大規模 WEB サーバのアクセスログを用いた実験から，提案手法の現実のシステムへの実現可能性も示された．

今後の課題として，更新要求の取扱いは重要である．提案手法のうち，経路選択による効果はデータ挿入・更新要求混在時でも得られるが，アクセス回送は難しい．より柔軟なデータ配置制御，WAL やライトバックキャッシュによる書き込み負荷の軽減を組み合わせることで，このような問題に対処することを検討している．またストレージノード性能の見積りの簡素化・自動化は重要な要素である．今回の実験では，予備実験の結果等からノード性能の上限を算出した．これらの値をより高い精度で推定可能な手法を考案することで提案手法の効果はさらに高まると考える．加えて，この性能見積り手法を改良することで本提案手法の異機種混在環境へ拡張することも検討している．

突発的なアクセス負荷上昇への対処や，アクセス回送常用化と本提案手法の併用については本論文の対象外としたが，重要な課題と考えられる．また，より詳細な手法評価として標準的なベンチマークツール等，より一般的な環境への有用性評価は重要である．以上の課題を達成したうえで，堅牢でディペンダブルな並列ストレージシステムの構築に関する検討を進めていきたい．

謝辞 本研究の一部は，独立行政法人科学技術振興機構戦略的創造研究推進事業 CREST，独立行政法人日本学術振興会科学研究費補助金特別研究員奨励費，情報ストレージ研究推進機構（SRC），文部科学省科学研究費補助金特定領域研究（18049026，19024028）および東京工業大学 21 世紀 COE プログラム「大規模知識資源の体系化と活用基盤構築」の助成により行われた．

参 考 文 献

1) Yokota, H.: Autonomous Disks for Advanced Database Applications, *Proc. International Symposium on Database Applications in Non-Traditional Environments* (*DANTE'99*), pp.441–448 (1999).
2) Frølund, S., Merchant, A., Saito, Y., Spence, S. and Veitch, A.: FAB: enterprise storage systems on a shoestring, *HOTOS 2003*, Kauai, HI (2003).
3) Ganger, G.R., Strunk, J.D. and Klosterman, A.J.: Self-* Storage: Brick-based Storage with Automated Administration, Technical Report CMU-CS-03-178, Carnegie Mellon University (2003).
4) Simitci, H.: *Storage Network Performance Analysis*, Wiley Technology Publishing (2003).
5) Weikum, G., Mönkeberg, A., Hasse, C. and Zabback, P.: Self-tuning Database Technology and Information Services: from Wishful Thinking to Viable Engineering., *VLDB*, pp.20–31 (2002).
6) Feelifl, H., Kitsuregawa, M. and Ooi, B.: A Fast Convergence Technique for Online Heat-balancing of Btree Indexed Database over Shared-nothing Parallel Systems, *11th Int'l Conf. on Database and Expert Systems Applications*, pp.846–858 (2000).
7) Dasgupta, K., Ghosal, S., Jain, R., Sharma, U. and Verma, A.: QoSMig: Adaptive Rate-Controlled Migration of Bulk Data in Storage Systems, *the 21st International Conference on Data Engineering* (*ICDE2005*), pp.816–827 (2005).
8) Lu, C., Alvarez, G.A. and Wilkes, J.: *Aqueduct:* online data migration with performance guarantees, *Conference on File and Storage Technologies* (*FAST'02*), Monterey, CA, pp.219–230 (2002).
9) Zhang, J., Sarkar, P. and Sivasubramaniam, A.: Achieving completion time guarantees in an opportunistic data migration scheme, *SIGMETRICS Perform. Eval. Rev.*, Vol.33, No.4, pp.11–16 (2006).
10) 小林 大，渡邊明嗣，山口宗慶，田口 亮，上原年博，横田治夫：複製データを併用した効率的なデータマイグレーションの検討，日本データベース学会 Letters，Vol.3, No.2, pp.65–68 (2004).
11) Scheuermann, P., Weikum, G. and Zabback, P.: Data Partitioning and Load Balancing in Parallel Disk Systems., *VLDB J.*, Vol.7, No.1, pp.48–66 (1998).
12) Hsiao, H.-I. and DeWitt, D.J.: Chained Declustering: A New Availability Strategy for Multiprocessor Database Machines, *Proc. 6th International Conference on Data Engineering*, Los Angeles, CA, IEEE Computer Society, pp.456–465 (1990).
13) Lumb, C.R., Merchant, A. and Alvarez, G.A.:

Façade: Virtual Storage Devices with Performance Guarantees, *2nd USENIX Conference on File and Storage Technologies* (*FAST'03*), pp.131–144 (2003).
14) Kistler, J.J. and Satyanarayanan, M.: Disconnected operation in the Coda File System, *ACM Trans. Comput. Syst.*, Vol.10, No.1, pp.3–25 (1992).
15) Saito, Y. and Levy, H.M.: Optimistic Replication for Internet Data Services, *Distributed Computing, 14th International Conference* (*DISC*), Toledo, Spain, pp.297–314 (2000).
16) 原 隆浩，春本 要，塚本昌彦，西尾章治郎，奥井 順：広帯域ネットワーク上のデータベース移動に基づく動的複製配置法，電子情報通信学会論文誌，Vol.J-82-D-I, No.8, pp.1049–1058 (1999).
17) Wolf, J.L., Yu, P.S. and Shachnai, H.: Disk Load Balancing for Video-On-Demand Systems, *Multimedia Systems*, Vol.5, No.6, pp.358–370 (1997).
18) Lumb, C.R., Golding, R. and Ganger, G.R.: DSPTF: Decentralized Request Distribution in Brickbased Storage Systems, *Proc. ASPLOS'04*, Boston, MA (2004).
19) 小林 大，渡邊明嗣，田口 亮，上原年博，横田治夫：データ移動コストとキャッシュを考慮した複製へのアクセス分散制御，日本データベース学会 Letters，Vol.4, No.1, pp.125–128 (2005).
20) 渡邊明嗣，横田治夫：分散ディレクトリ探索コストを考慮した並列データアクセス偏り制御，電子情報通信学会和文論文誌 DI，Vol.85-DI, No.9, pp.877–886 (2002).
21) Bernstein, P.A. and Goodman, N.: An Algorithm for Concurrency Control and Recovery in Replicated Distributed Databases, *ACM Trans. Database Syst.*, Vol.9, No.4, pp.596–615 (1984).
22) Hitachi Global Storage Technologies: *Deskstar T7K250 Hard Disk Drive Specification*, ver. 1.7 edition (2006). http://www.hitachigst.com
23) Lawrence Berkeley National Laboratory: The Internet Traffic Archive. http://ita.ee.lbl.gov/
24) Arlitt, M. and Jin, T.: Workload Characterization of the 1998 World Cup Web Site, Technical Report HPL-1999-35R1, Hewlett-Packart Laboratories (1999).

(平成 19 年 9 月 20 日受付)
(平成 20 年 1 月 13 日採録)

(担当編集委員 片山 紀生)

小林 大

2003 年東京工業大学工学部情報工学科卒業．2005 年東京工業大学大学院情報理工学研究科計算工学専攻修士課程修了．現在，同専攻博士後期課程在学中．2006 年より日本学術振興会特別研究員 DC．並列ストレージシステム，データ工学等の研究に従事．日本データベース学会学生会員．

横田 治夫（正会員）

1980 年東京工業大学工学部電子物理工学科卒業．1982 年東京工業大学大学院理工学研究科情報工学専攻修士課程修了．同年富士通（株）．同年 6 月（財）新世代コンピュータ技術開発機構研究所（ICOT）．1986 年（株）富士通研究所．1992 年北陸先端科学技術大学院大学情報科学研究科助教授．1998 年東京工業大学大学院情報理工学研究科助教授．2001 年東京工業大学学術国際情報センター教授．工学博士．主として分散インデキシング，データ工学向けアーキテクチャ，高機能ストレージシステム，ディペンダブルシステム等に関する研究に従事．電子情報通信学会フェロー．日本データベース学会理事．人工知能学会，IEEE，ACM 各会員．