過渡状況を考慮した 分散データ格納環境のための並列偏り除去手法

渡邊　明嗣[†]　花井　知広[†]　山口　宗慶[††]　田口　亮[†††]　林　直人[†††]
上原　年博[†††]　横田　治夫[††††]

† 東京工業大学 大学院 情報理工学研究科 計算工学専攻; 152-8552 東京都目黒区大岡山 2-12-1
†† 東京工業大学 工学部 情報工学科; 152-8552 東京都目黒区大岡山 2-12-1
††† NHK 放送技術研究所; 157-8510 東京都世田谷区砧 1-10-11
†††† 東京工業大学 学術国際情報センター; 152-8550 東京都目黒区大岡山 2-12-1
E-mail: †{aki,hanai}@de.cs.titech.ac.jp, ††muu@de.cs.titech.ac.jp,
†††{taguchi.r-cs,hayashi.n-gm,uehara.t-jy}@nhk.or.jp, ††††yokota@cs.titech.ac.jp

あらまし　我々の提案している自律ディスクでは，クラスタ構成変更に対する効率的な再配置を行うため，データを値域分割によって各ディスクに分配する．値域分割による分配では，負荷偏りを除去するための動的な再配置戦略が重要である．我々は，スケーラビリティに注目し，大域的な負荷分布に基づいて移動計画を作成するアルゴリズムを提案してきた．しかしながら，大域的な負荷分布のみに基づく移動計画には，移動過程で局所的偏りが生じる問題がある．本稿では，大域的な負荷分布に基づいて作成された移動計画に隣接間での負荷情報の交換を組み合わせることで，高いスケーラビリティを保ちながら，過渡的偏りを抑制する手法を提案する．
キーワード　自律ディスク, 偏り制御, データ配置, 負荷評価

A Transient States Conscious Skew-Handling Technique for Distributed Data Storage

Akitsugu WATANABE[†], Tomohiro HANAI[†], Munenori YMAGUCHI[††], Ryo TAGUCHI[†††], Naoto HAYASHI[†††], Toshihiro UEHARA[†††], and Haruo YOKOTA[††††]

† Department of Computer Science, Graduate School of Information Science and Engineering, Tokyo Institute of Technology
2-12-1 Oookayama Meguro Tokyo, 152-8552 Japan
†† Department of Computer Science, Faculty of Engineering, Tokyo Institute of Technology
††† NHK Science & Technical Research Laboratories
1-10-11 Kinuta Setagaya Tokyo, 157-8510 Japan
†††† Global Scientific Information and Computing Center, Tokyo Institute of Technology
E-mail: †{aki,hanai}@de.cs.titech.ac.jp, ††muu@de.cs.titech.ac.jp,
†††{taguchi.r-cs,hayashi.n-gm,uehara.t-jy}@nhk.or.jp, ††††yokota@cs.titech.ac.jp

Abstract　The Autonomous Disks distribute data objects into each disks by using a value range partitioning strategy to reallocate data efficiently for cluster reconfigulation. When using the value range partitioning strategy, some dynamic reorganization algorithm for handling skew is required. To obtain higher scalability, we have proposed a migration planning algorithm based on global load distribution. However transient skew may arise if the planning algorithm does not consider local load distribution. In this paper, we propose a skew handling method by using planning based on both global and local load distribution to control transient skews with high scalability.
Key words　Autonomous Disks, skew handling, data reorganization, load evaluation

1. はじめに

プロセッサエレメント（PE）間でディスク，メモリを共有しない並列無共有システムはそのスケーラビリティ，アベイラビリティ，費用対効果の高さから大いに注目されている．並列無共有システムでは，処理を PE 間に分散させて並列実行することによって高いスループットを実現するため，負荷が偏っている場合は，実行時間は最も負荷の大きい PE が処理を完了するまで引き延ばされてしまう．したがって，並列無共有システムの性能および規模の向上を線形的にするためには，負荷の偏りを制御する負荷分散技術が非常に重要である [1].

並列無共有システムのデータベースへの応用においても，PE の負荷を平均化する技術が数多く研究されてきた [2]〜[6]．データベース問い合わせ処理の過程における中間データの生成のように負荷の偏りには何種類か理由があるが，負荷分散に最も大きな影響を与えるのはデータ配置である．データベースのデータの水平分割は並列データベースにおける各 PE での操作において特に重要である [2].

データを PE に割り振る水平分割戦略は，ハッシュ，ラウンドロビン，値域分割の 3 つに大別される．これらのうち，値域分割による水平分割には，クラスタ I/O が利用可能かつ，レンジクエリや近傍探索の効率的な実装が可能で，しかも，分割数変更時の再配置コストが低いという特長がある．しかしながら，値域分割による水平分割にはデータベースの運用に伴ってデータオブジェクトの追加・削除およびアクセスパターンの変化によって負荷に偏りが生じる問題があるため，これを取り除くためのデータの動的再配置，すなわち動的偏り制御が必要である．

読み書きの両方において高速にアクセスを行うことができる分散ディレクトリ構造は，高速な動的偏り制御に大いに役立つ．値域分割の長所を利用しつつ各 PE への高速アクセスを実現するための並列ディレクトリ構造として，並列 Btree が提案された [7]．我々が提案した Fat-Btree は更新手続きのコストを考慮した並列 Btree 構造であり，インデクス木の根からの距離に応じてノードの冗長性が減少する構成方法を用いることによって，インデクス探索の負荷を効率的に分散し，また，値域分割の境界変更を効率的に行う [8].

我々は負荷情報を伝達し新しい配置の決定するためのスケーラブルな制御方式 TCSH（Tree-Communication-Skew-Handling parallel control）を提案している [9], [10]．TCSH では，オブジェクトの移動先が隣接 PE に限られるという値域分割の再配置が持つ制約に注目し，隣接 PE にオブジェクトを移動すべきかどうかを判断するための最低限の情報のみを通信木に沿って交換する方式を用いることで，高いスケーラビリティを実現している．また，我々はアクセスパターンの揺らぎや負荷評価の誤差によって引き起こされる過剰な移動の抑制と，分割数変更時の速やかな偏りの解消を達成するために，移動速度決定戦略 RSH（Reconstruction conscious Skew Handling）[11] を提案している．

通信木を用いた制御は，PE 間の通信と依存関係を疎にすることによってスケーラビリティの高い負荷情報の伝達と新しい配置の決定を実現する．しかし，偏りが大きく，かつ大容量のオブジェクトを含む状況では，TCSH による制御が性能の低下を招く場合があった．新しい配置に遷移するまでの過程で負荷偏りが大きい過渡的な状態を経由する場合や，境界近傍に極めて高い負荷を持つオブジェクトがあるなどの理由で TCSH が決定した移動の一部が行われなかった場合に過渡的かつ局所的な偏りが生じ，性能が低下する．

本稿では，偏りが大きくかつデータサイズが大きいオブジェクト（大容量オブジェクト）を多数含む状況における，スケーラビリティと過渡的かつ局所的な偏りの抑制の双方に注目した動的偏り除去機構を提案する．本稿の提案では，移動可否の判断に局所的な負荷分布の分析を利用することで，TCSH によって達成しているスケーラビリティを損なわずに過渡的かつ局所的な偏りの抑制を達成する．また，我々が提案している自律ディスク [12] を用いたマルチメディアコンテンツサーバ [13] 上に提案手法を実装し，評価実験を行う．以下，2. 節では偏り除去機構の提案を行う．3. 節では提案手法の評価実験を行う．最後の節では本稿を総括し，また現時点における研究課題について述べる．

2. 偏り除去機構

本稿では，大容量オブジェクトの格納を目的とした並列データベースに注目して議論する．また，本稿ではデータの水平分割戦略に値域分割戦略が用いられていることを仮定する．これは，値域分割を用いた分割を行うことで，レンジクエリ，完全一致クエリを効率的に扱うことができ，さらにクラスタ I/O の利用や，分割数変更時の効率的な再構成を行うことができるためである．

2.1 分散ディレクトリ構造

B-tree を基底とした階層型インデクスは，値域分割戦略を実現する効率的な手法の一つである．典型的な階層型インデクス方式では，PE に割り付けられた値域とキーとの対応を決定するための大域インデクスと，PE 内でのキーに対応するデータの所在を決定するための局所インデクスの 2 つのインデクスを用いてデータの管理を行う．我々が提案している Fat-Btree [8] は，階層型インデクス方式と同様に値域分割戦略を効率的に取り扱うことを目的としているが，インデクス木の根からの距離に応じてノードの冗長性が減少する構成方法を用いることによって，インデクス探索の負荷を効率的に分散し，かつ，分割境界変更のコストをも軽減する特長がある．

2.2 オンライン負荷分散

本稿では，オンライン負荷分散は以下の 5 つのステップを経て実行される [11], [14].

（1） PE 負荷の計測
（2） 負荷分布の伝達
（3） 配置決定
（4） 分割境界移動についての実行時判断
（5） 実際のデータオブジェクト移動

2.2.1 PE 負荷の計測

本稿では，大容量オブジェクトの格納を目的としたデータベースに注目して議論することから，アクセス量による評価 [13] を

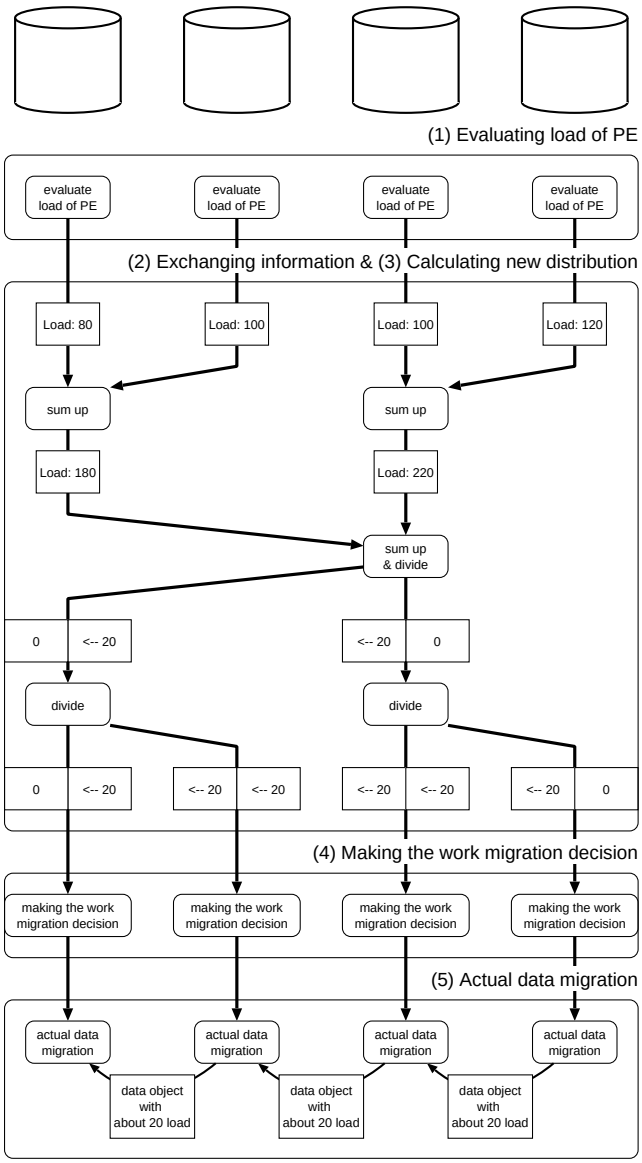

図 1　TCSH による負荷分布の伝達と配置決定の例

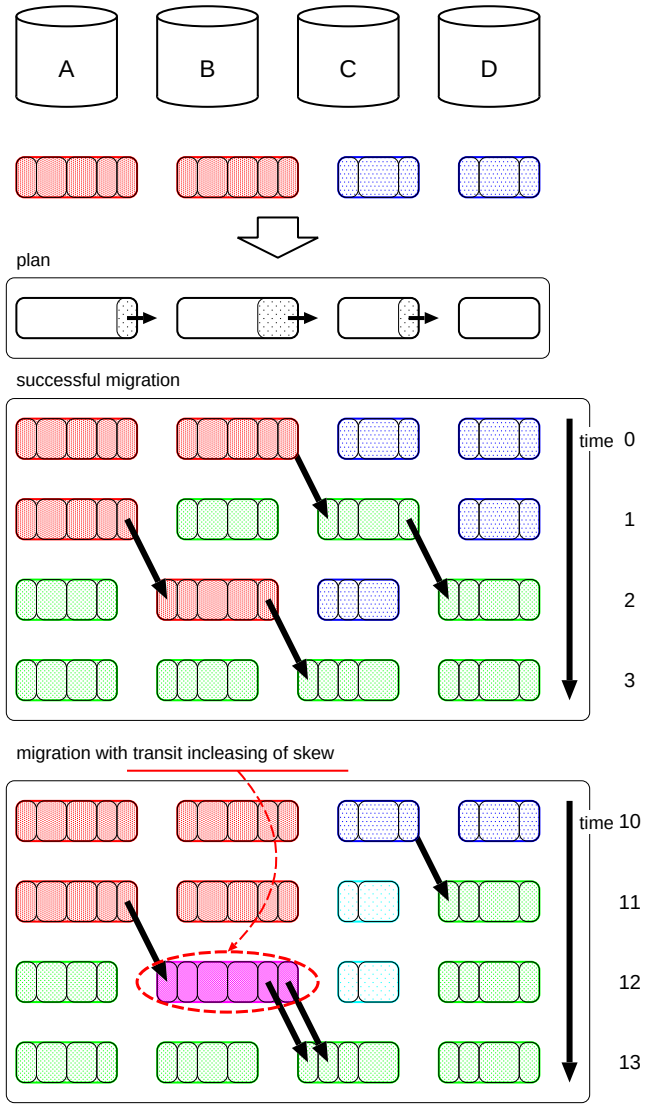

図 2　移動順序による過渡的偏り増大

用いた (1) PE 負荷の計測を行う．この評価指標では，負荷が観測時間内にアクセスされたデータ量に比例することを仮定し，インデクスの探索コストや CPU 負荷を無視する．大容量オブジェクトのアクセス履歴を収集するコストはオブジェクトそれ自体に対するアクセスコストに比べて小さいため，本稿ではこれを無視する．アクセスパターンの動的な変化に対応するためには，最新の状況を反映するようなアクセス履歴への重み付けが必要である．本稿では一定回数毎にアクセス履歴の重みが半減するような重み付けを行っている．

2.2.2　負荷分布の伝達と配置決定

我々が提案している偏り除去の並列制御手法 TCSH では，(2) 負荷分布の伝達と (3) 配置決定の 2 つの段階において，通信木を用いた手法の利用することにより，均衡化した配置に至るための分割境界の移動量の算出をスケーラブルに行う（図 1）[9], [10]．TCSH は隣接 PE へ移動させるべき負荷の量を各 PE に指示する．境界近傍のオブジェクトの負荷などを考慮した移動についての詳細の決定は，各 PE において行われる分割境界移動についての実行時判断に委ねられる．TCSH では，通信木を用いて負荷情報の集計を行うことで処理の並列度を高め，トークン巡回方式を用いた並列制御方式に比べて高いスケーラビリティと精度を実現している [10].

しかしながら，TCSH が決定するのは，新しい配置とその配置に至るために必要な負荷の移動量のみであり，移動順序や移動が可能であるかどうかの判断は行わない．そのため，新しい配置に遷移するまでの過程で負荷偏りが大きい過渡的な状態を経由するような移動順序が選ばれてしまったり，境界近傍に極めて高い負荷を持つオブジェクトがあるなどの理由で TCSH が決定した移動の一部が行われなかった場合に過渡的かつ局所的な偏りが生じ，性能が低下する．

図 2 は，移動順序が規定されていないことによって過渡的に偏りが増大する状況の例である．図中のシリンダがクラスタを構成する PE に，シリンダの下に描かれたベルトの長さが対応する PE の負荷に相当する．ベルトに重ねられたボックスはオブジェクトと対応しており，負荷の移動はこのボックス単位でのみ行われる．最上段に示された偏りのある初期状態に対す

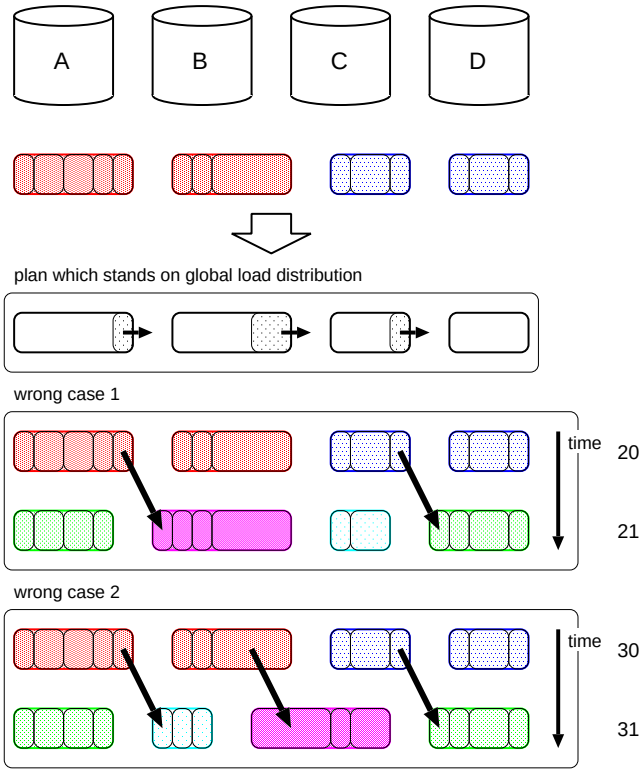

図 3　移動実行時の状況に起因する偏り増大

る TCSH の移動計画が plan であり，グレーで示された部分が移動する負荷の量，矢印が移動する隣接 PE の方向を示している．初期状態において，TCSH は A,B が C,D よりも負荷が高い状態を解消するべく，plan の枠内に示されたような移動計画を立案した．time 0 から始まる系列はこの移動計画に沿った理想的な移動が行われた状況を，time 10 から始まる系列は同じくこの移動計画に沿っているものの，過渡的に偏りが増大してしまった状況をそれぞれ示している．2 つの系列の違いは移動が発生した順序である．TCSH が計画した 4 つの PE 間のオブジェクト移動を time 0 から始まる系列では B,C 間の移動，A,B および C,D 間の移動，B,C 間の移動の順に解決しているのに対し，time 10 から始まる系列では C,D 間の移動，A,B 間の移動，C,D 間の移動の順に解決しており，初期状態において負荷が高かった B の負荷が一層高くなるような状態 (time 12) を経由してしまっている．

図 3 は，移動実行時の状況に起因する偏り増大の例である．初期状態において，TCSH は A,B が C,D よりも負荷が高い状態を解消するべく，plan の枠内に示されたような移動計画を立案した．しかし，この例では B,C 間の境界近傍にあるオブジェクトの負荷が大きいために，計画を満たすように移動することができない．time 20 から始まる系列では B,C 間の移動が行われなかったにも関わらず A,C 間，C,D 間で移動が行われたために，初期の状態よりも偏りが大きくなってしまっている．time 30 から始まる系列では B,C 間の移動が行われたが，移動計画よりも大きな負荷が移動したために，やはり初期の状態よりも偏りが大きくなってしまっている．

移動順序による過渡的偏り増大は予測可能であり，配置決定時に移動順序を規定するような要素，例えば，移動順序を満たすような遅延時間を設定するなどで回避できるが，移動順序の決定と伝達に余分なコストが生じる．移動実行時の状況に起因する偏り増大の軽減には，境界近傍の負荷状態の伝達が必要であり，境界近傍にあるオブジェクトの負荷計測と負荷情報の伝達に余分なコストが生じる．また，いずれの場合にも配置決定前に取得した情報を利用するため，状況の変化によって精度が落ちる可能性がある．

より効率的で精度が高い解決のため，我々は次節において (4) 分割境界移動についての実行時判断の段階でこのような過渡的偏り増大および移動実行時の状況に起因する偏り増大を軽減する効率的手法を提案する．

2.3　分割境界移動についての実行時判断

本節では，(4) 分割境界移動についての実行時判断を，前節で述べた過渡的偏り増大および移動実行時の状況に起因する偏り増大を軽減する機能に注目して議論する．

分割境界移動についての実行時判断を行う段階では，配置決定戦略が決定した移動すべき負荷の量を基準に移動するデータオブジェクトを決定する．最も単純な実行時判断では，配置決定戦略が決定した移動すべき負荷の量に見合う量の負荷を持ったデータオブジェクトの集合を移動するデータオブジェクトにする．速度係数（speed factor）[14] を取り入れた実行時判断の手法では，配置決定戦略が決定した移動すべき負荷の量に速度係数と呼ばれる定数または関数を掛けた値に見合う量の負荷を持ったデータオブジェクトの集合を移動するデータオブジェクトにすることで，移動を分割して一度に移動するデータオブジェクトの量を減らし，負荷評価の誤差などによる配置決定の偏り増大を軽減する [11], [14].

2.3.1　クラスタ再構築を考慮した実行時判断手法

我々は速度関数を用いた実行時判断としてクラスタ再構築を考慮した手法 RSH（Reconstruction conscious Skew Handling）を提案している [11]．この手法では，配置決定戦略が決定した移動すべき負荷の量 h を $l = h \times \mathrm{rsh}(h, a)$ のように正規化し，移動したオブジェクトの負荷の合計が負荷の量 l を超える直前までオブジェクトを移動する．

$$\mathrm{rsh}_{\lambda,\chi,\zeta}(h,a) = \begin{cases} 0 & (h/a < \zeta) \\ \lambda(h/a) & (\zeta \leqq h/a \leqq \chi/\lambda) \\ \chi & (\chi/\lambda < h/a) \end{cases}$$

ただし，h は移動予定の熱，a は移動分割を行うノードの負荷．λ は過剰負荷率が与える影響の大きさ，χ は移動速度の上限，ζ は移動の閾値を表す．

速度係数を取り入れた手法は分割境界が移動する速度を低く抑えるため，図 2 のような移動順序による過渡的偏り増大を軽減する効果が期待できる．しかしながら，図 3 のような移動実行時の状況に起因する偏り増大を軽減する効果は期待できない．

2.3.2　隣接 PE 間通信による偏り増大の抑制

過渡的偏り増大および移動実行時の状況に起因する偏り増大を回避するために，我々はクラスタ再構築を考慮した手法を発展させ，これに移動先と移動元の PE のみからなる小さな領域

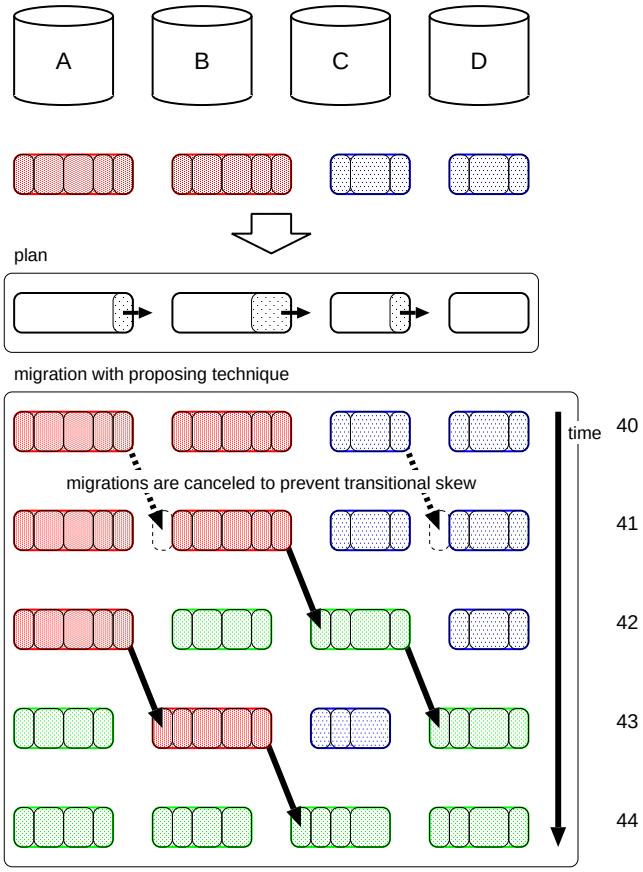

図 4　移動順序による過渡的偏り増大の抑制

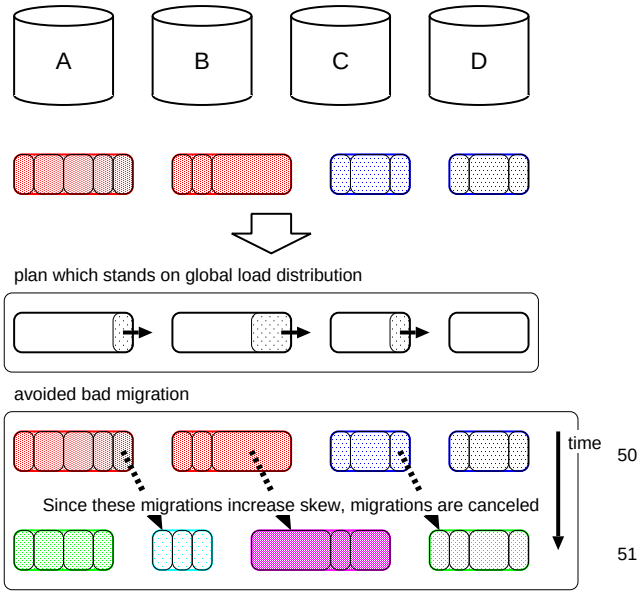

図 5　局所的負荷分布によって起る偏り増大

を用いた管理を導入する．我々が問題としている偏り増大は，“その時点において” 負荷が小さい PE から負荷が大きい PE へとデータオブジェクトの移動が行われることである．PE の負荷情報を隣接 PE のみからなる小さな領域内で交換することによって，そのような移動の可能性を検出し，その時点における領域内の負荷分布に応じた分割境界移動を行うことができる．我々が提案する実行時判断手法では，配置決定戦略が要求する負荷の移動量に速度係数 α を掛けた値に達するまで，データオブジェクトを移動する．ただし，移動後に移動先 PE の負荷が移動元 PE の負荷を閾値 τ を超えて上回る場合には，それ以上のデータオブジェクトの移動を行わず，一定時間 i の待機を挟んだ再試行を一定回数 r_m まで試みる．

この手法で取得するのは隣接 PE の状態のみであるため，クラスタを構成する PE 数が増えた場合でも実行時間は変わらず，待機時間と再試行回数にのみ影響される．すなわち，この実行時判断手法は TCSH が提供する高いスケーラビリティを損なわない．

図 4 は，提案手法が過渡的に偏りが増大する移動順序を避ける状況の例である．初期状態において，TCSH は A,B が C,D よりも負荷が高い状態を解消するべく，plan の枠内に示されたような移動計画を立案した．この計画には，図 2 に示したような過渡的に偏りが増大する移動順序の可能性がある．提案手法は移動元と移動先の負荷を比較し，移動後に移動先 PE の負荷が移動元 PE の負荷を上回るような移動を後回しにする．time 40–41 における A,B 間，C,D 間の移動は移動後に移動先 PE の負荷が移動元 PE の負荷を上回るため，後回しにされる．結果として，図 2 に示した過渡的に偏りが増大する移動順序が避けられ，偏り除去過程における性能低下は免れる．

図 5 は，提案手法が移動実行時の状況に起因する偏り増大を避ける例である．初期状態において，TCSH は A,B が C,D よりも負荷が高い状態を解消するべく，plan の枠内に示されたような移動計画を立案した．この計画には，図 3 に示したような移動実行時の状況に起因する偏り増大の可能性がある．初期状態において，A,B 間，B,C 間，C,D 間のオブジェクト移動が計画されているが，いずれの移動も，移動後に移動先 PE の負荷が移動元 PE の負荷を上回るため，後回しにされる．結果として，図 3 に示した過渡的に偏りが増大する移動順序が避けられ，偏り除去による性能低下は免れる．

3. 評 価 実 験

本節では，我々が提案しているマルチメディアコンテンツサーバ（図 6）[13] 上における，前述の実行時判断手法の評価実験を行う．マルチメディアコンテンツサーバは動画データを中心としたコンテンツ群を配信する Web サーバであり，大容量データである動画データを格納するために配置変更などのコンテンツ管理の粒度が粗い特徴がある．本実験で用いるマルチメディアコンテンツサーバは，PC 上の模擬自律ディスクを利用して構成されている．模擬自律ディスクは，我々が提案している自律ディスク [12] を PC 上に Java™ を用いて実装したものである．我々はハードディスクのプロセッサとキャッシュメモリを用いた分散データ格納環境として自律ディスクを提案しており，その機能検証を PC 上に実装した模擬自律ディスクを用いて行っている．自律ディスクは改変可能なルールとその解釈機構を備えることによって，コンテンツの探索，負荷分散のための移動，障害回復処理などを各 PE が自律的に行う．クライアントとの通信は HTTP を用いて行う [15]．自律ディスクは接続されたネットワーク上でクラスタを構成し，マルチメディア

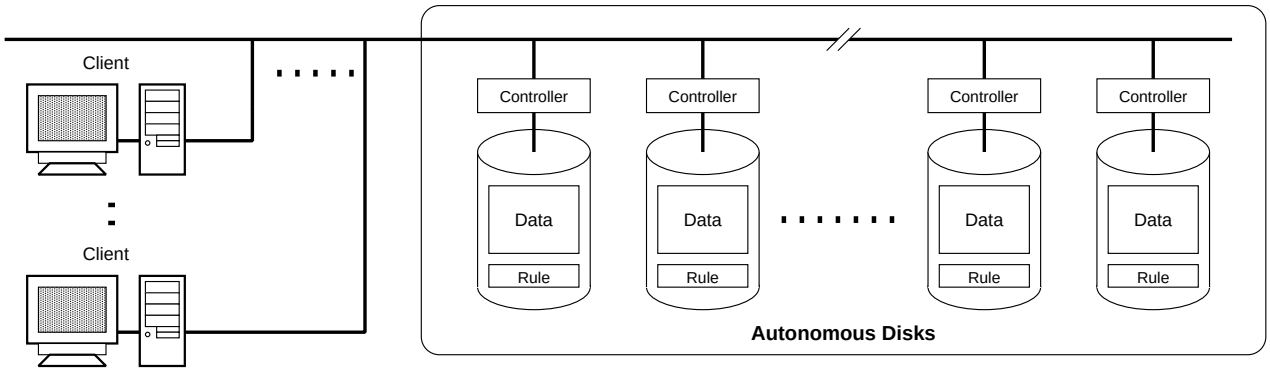

図 7　自律ディスク

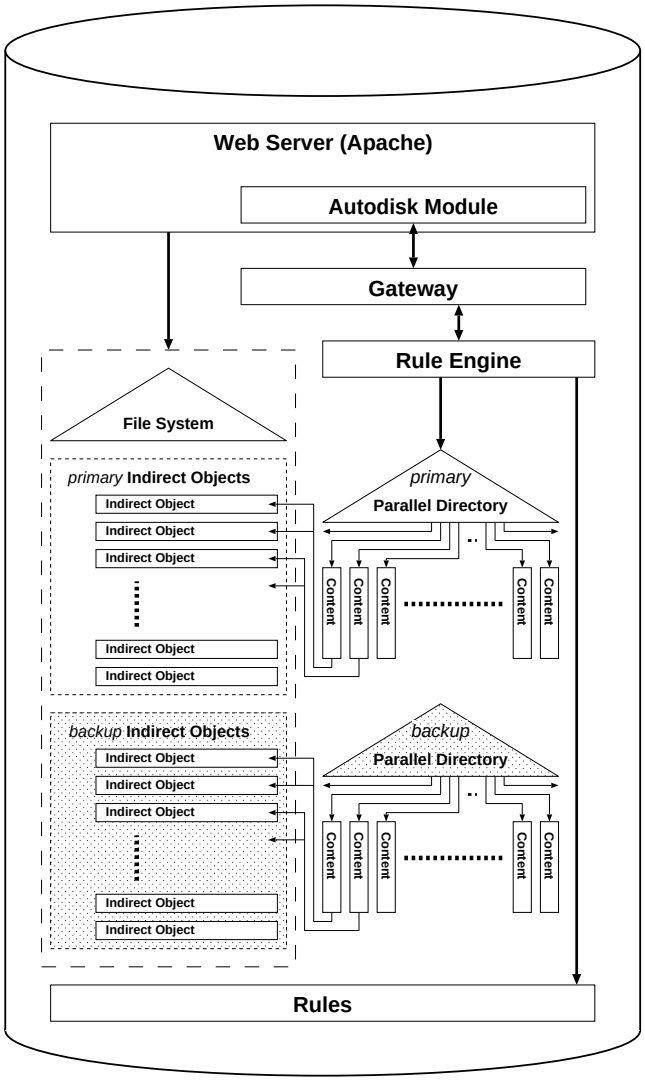

図 6　マルチメディアコンテンツサーバ構成図

コンテンツを提供する（図 7）.

評価実験では，PE 数 4 のマルチメディアコンテンツサーバに 50MB の動画ファイル 1 つと 5KB の静止画ファイル 5 つの組からなるコンテンツを 40 組登録した後に，コンテンツ毎のアクセス頻度に偏りのあるクエリ系列を与え，コンテンツ単位の移動による偏り除去を行いながら各 PE の状態を観測した．本実験で用いるマルチメディアコンテンツサーバには，コンテンツの格納位置と負荷を表示する機能（状態表示機能）が備えられており（図 8，中央上に並んだ小型ディスプレイに表示），我々はこの状態表示機能とアクセスログの解析によって，コンテンツの移動と各 PE における時間当データアクセス量の推移を計測した．

図 8　マルチメディアコンテンツサーバ

評価実験では，従来手法および提案手法のそれぞれの実行時判断手法において，速度係数 α の算出方法をクラスタ再構築を考慮した実行時判断手法 RSH のものに揃え，本稿で提案した移動先と移動元の PE のみからなる小さな領域を用いた管理の導入による効果を見た．評価実験の諸項を表 1 に示す．

図 9 は移動先の負荷を考慮しない実行時判断手法を用いた場合，図 10 は移動先の負荷を考慮する提案手法を用いた場合の推移である．図中の縦線は，偏り除去を開始した時点を示している．また，格納されているコンテンツの値域は PE の自然な順序付けにしたがって隣接している．実験では PE0 に負荷が集中するようなクエリ系列を用いた．そのため，配置決定戦略は負荷を均衡化するために（PE0→PE1，PE1→PE2，PE2→PE3）のようなデータの移動を要求する．

図 9 では，300（sec）～800（sec）の区間で PE0 のデータアクセス量が漸減し，代わりに PE1 のデータアクセス量が上昇している．これは，図 2 で示したような移動順序による過渡的偏り増大の結果と考えられる．すなわち，PE0→PE1 の移動が先

表 1　評価実験の諸項

項目	値
システム構成	
HDD シーケンシャルアクセス速度	22MB/sec
CPU	Intel Pentinum (R)III 933MHz
PE 数	4
メッセージ転送の準備時間	200μs/メッセージ
ネットワーク	1000Base-SX
Java^{TM} 実行環境	Sun JRE 1.4.1 Server VM
データベース	
インデクスノードのサイズ	4KB
コンテンツのサイズ	≅50MB
コンテンツ数	40
クエリ	
クエリ数	1000
平均到着間隔	0.75sec
コンテンツのアクセス偏り	zipf(1)
初期状態における hotspot	左端
hotspot のアクセス頻度	400%
偏り除去	
負荷評価手法	アクセス量による評価 [13]
並列制御手法	TCSH [9]
偏り除去頻度	30sec 毎
実行時判断手法	RSH [11], 提案手法
偏り制御の閾値 ζ	0.25
許容される過渡的偏り τ	2MB/sec
速度係数の上限 χ	1
過剰負荷の影響見積もり γ	10
再試行までの待機時間 i	-
再試行回数 r_m	0

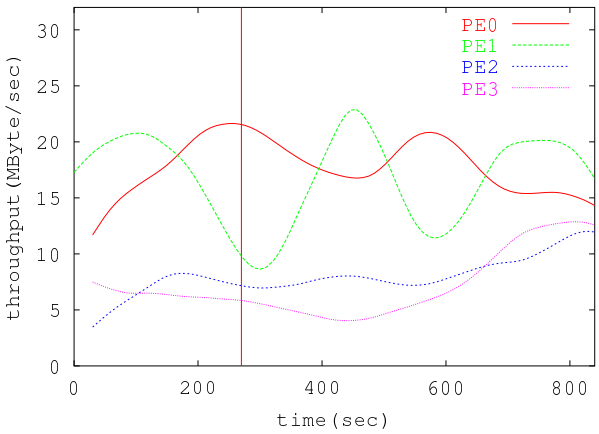

図 9　従来手法を用いた場合のデータアクセス量の推移

に実行されたために，PE1 の負荷が上昇したものと考えられる．前述のコンテンツの格納位置と負荷を表示する機能を用いた観測においても，PE0 に格納されていたコンテンツが PE1 に移動し，PE1 の負荷が上昇していく様が確認された．

一方，図 10 では，PE1 の負荷は上昇せず，700（sec）の時点まで漸減し続けている．これは，PE0 と PE1 の負荷が逆転するような移動が抑制された結果である．そのため，300（sec）近傍の早い段階から PE2 および PE3 へ負荷が移動し，負荷が速

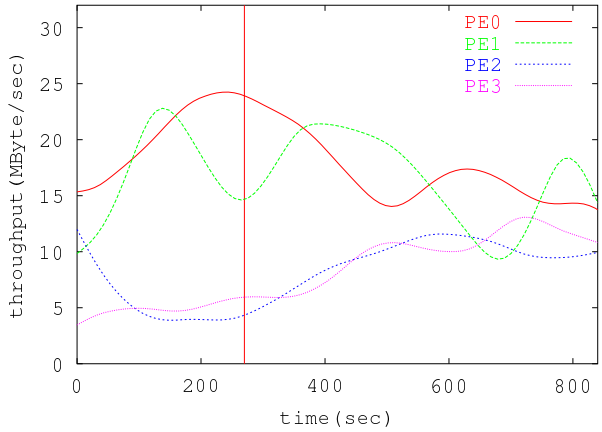

図 10　提案手法を用いた場合のデータアクセス量の推移

やかに均衡している．コンテンツの格納位置と負荷を表示する機能を用いた観測においても，PE1 に格納されていたコンテンツの PE2 への移動と，PE0 に格納されていたコンテンツの PE1 への移動がバランスよく進行している様子が確認された．

4. 結　　論

本稿では，偏り除去の過程において，実際の移動が実行される順序に起因する過渡的偏り増大および移動実行時の状況に起因する偏り増大が生じうることを指摘し，これらの偏り増大を回避する分割境界移動についての実行時判断手法を提案した．提案手法は，移動先と移動元の PE のみからなる小さな領域を用いた管理を導入することで，負荷が小さい PE から負荷が大きい PE へとデータオブジェクトの移動を抑制し，これらの偏り増大を軽減する．我々は自律ディスクを用いたマルチメディアコンテンツサーバによる評価実験によって，これらの効果を確認した．今後は多様な条件下での検証実験によって，実行時判断手法と偏り除去との関連を調べる．

謝　　辞

本研究の一部は，文部科学省科学研究費補助金基盤研究（14019035）および情報ストレージ研究推進機構（SRC）の助成により行なわれた。

文　　献

[1] C. Xu and F. C. M. Lau: “Load Balancing in Parallel Computers”, Kluwer Academic Publishers (1997).

[2] K. A. Hua and C. Lee: “Handling Data Skew in Multiprocessor Database Computers Using Partition Tuning”, Proc. of VLDB ’91, pp. 525–535 (1991).

[3] C. B. Walton, A. G. Dale and R. M. Jenevein: “A Taxonomy and Performance Model of Data Skew Effects in Parallel Join”, Proc. of 17th Int’l. Conf. on VLDB (1991).

[4] H. Lu and K.-L. tan: “Dynamic and Load-Balanced Task-Oriented Database Query Processing in Parallel Systems”, Proc. of Third EDBT, pp. 357–372 (1992).

[5] K. A. Hua and J. X. W. Su: “Dynamic Load Balancing in Very Large Shared-Nothing Hypercube Database Computers”, IEEE Transactions on Computer, **42**, 12, pp. 1425–1439 (1993).

[6] D. DeWitt and J. Gray: “Parallel Database Systems: The Future of High Performance Database Systems”, Communications of the ACM, **35**, 6, pp. 85–98 (1992).

[7] B. Seeger and P. Larson: “Multi-Disk B-trees”, Proc. of ACM SIGMOD Conf. ’91, pp. 436–445 (1991).
[8] H. YOKOTA, Y. KANEMASA and J. MIYAZAKI: “Fat-Btree: An Update-Conscious Parallel Directory Structure”, Proc. of 15th Int’l Conf. on Data Engineering, pp. 448–457 (1999).
[9] 渡邊, 横田：“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, **J85-D-I**, 9, pp. 877–886 (2002).
[10] 渡邊, 横田：“並列データアクセス偏り制御におけるスケーラブルな並列制御”, DEWS 2002 (2002). C1-3, http://www.ieice.org/iss/de/DEWS/proc/2002/index.html.
[11] 渡邊, 横田：“分散ディレクトリ偏り制御とシステム再構成を統合する再配置制御”, DBSJ Letters, **1**, 1, pp. 3–6 (2002).
[12] H. Yokota: “Autonomous Disks for Advanced Database Applications”, Proc. of International Symposium on Database Applications in Non-Traditional Environments (DANTE’99), pp. 441–448 (1999).
[13] 渡邊, 花井, 山口, 横田：“自律ディスクを用いたマルチメディアコンテンツサーバ”, Technical Report DE2002-86, DC2002-22, 電子情報通信学会 (2002).
[14] H. Feelifl, M. Kitsuregawa and B.-C. Ooi: “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 (2000).
[15] 花井, 横田：“自律ディスクを用いた Web サーバーの構成”, DEWS 2002 (2002).