負荷均衡化とシステム再構成を統合する データ移動制御手法

A Data Migration Control Method to Unify Load Balancing and System Reconstruction

渡邊 明嗣♥ 横田 治夫♦

Akitsugu WATANABE Haruo YOKOTA

スケーラブルな分散データ格納システムを構築するためには，クラスタ構成の再構築手続きと負荷偏り除去が重要である．これらの手続きはいずれもデータの再配置を伴うため，データ移動要求の衝突を回避する機構が必要である．クラスタ再構成中の状態を負荷が偏った状況と捉えることで，これらの移動手続きを負荷偏り除去手続きの中に統合できる．再構成に伴う負荷の偏りと要求される移動量は定常状態のそれに比べて大きく，短時間での処理が困難である．本稿では，クラスタ再構成に伴う大きな偏りと定常状態の小さな偏りの両方を効率的に除去できる偏り除去の並列制御機構と分割数決定機構を提案し，その性能を評価する．

Cluster reorganization and skew handling procedure take important place to constructuring scalable distributed data storage systems. Because both procedures involve data reorganization, we should adjust migration requirements of these procedures. By dealing state under cluster reconstruction with highly load skewed situation, data migration by cluster reconstruction is unified to skew handling procedure. However, since skews generated by cluster reconstruction are much higher than usual skews, it is hard to remove it quickly. In this paper, we develop our parallel control method for skew handling and propose a high skew adaptive splitting method for migration. The experiment indicates that the proposed method can treat both ordinary skews and highly skews generated by cluster reconstruction.

1. はじめに

並列無共有システムは，対費用効果，耐故障性，および，拡張性が優れていることから，大規模データの格納に適した構成として注目されている．並列無共有システムでは，独立したディスクおよびメモリを持つ多数のノードが共有のネットワークに接続されており，各ノードは唯一の共有資源であるこのネットワークを利用してメッセージ交換を行う．

我々はストレージ主導型の分散データ格納アーキティクチャとして自律ディスク[5](図1) ryouを提案している．自律ディスクは自律的管理を行う大規模ストレージのための構成手法であり，現在ディスクコントローラとしてディスクに搭載されているPU を利用することで，高度なデータ管理を安価に実現することを狙っている．

自律ディスクの目的は可用性，拡張性が高い分散ストレージクラスタの実現である．我々はそのために重要な機能的特長として，(a) 透過的アクセス，(b) 自律負荷分散機構，(c) オンライン再構築機構，(d) 障害回復，を挙げている．

(a) 透過的アクセスを実現する手段の一つに我々が提案している分散ディレクトリ構造Fat-Btree があり，プロトタイプが自律ディスクに実装されている．

高い拡張性を持つデータ格納システムを並列無共有システム上に構成するためには，効率的なデータ再配置が重要である．データ再配置には(1) ノードの追加・削除に伴うもの，(2) ノード間の負荷を均衡化するためのもの，の2 つがある．これら2 つが協調しながら並行に実行されることは，高い可用性のために重要である．また，データ再配置は分散ディレクトリ構造を利用するため，用いる分散ディレクトリ構造の特徴を考慮した設計が不可欠である．

我々は自律ディスクにおけるディスククラスタのオンライン再構築手法を提案しており[3]，サービス停止を必要としないディスククラスタの再構築が可能である．しかし，この再構築手法はノード間の領域使用量を平均化する負荷分散機構[6] の利用を前提としており，そのため，[1，2] によって指摘されているオブジェクト間にアクセス偏りがあるような状況での負荷分散における問題を残している．一方で，従来のオブジェクト間にアクセス偏りがある状況を考慮した負荷分散機構[2，4] ではディスククラスタ再構築を考慮していないため，ノードの追加に伴う大量のデータ移動を速やかに解決できない問題があった．

本研究ではディスククラスタ再構築を考慮した負荷分散機構についての考察を行い，ディスクの追加に伴う負荷の偏りを効率良く解消できる負荷分散機構の構成案を示す． また，提案構成案についてシミュレーションを用いた評価とその考察を行う．

2 章ではクラスタの再構築を考慮した負荷分散機構の構成案を示す．3 章ではシミュレーションを用いた提案構成案の評価とその考察を行う．4 章では本稿の結論を述べる．

2. クラスタ再構築を考慮した負荷分散機構

オンライン並列負荷分散機構は並列制御機構，負荷評価機構， 移動速度決定機構の 3 つの要素から構成される．負荷評価機構は，ノード内負荷評価機構と部分木負荷評価機構によって構成される．

2.1 並列制御

我々は実行時間がO(log(ノード数)) であるスケーラブルな負荷分散の並列制御TCSH(Tree Communication Skew Handling parallel control) を提案している[4]．しかしながらこの提案では，クラスタを構成するノードが均質であることと，クラスタ構成が変化しないことを仮定しているため，クラスタ再構築を扱うためには若干の変更が必要である．すなわち，各ノードの負荷情報だけではなく，各ノードの(均質でない) 処理能力をも集計する必要がある．

図2 はクラスタ再構築を考慮した負荷分散機構の並列制御TCSH の改良案である．TCSH は以下の5 つの実行フェーズを経て，クラスタの負荷を平均化する．

1．通信木の構築

コーディネーターとなるノードが負荷分散を管理するコ

♥ 非会員 東京工業大学 大学院 情報理工学研究科 計算工学専攻 博士後期課程 aki@de.cs.titech.ac.jp

♦ 正会員 東京工業大学 学術国際情報センター 教授 yokota@cs.titech.ac.jp

ーディネータープロセスを作り，通信木を構築する．通信木の頂点はプロセスに，辺はプロセス間通信路に対応する．コーディネータープロセスを根とするバランス2 分木で，各ノードと1 対1 対応したプロセスである葉を持つ．

2．ノード内負荷の評価およびその伝達

葉でノード内負荷評価を行い，結果とノードの処理能力を通信木の枝を上りながら合計する．

3．負荷分布の伝達

各枝は，子から受け取った負荷情報を元に子の処理能力に応じた負荷分担と子の間での負荷移動量を決定する．子は，親から負荷分担と負荷移動量を受取り，それを自らの子に分担させる．

4．移動速度の決定

過剰な移動を抑制するために，移動速度決定機構はフェーズ3 で受け取った負荷移動量に移動速度係数を掛け，移動をより小さな単位に分割する．

5．部分木負荷評価機構による移動する部分木の決定と移動

フェーズ4 で決定された負荷移動量を基に，部分木負荷評価機構が隣接ノードに移動する部分木を決定し，移動する．

フェーズ 1, 2, 3 は通信木に沿って実行されるため，その実行時間のオーダーは O(log(ノード数)) である．フェーズ 4,5 は各ノードで独立に実行されるため、その実行時間のオーダーは O(1) である．したがって，TCSH の実行時間は O(log(ノード数)) である．

近年におけるネットワーク，ディスク，プロセッサ 3 者の速度向上の現状から，ディスクアクセスはボトルネック発生原因の最も有力な候補である．ディスクアクセスに注目したシステム負荷の評価は熱と呼ばれる指標に反映される[1]．我々は分散ディレクトリが持つインデクスノードの冗長なコピーによってもたらされる負荷分散効果と，キャッシュによる負荷軽減効果を考慮することで，ノード内負荷の評価精度を高める負荷評価機構 DTC(Directory Traverse Cost based load evaluation) を提案している[4]．

我々は記録するアクセス履歴の粒度による DTC のバリエーション DTC n を用意している．DTC n は，インデクス木の根からの距離が n 以下のインデクスノードについてのみアクセス履歴を記録し，その先では負荷が均等に分配されていることを仮定する．細かい粒度は負荷評価の精度を高め，正確な偏り除去を可能にするが，アクセス履歴の記録コストが大きい．粗い粒度はアクセス履歴の記録コストが小さいが，負荷評価の精度において劣る．

我々が[4] で行ったシミュレーションでは，n = 1 のインデクスノードのアクセス履歴を用いた評価の場合に，記録コストに対して得られる精度向上とその利益が最も大きくなった．

2.3 移動速度決定機構

負荷分散の過程では，一度移動したデータを元の場所に戻すような移動がしばしば起こる．このような"無駄な移動" は負荷評価の誤差や負荷分布の変化などによって起こるもので，システムに余分な負荷を与え，性能を低下させる．このような無駄な移動を抑制には，負荷の移動量に速度係数を掛けてデータの移動を分割するアプローチが有効である．

Feelifl らはデータ移動量の閾値と速度係数を用いて無駄な移動を抑制するアプローチを提案している[2]．このアプローチでは，負荷の移動量のノード内負荷に対する比が閾値ζ以下であるような移動を抑制し，また，負荷の移動量に速度係数α(0 < α< 1) を掛けてデータ移動を複数回に分割する．次式は，このアプローチにおける移動予定の熱と移動量に掛けられる係数fk の関係を示したものである．

$$\mathrm{fk}_{a,z}(h,a)=\begin{cases}0 & (h/a<z)\\ a & (h/a\geq z)\end{cases}$$

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．関数fk は移動予定の熱h をh = h×fk(h,a) のように正規化する．

すなわち，fk = α= 0 の時は一切の移動を行わず，fk = α= 0.5の場合には観測された偏りの半分に相当するページを移動し，fk = α= 1 の場合には観測された偏りの全部に相当するページを移動する．

このアプローチは定常状態における小さな偏りの除去に伴う無駄な移動を効果的に抑制する反面，非常に大きな偏りの除去が遅くなる欠点を持ち，クラスタ再構築に伴うノードの追加/削除によってもたらされる非常に大きな偏りを効率的に除去することが難しい．大きな偏りを速やかに除去しようとしてαを大きな値に設定すると定常状態における無駄な移動が増え，定常状態の無駄な移動の抑制を期待してαを小さな値に設定すると大きな偏りの除去に対応できない．

この問題は，システムに与える影響の度合いが異なる大きな偏りと小さな偏りを同じ速度係数で取り扱うことから生じている．速度係数を定数ではなく偏りの大きさに応じて変化する値とすることで，大きさの異なる偏りを効率よく扱うことができる．我々は負荷偏りによるシステムの性能低下率がシステムの過剰負荷率に対応していると予想し，次のような移動速度決定関数rsh(Reconstruction consious Skew Handling) を用意した．

$$\mathrm{rsh}_{l,c,z}(h,a)=\begin{cases}0 & (h/a<z)\\ l(h/a) & (z\leq h/a<c/l)\\ c & (c/l\leq h/a)\end{cases}$$

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．関数rsh は移動予定の熱h をh = h×rsh(h,a)のように正規化する．この関数は，過剰負荷率を用いてクラスタ再構築などによって高い移動速度が必要とされる状況と，無駄な移動を抑えるために移動速度を低く保つべき状況とを区別し，その中間的な領域を滑らかに連結する．定数ζは無駄な移動を抑えるために移動を行わない領域の上界を決定し，定数λは，この中間的な領域の幅とその領域における移動速度の増加率を決定する．

この関数はクラスタの再構築を負荷分散機構で取り扱うという本稿の目的を満たすものである．我々は次章で，この関数のシミュレーションによる評価を行う．

3. シミュレーションによる評価

この実験では，2.3 で提案した移動速度決定関数を用いた負荷偏り除去を行い，ノード追加時の負荷偏り除去速度と，定常状態での"無駄な移動" 量を測定する．比較のため，同時に[2] の手法を用いた負荷偏り除去も行い，結果を示す．

以下の実験では，表 2 に示したパラメータを用いる．偏り除去の並列制御には本稿で提案した TCSH の改良案(図 2) を用いる．ノード内負荷評価戦略 GLE(Global Load Evaluation) として，一定時間内のデータアクセスに要した時間の総計を用いる．部分木負荷評価戦略 LLE(Local Load Evaluation) として，前出の DTC 1[4] を用いる．移動速度

決定関数として，$rsh_{2,1,1/32}$ を用いる．rsh の母数λ= 2 は偏りの無いクラスタにノードが追加された場合の負荷偏りの比率　0.5 における移動速度を最大にすることを狙って選択した．　χ= 1 は移動速度が 1 を越えないという[2] の制約を踏襲した選択である．　ζ= 1/32 は，偏りの無いノード数 15 のクラスタに新しいノードを追加した場合の負荷偏り 1/15　1/16 を基に，それに 2 倍のマージンを持たせるよう選択した．

ホットスポットになる可能性があるノードへの負荷の集中の度合いを示すため，最大負荷，すなわち最も負荷の大きいノードの負荷の平均に対する比を用いる．例えば，最大負荷 200%は，少なくとも 1 つのディスクに，負荷が完全に分散されている場合に比べて倍に相当する負荷が与えられていることを表す．大きな最大負荷は負荷が集中したノードの存在と，そのノードがボトルネックになることによる性能低下の可能性を示唆する．

図 3 はノード追加時および定常状態での最大負荷率の偏り除去に伴う推移を示したものである．提案手法は，ノード追加時においては中程度の移動速度であり，ノード追加に対する偏りの収束速度と収束結果に優れた $fk_{1/2,1/32}$ に近い推移を示す．また，定常状態では，移動速度が大きく，無駄な移動が行われにくい $fk_{1/8,1/32}$ とほぼ同等の安定した推移を示す．表 1 に定常状態における無駄な移動の量を示した．結果は $fk_{\alpha,1/32}$ はαの値が大きい場合に多くの無駄な移動を行うとの予想に一致しており，α= 1/1 の場合の無駄な移動は平均で 500 ページ程度でデータの移動時間にして約 5 秒に相当する．提案手法は無駄な移動が少なく，負荷分散機構がシステムに与える負荷が小さい．

4. 結論

本稿では，負荷分散機構がクラスタ再構築におけるデータ移動手段として利用されることを前提に，クラスタ再構築におけるデータ移動において有効な特長を有する負荷分散機構の提案を行った．さらに，従来の並列制御機構を拡張した並列制御機構を提案した．また，クラスタ再構築において有効な移動速度決定機構の案 $rsh_{\lambda,\chi,\zeta}$ を示した．$rsh_{\lambda,\chi,\zeta}$ ではノード追加時のような高速に均衡化を図るべき状況と安定した定常的な状況のそれぞれに合わせた移動速度を設定できる．さらに，定数λを導入することで，その中間的な状況でも適宜移動速度を調整するとともに，その領域の幅もシステムの構成に応じて変更することができる．

我々は提案移動速度決定機構の性能をシミュレーションによって評価し，それが従来手法では両立できなかった定常状態での安定した負荷分散と，ノード追加から定常状態への移行時間の短縮を両立していることを示した．我々が提案した負荷分散機構はより単純な機能部品の組み合わせによって構成されており，システムの状況に合わせた調整が可能であることから，様々な構成の分散データ格納システムへの応用が期待できる．また，提案負荷分散機構の実行時間は O(log(ディスク数)) であり，クラスタの自律再構成との組合わせによって高いスケーラビリティを持つシステムの構築が可能である．ノードの取り外しを含めた提案手法の評価及び適切な移動速度決定の手法は将来の課題である．

[謝辞]

本研究の一部は，文部科学省科学研究費補助金基礎研究(12680333,13224036,14019035) および情報ストレージ研究推進機構(SRC) の助成により行われた．

[文献]

[1] G. Copeland, W. Alexander, E. Boughter, and T. Keller. Data Placement in Bubba. In Proc. of ACM SIGMOD Conf. '88, pp.99-108, 1988.

[2] Hisham Feelifl, Masaru Kitsuregawa, and Beng-Chin Ooi. A fast convergence technique for online heat-balancing of btree indexed database over shared-nothing parallel systems. In 11th Int'l Conf. on Database and Expert Systems Applications, Sep 2000.

[3] 伊藤大輔, 横田治夫. 自律ディスクにおけるクラスタ再構築アルゴリズムの実装. 信学技報，電子情報通信学会，FTS2001-20.電子情報通信学会, July 2001.

[4] WATANABE, Akitsugu and YOKOTA, Haruo. A Directory-Traverse-Cost Based Skew Handling for Parallel Data Access.IEICE Trans. Inf. & Syst. (Japanese Edition), 2002. in press.

[5] Haruo Yokota. Autonomous Disks for Advanced Database Applications. In Proc. of International Symposium on Database Applications in Non-Traditional Environments (DANTE'99), pp.441-448, Nov. 1999.

[6] Haruo YOKOTA, Yasuhiko KANEMASA, and Jun MIYAZAKI. Fat-Btree: An Update-Conscious Parallel Directory Structure. In Proc. of 15th Int'l Conf. on Data Engineering, pp. 448-457, 1999.

渡邊明嗣　Akitsugu WATANABE
東京工業大学大学院情報理工学研究科計算工学専攻博士後期課程在学中．2002 東京工業大学大学院情報理工学研究科計算工学専攻博士前期課程修了．分散ディレクトリの研究に従事．

横田治夫　Haruo YOKOTA
昭 55 東工大・工・電物卒．昭 57 同大大学院・情報・修士課程了．同年富士通(株) 入社．同年 6 月(財) 新世代コンピュータ技術開発機構研究所．昭 61(株) 富士通研究所勤務．平 4 北陸先端大・情報・助教授．平 10 東工大・情報理工・助教授．平 13 東工大・学術国際情報センター・教授．工博．主としてデータベース，データ工学向けの並列アーキティクチャ等に関する研究に従事．情報処理学会，人工知能学会，IEEE，ACM 各会員．

表 1 定常状態における，"無駄な移動" が行われたページ数の平均

[2] a= 1/8, ζ= 1/32	35.77±24.08
提案手法	37.37±28.92
[2] a= 1/4, ζ= 1/32	86.45±49.04
[2] a= 1/2, ζ= 1/32	209.31±103.08
[2] a= 1/1, ζ= 1/32	526.29±259.71

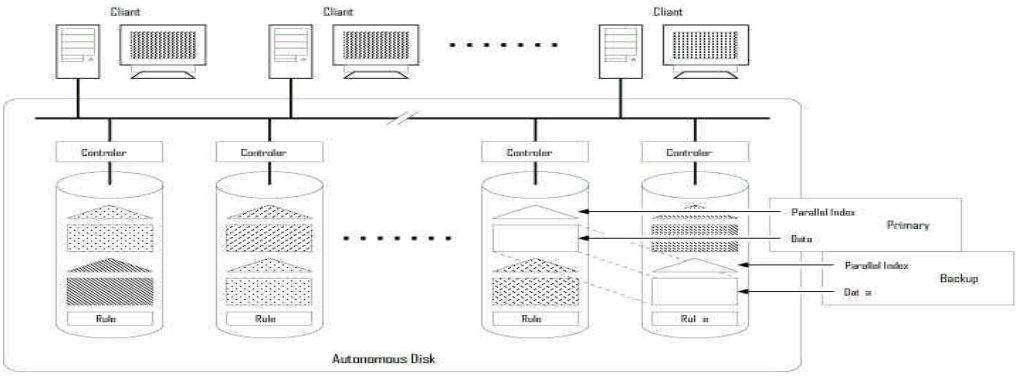

図 1 自律ディスク

```
Start TCSH() {
    p はこのプロセスを表す;
    D はクラスタに属するノードの全体集合を表す;
    任意のノードにプロセス TCSH() を作り， D, p を送る;
    作成した TCSH() から負荷 h, 処理能力 a を受信する;
    作成した TCSH() に負荷割当 h, 0, 0 を送信する;
}
TCSH() {
    p はこのプロセスを表す;
    親プロセス pp から，ノード集合 S, pp を受信する;
    if(|S| >1) {
        S を左集合 L と右集合 R に等分する;
        L,R それぞれから任意のノード l pe, rpe を選ぶ;
        /* フェーズ 1 */
        l pe に TCSH() プロセス l p を作り L, p を送る;
        rpe に TCSH() プロセス rp を作り R, p を送る;
        /* フェーズ 2 */
        l p から負荷 lh, 処理能力 la を受信する;
        rp から負荷 rh, 処理能力 ra を受信する;
        pp に(lh+rh),(la+ra) を送る;
        /* フェーズ 3 */
        負荷割当 ah, 左/右へ移動する負荷 tlh,trh を pp から受信する;
        L への負荷割当 lah = ah×la/(la+ra));
        R への負荷割当 rah = ah×ra/(la+ra));
        L に負荷割当 lah, 左/右に移動する負荷 tlh, ((lh.tlh).lah) を送信する;
        R に負荷割当 rah, 左/右に移動する負荷((rh.trh).rah), trh を送信する;
    } else {
        /* フェーズ 2 */
        ノード内負荷評価戦略GLE を用いてノードp の負荷を測定し，
        ch に代入する;
        ノード p の処理能力を測定し，ca に代入する;
        pp に ch,ca を送る;
        /* フェーズ 3 */
        負荷割当 ah, 左/右へ移動する負荷 tlh,trh を pp から受信する;
        /* フェーズ 4 */
        移動速度決定戦略 SF を用いて tlh と ca から
        今回移動する負荷 ctlh を決定する;
        trh,ctrh についても同様;
        /* フェーズ 5 */
        ctlh が正なら，部分木負荷評価戦略 LLE を用いて
        ctlh から移動する部分木の大きさを決定し，
        部分木を左側の隣接ノードに移動する;
        ctrh についても同様;
    }
}
```

図 2 クラスタ再構築を考慮した TCSH の改良案

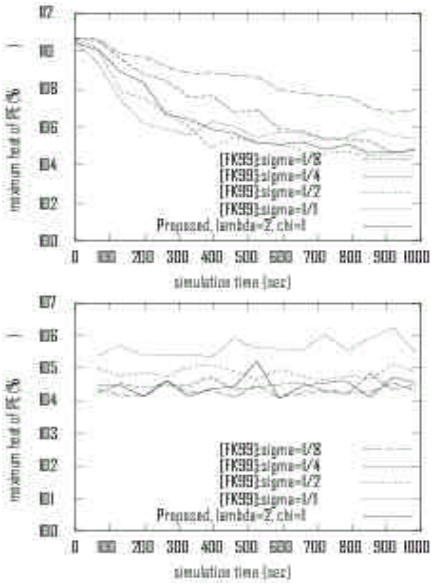

図 3 ノード追加時の負荷偏り除去過程(上) と定常状態での負荷偏り除去過程(下)

表 2 シミュレーションに用いたパラメータ

パラメータ	値
インデクスノードのサイズ	4KB
データページのサイズ	4KB
メッセージセットアップ時間	200 μs/メッセージ
ディスクのアクセス速度	10ms/4KB
メモリのアクセス速度	1 μs/4KB
ノード数	16
キャッシュ連想度	8
キャッシュセット数/ノード	128
データページ数	1M (4GB)
インデクスノード	
最大エントリ数	200
エントリ利用率	ln2
インデクス木の高さ	4
アクセス偏り(zipf 分布の母数)	0.5
クエリの到着間隔	2.0 ms
偏り除去の間隔	32K クエリ(1 分)