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

Access Distribution Control for Data Replication Based on Data Migration Cost and Caches Behaviors

小林 大♢ 渡邊 明嗣♢ 田口 亮♠
上原 年博♠ 横田 治夫♣

Dai KOBAYASHI Akitsugu WATANABE
Ryo TAGUCHI Toshihiro UEHARA
Haruo YOKOTA

本研究では負荷均衡化と要求性能保障の両立を目的とし，我々の提案する自律ディスクに対し複製データへのアクセス回送による負荷分散を導入する．各ストレージノード上のキャッシュを考慮した回送先アクセス対象の決定方法について考察しキャッシュヒット率をパラメータとした回送先決定手法を提案する．また，自律ディスクが従来から備えるデータ移動による各負荷均衡化手法と複製へのアクセス回送との切り替え・併用制御手法を提案し，自律ディスクにより構成されたシステムにおいて要求性能保障を達成する．

In this paper, we introduce data replication function to the Autonomous Disks System, toward realization both skew handling and performance guarantee. In first, we propose the method that uses a hit ratio of each cache memory to recognize the popularities of data. Additionally, we also propose a control method to coordinate replication and migration functions complementary.

1. はじめに

CPU やメモリ等を組み込んだ多数の高機能で安価なストレージノードを，ネットワーク結合することにより構成されたストレージシステムにおいて，ストレージノード間のアクセス負荷の均衡化は重要である．

データマイグレーションを行うことで，アクセス負荷均衡化が達成される [1]．これは，システム運用中に逐次計測している各ノードの負荷情報を利用し，動的にデータ再配置を行う手法である．しかしデータマイグレーションは，ディスクアクセスやネットワーク転送を伴い，ストレージ装置の性能を一時的に低下させることが問題となる．

また，複製データを用いたアクセス負荷均衡化が多数提案されている [2,3]．データの複製を複数の異なるノードに配置し，複製データへアクセスの一部を分配することで，ノード間の負荷分散を達成できる [2]．また，複数の複製へのアクセス分配割合を，動的に変更することで，ある範囲の負荷傾向変化へは性能低下なく適応可能である [3]．一方で，大きな負荷偏りの変化に適応するためには，動的な複製配置変更を行う必要があり，そのアクセスによる性能低下といった問題がある．

我々は，自律ディスクと呼ぶネットワークストレージシステムを提案している [4]．自律ディスクでは，データマイグレーションによる性能低下の問題があった．我々はこれまでに，データマイグレーションによるアクセス負荷を，システムに対する要求性能保持の制約と複製データへのアクセス回送の併用により分散させる手法を提案してきた．[5,6]

本稿では，さらにこの方式を拡張し，複製データへのアクセス分散の常用化とデータマイグレーションの発動制御により不必要なデータ移動を削減することを目的とする．以下では 2 つの提案を行う．

1 つは，複製による負荷分散能力を最大化するため，各ストレージノードのキャッシュヒット率を考慮したアクセス分配先決定方法について検討する．

2 つ目として複製へのアクセス分散とマイグレーションの切り替え・併用制御を提案する．提案手法では，変化する負荷傾向を元に切り替え閾値を逐次求めることで，マイグレーション頻度の減少を試みる．またマイグレーションによる性能への影響が少ないデータ移動経路の選択を行う．さらに，マイグレーションによる増加負荷量を複製へのアクセスの分配率算出に利用し，負荷をより多くのノードに分散する．

本稿の構成を以下に示す．つづく 2. においてキャッシュを考慮したアクセス分配先判定手法を提案し，有効性を論ずる．3. で自律ディスクにおける複製へのアクセス回送とマイグレーションの切り替え・併用制御法について述べ, 4. にて自律ディスクシミュレーションに提案手法を実装し，その効果を確認する．最後に 5. にてまとめと今後の課題を述べる．

2. アクセス分配判定手法

高機能ストレージや，PC クラスタでは各ストレージノードは独立の大容量キャッシュ（バッファ）メモリを有している．アクセス分配により各ノード内のデータ利用傾向が変化するため，キャッシュの振る舞いもそれに伴い変化する．よってその利用効率も考慮する必要がある．

本節では，キャッシュヒット率の高いアクセス分配判定手法を提案する．アクセス分配判定手法とは，あるノードに対するアクセスのうち，負荷均衡化戦略により与えられた分配するアクセスの率（以下，アクセス分配率）を満たす量のアクセス部分集合を決定するための手法とする．

2.1 キャッシュヒットを利用したアクセス分配決定

我々は，各ノード上キャッシュのヒット率情報を利用したアクセス分配判定手法を提案する．本手法は，キャッシュヒット率と関連の高い，各データのアクセス頻度情報を利用してアクセス分配を判定する．アクセス頻度情報を低コストで取得するために，各ノード上のキャッシュヒット情報を利用する．

本手法ではキャッシュミスしたアクセスを優先的に複製側へ分配する．アクセス分配率がキャッシュミス率を上回るとき，キャッシュヒットしたリクエストを確率的に分配する．高頻度利用データへのアクセスを優先的に複製側へ分配すると，プライマリ側ではアクセスの局所性が失われ，複製側では高頻度利用データが限られたキャッシュ資源を奪いあう．一方，低頻度利用データへのアクセスを優先的に複製側へ分配する場合，高頻度利用データが 2 つのノードへ分散し，キャッシュ資源を効率よく扱えるためである．

以下では提案手法をキャッシュ h/m(Hit or Miss) 分岐方式と呼ぶ．観測している最近のヒット率を h, 設定された回送率を r とすると，キャッシュ h/m 分岐方式は以下のように判定を行う．

1. あるリクエストがプライマリ側にきたら，まずプライマリ側のディスクのキャッシュをサーチする．
2. プライマリデータ格納ノードのキャッシュにおけるヒット/ミスを見る．ここで，ヒットあるいはミスしたとしても，キャッシュ内容への変化は施さない．この STEP のアクセスをヒット率の計算に加える．
3. 設定された分配に従い，以下の式により決定する．
   (a) ミス率 $(1-h)$ が分配率 r を上回るとき: ミスしたアクセ

♢ 学生会員 東京工業大学 大学院 情報理工学研究科 計算工学専攻
{daik,aki}@de.cs.titech.ac.jp
♠ NHK 放送技術研究所 {taguchi.r-cs,uehara.t-jy}@nhk.or.jp
♣ 正会員 東京工業大学 学術国際情報センター yokota@cs.titech.ac.jp

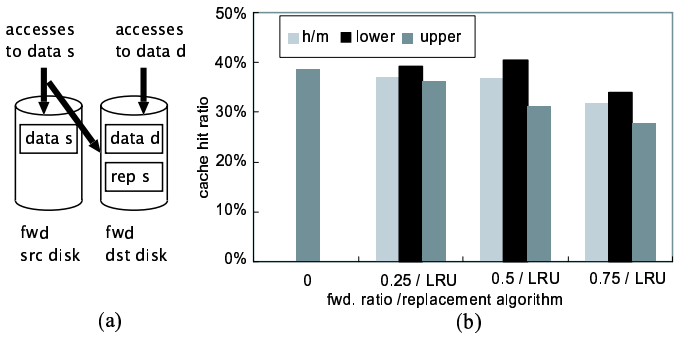

図 1: 実験構成および実験結果: (a) 実験構成, (b) 提案手法によるキャッシュヒット率の変化

Fig.1 Experimental configulations and results: (a) A two-node system, (b) Cache hit ratio with proposed method

スのうち $r/(1-h)$ を分配

(b) ミス率が分配率を下回るとき: ミスしたアクセスは全て分配する．ヒットしたアクセスのうち $(r+h-1)/h$ も分配

4. アクセスが複製側へ分配されない場合，改めてキャッシュを操作する．この STEP のアクセスもまたヒット率計算に加える．

以上により，r によって指定された率のアクセスを複製側へ分配することができる．

2.2 シミュレーション

本提案手法について，ヒット率とオーバヘッドに関し考察を行う．

キャッシュヒット率について，以下のシミュレーションを行った．アクセス系列としては，我々の研究室の WEB サーバ[1]のログから採取したパターンを用意した．アクセス数は年間 2 万件程度で，アクセス傾向は四半期毎程度に変化しており，一部のよく使われるファイルへのものである zipf 分布に近い．

実験の仮定として，単純な 2 ノード構成とした．プライマリノードには他のノードからリクエストが分配されることはなく，複製保持ノードには複製へのアクセスとそのノードが持つプライマリデータへのアクセスの両方が到達することとした．(図 1(a))．格納されるデータの間に強い関連がなければ，ノードは負荷の高いノードから始まる少数個のノードの鎖状部分集合に分かれる．その中間ノードのアクセス傾向はあまり変化しないと考え，今回はその一番単純な形とした．

各ノード内データ総数は 15Mpage, キャッシュ容量は 32Kpage とし，キャッシュアルゴリズムは Set-assosiative LRU とした．分配率と，2 ノード間アクセス量比率を変化させ計測した．

観測結果を図 1(b) に示す．h/m で示すグラフが提案手法のものであり，lower は低頻度アクセス優先分配の，upper は高頻度アクセス優先分配の理論値である．これより，本手法が upper と lower の中間程度のヒット率を示せることを確認した．

キャッシュ走査オーバヘッドについて考える．本手法では，1 アクセスあたり 2 回キャッシュ走査を行うため，キャッシュ走査オーバヘッドは 2 倍に増える．単純な LRU の $O(n)$ や $O(\log n)$ のアルゴリズムでは，同等のオーバヘッドを得るためにはアクセス粒度を大きくする必要がありヒット率は減少する [7]．しかし，近年の改良された LRU やその他の手法は $O(1)$ で走査が行えるため [8]，本手法によるオーバヘッド増加は問題とならないと考える．

3. 複数の技法の切り替え制御

続いて本節では，自律ディスクにおいて，複製データへのアクセス回送とマイグレーションを併用し負荷の変動へ対処するアプローチの下，これらの協調制御手法として RM-Combination を提案する．RM-Combination は 2 種の手法の切り替え制御と，併用制御の 2 機能からなる．

切り替え制御 現在の格納データに対する負荷偏りを複製のみで解決不可能であると判断した時点でマイグレーションを発動する．

併用制御 マイグレーションによるデータ移動経路の選択と，マイグレーションによる各ノードの負荷上昇を考慮したアクセス回送率戦略立案を行う．

以下では，まず前提とする環境を述べ，マイグレーションに起因する負荷見積もり方を紹介する．そして RM-Combination について述べる．

3.1 前提とする環境

ここでは，次の手順を想定する．

全てのリクエストはシステムを構成する各ストレージノードに均等に到着する．各ノードは，分散ディレクトリを走査し，リクエストが該当するデータ（のプライマリデータ）を保持しているノードにリクエストを回送する．プライマリ保持ノードは指定された分配率を保持しており，2.1 で述べたような判定手法を用いて，アクセスの回送の是非を決定し，必要があればリクエストを複製保持ノードへ転送する．

回送率は全てのノードへのリクエストが均衡化するよう，以下のように決定されることを前提とする．全てのノードは，現在の負荷値を記録しており，定められたインターバル時間ごとに次の作業を行う．

1. 全ノードは，一つのノードに負荷情報を送信する．
2. 負荷情報を受け取ったノードは，全てのノードで負荷が平坦化するような回送率を計算する．
3. 計算し終わった回送率を各ノードに送信する．
4. 各ノードは与えられた回送率をセットする．

データ移動による負荷の見積もり

データ移動に伴う各ノードの負荷上昇量については，各ノードのスループット対レスポンスタイム性能曲線を用いて推定する方法を提案している [5]. ここではその概略を述べる．

移動するデータ量を Q_{mig}[Byte] とする．移動による負荷の平均を L_{mig} [Byte/s], 移動にかかる時間 t_{mig}，及び Q_{mig} の関係は $L_{mig} = Q_{mig}/2t_{mig} = Q_{avg}/2(t_1 \times s)$ となる．ただし，データ 1 つ辺りの平均転送時間を t_1 [s] とデータ一つあたりの平均サイズ Q_{avg} [Byte] とした．

分母の 2 は読み出しと書き込みが逐次的に行われる場合のためである．読出しと書き込みが同時に行われる場合必要ない．

今，システムに対する外部からのアクセスリクエストに対する性能要件として定義されたレスポンスタイムを R_{req} [s] とおく．データマイグレーションによるアクセスは，各ノードにおいては外部からのアクセスリクエストと等価に扱われるため，データ一つあたりの転送時間 $t_1 < R_{req}$ となる．よって，マイグレーションによる負荷 L_{mig} は, $Q_{avg}/2R_{req}$ により求められる．

実際のマイグレーション負荷が過小評価であり L_{mig} よりも大きい場合，誤った見積もりによる負荷均衡化戦略により当該ノードの負荷は計画よりも上昇する．その結果，ノード性能が落ち, マイグレーション速度は減少する．以上を繰り返すことにより，マイグレーション負荷は L_{mig} に収束すると考えられる．

3.2 切り替え制御

複製のみを利用した負荷均衡化とマイグレーションの切り替え制御を行う．現在のデータ配置に対する負荷偏りを複製のみで解決不可能であると判断した時点でマイグレーションを発動し，マイグレーションによる負荷均衡化の一纏まりの処理が終了するまで両手法を併用する．本節では，各ノードが許容最大性能を超えないための 3 つの閾値の併用について述べる．

Chained Declutering + one-copy 複製配置では，各データに対し保持ノードが 1 つあるため，あるデータに対する負荷は各ノー

[1] http://www.de.cs.titech.ac.jp/

ド許容性能の最大 2 倍まで許容される．そこで，この全ノード負荷平均の 2 倍を閾値 1(相対閾値) として用いる．ノード i の複製はノード $i+1$ へ配置されており，またノード i はノード $i-1$ の複製を保持しているとする．$LP_i(t)$ を時刻 t におけるノード i に格納されたプライマリデータへの負荷，$LB_i(t)$ を複製データへの負荷とする．ノード $i+1$ へ回送を行わなかった場合のノード i の負荷は $LP_i(t)+LB_{i+1}(t)+LB_i(t)$ である．今，全 n ノードにかかる負荷の平均 $L_{avg}(t) \triangleq \frac{1}{n}(\sum_i LP_i(t)+LB_i(t))$ に対し，その 2 倍を超える負荷のノードが存在するとき，すなわち $LP_i(t)+LB_{i+1}(t)+LB_i(t) > 2\times L_{avg}(t)$ の時，時刻 t+1 において不均衡が発生すると考えられる．よってデータマイグレーションを行い，大域的な負荷を均衡化する．

負荷の偏りが 2 倍以内であったとしても，平均負荷がノード性能と比較して十分高い場合，破綻する可能性がある．そこで閾値 2(絶対閾値) を導入する．ノード i の許容最大負荷量を Lm_i とする．絶対閾値は，そのノードへの負荷が許容性能の 2 倍から，急激な変化に対応するためのマージンを除いた値とする．即ち，$max_i(LP_i(t)+LB_{i+1}(t)+LB_i(t)) > (Lm_i - L_{mig})\times 2$ となる t において，マイグレーションを実行する．ここで L_{mig}(単方向マイグレーションのための負荷値) をマージンとして用いた．

また，負荷の偏りが大きいが，全体負荷量が低い場合，データを移動する必要はない．ここでは閾値 3(下限閾値) として，$max_i(LP_i(t)+LB_{i+1}(t)+LB_i(t)) < (Lm_i - L_{mig})$ を置いた．

以上のように，$max_i(LP_i(t)+LB_{i+1}(t)+LB_i(t)) > (Lm_i - L_{mig})\times 2 \bigvee LP_i(t)+LB_{i+1}(t)+LB_i(t) > 2\times L_{avg}(t)$ である閾値によりマイグレーションを発動する．

3.3 併用制御

切り替え制御によりマイグレーションを行うことが決定したら，改めて全てのノードから負荷情報を取得し [9] データ移動戦略を決定する．ここで，我々が提案するマイグレーション時に複製へのアクセス回送を併用する手法 [5,6] を用いる．

3.3.1 データ移動経路決定, 回送率算出

自律ディスクはデータ配置戦略として値域分割を，複製配置戦略として Chained Declustering を用いている．これは，格納されるデータを一意の順序付けで並べ，その部分範囲を各ノードに割り当て，さらにデータ配置戦略によって関連ができた”隣の”ノードに対して自身の持つデータの複製を配置している．

よって，右方向と左方向マイグレーションの 2 種類がある．各方向のデータ移動には，2 通りのデータ経路が選択できる（図 2）．ここでは，移動元ノードに負荷がかかる経路を S(Self)-pattern，移動元ノードの両サイドいずれかのノードに負荷がかかる経路を N(Neighbor)-pattern と呼ぶ．図 2 では左図のように pri2 中のデータを pri3 に，耐故障性を考慮して bak2 中の複製を bak3 へ移動する．この場合，disk3 中では bak2 → pri3 を，ポインタ張替えにより行うことができる．よって，pri2 → bak3(負荷は disk2 と disk4）もしくは bak2 → bak3(負荷は disk3 と disk4) の何れかにより，目的のデータ移動を行うことができ，マイグレーションに起因する負荷を 4 操作から 2 操作へ減らすことができる．左方向も同様に 2 つの経路を考えることができる．

3.3.2 データ移動経路決定アルゴリズム

S,N の pattern を各ノードの負荷を考慮して決定することで，各マイグレーションによる負荷はより余裕のあるノードへとかかるようにデータ経路が設定される．データ移動経路決定アルゴリズムは以下に記す．

1. 経路パターンに拠らない負荷を，各ノードの負荷リストに加える．ノード i が右方向マイグレーションを行うならノード $i+2$ へ，左方向マイグレーションを行うなら，ノード $i-1$ へ L_{mig} を加える．
2. ノード 1 から順にマイグレーションを行うかを見る．方向に拠らず，ノード i とノード $i+1$ の負荷値を比べ少ない方へ L_{mig} を加える．ノード i に加えたなら経路パターンを S-pattern に設定する．ノード $i+1$ に加えたなら経路パターンを N-pattern に設定する．
3. 以上を全てのマイグレーション元ノードにおいて行う．

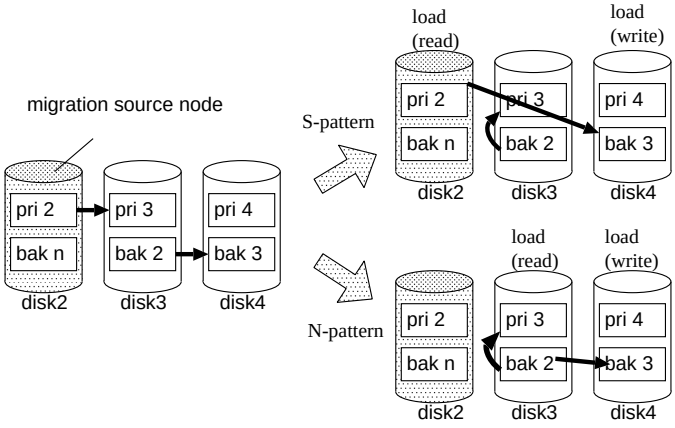

図 2: 右方向マイグレーション実行時の 2 種類のデータ経路. $disk_i$ に格納されたプライマリデータ領域を pri_i, その複製データ領域を bak_i と表記した.

Fig.2 Two data paths when migrating right. Pri_i indicates $disk_i$'s primary data, and bak_i indicates the replica of pri_i.

この後，現在の負荷情報と，マイグレーションにより増加する見積もり負荷情報を加えたリストに対し，3.1 で述べた方法と同様にして，全ての負荷が均衡化するように回送率を決定する．以上の結果を全てのノードに対し配信し，終了を待つ．

4. 実験

提案する RM-Combination の性能保障面での有効性を示すために，自律ディスク模擬実装上に提案手法を実装し実験を行う．自律ディスクに提案手法であるキャッシュ h/m 法および RM-combination を実装し，クライアントからのアクセス負荷偏り・高負荷環境下における各ノードの挙動を観察した．

4.1 実験環境

実験には，自律ディスクを実装した先進ストレージシミュレータを用いた．用いた実装の詳細は [6] を参照されたい．自律ディスク 4 台構成システムに対し，1MB のデータを合計 3000 個挿入した，それらに対して Pentium4 2.8cGHz のクライアントから 6 個のスレッドを用いて，値域のあるデータ付近に正規分布上の偏りがおこるようアクセスを発生させた．また，正規分布の山の部分が 1 時間に 1 回同じデータにあたる速度で変化させた．

4.2 実験結果と考察

負荷をかけはじめてからおよそ 40 分間，各ノードのワークロードと回送率の変化を観測した．

図 3(a) にシステム中ノード 0 に格納されているプライマリデータと，ノード 1 に格納されているその複製データへの合計スループットを示す．以下のグラフはすべて，横軸は開始してからの相対時間であり，縦軸は 2 秒毎に集計したスループットである．各データの利用傾向は時間と共に大きく変化している状況が確認できる．また，アクセス傾向が大きく変化している部分はノード 0 中の，アクセス集中している一部のデータが他のノードへ移動したことを表す．

図 3(b) にノード 0 が実際に供給しているデータ（ノード 0 のプライマリデータとノード 3 の複製データ) の合計スループットを示す．曲線は，17MB/s 程度でほぼ一定であり，複製へのアクセス回送によってノード負荷が一定に抑えられているのがわかる．図

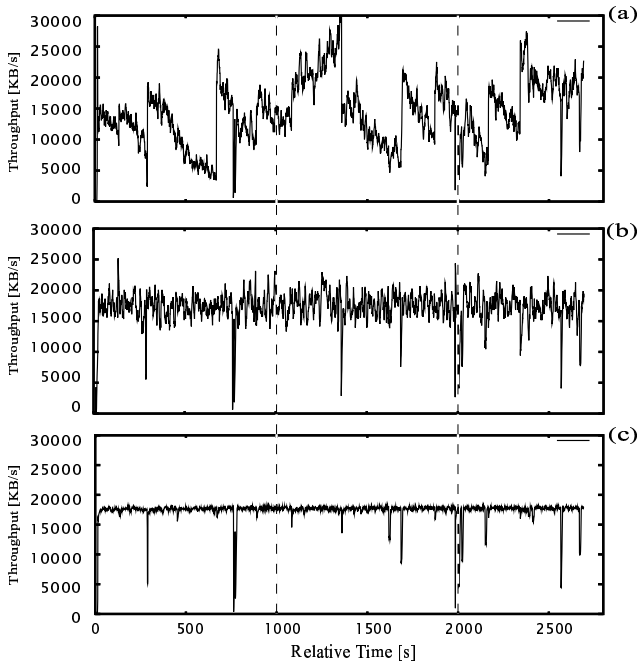

図 3: 実験中における各スループット．(a) データ 0 の負荷, (b) ノード 0 の負荷, (c) システム平均

Fig.3 Experimental Result: (a)load of data0, (b) load of data in node0, (c) system average

3(c) はシステム全体の合計スループットから算出した 1 台あたりの平均するープットであり，(b) と比べさらに分散が小さい．このことから，(b) の大きな振幅はアクセス回送率戦略の誤差によるものであると考えられる．

これらの図から，マイグレーションによる負荷が複製へのアクセス回送により分散されていることがわかる．(a) の図において大きく曲線が変化している時点において，(b),(c) で大きくスループットが低下している傾向が見られないためである．これにより，我々の提案する RM-Combination がマイグレーションによる負荷を適切に吸収し分散できていることと考える．

図において急激にスループットが落ちている点が散見される．これらは，全てのノードのスループットが同期的に急激に落ちていることから，クライアント側において Java VM 上のガーベジコレクタ等で一時的に処理が停止したためであると考える．

5. まとめと今後の課題

本稿ではストレージノード間の負荷均衡化と要求性能保障の両立を目的とし，我々の提案する自律ディスクに対し複製へのアクセス回送による負荷分散機能を導入した．

アクセス回送による負荷均衡化をより効率よく実現するため，ストレージノード上のキャッシュヒット率を考慮したアクセス分配判定手法を提案し，シミュレーションによりその有効性を示した．

また，自律ディスクが従来から備えるデータ移動による負荷均衡化手法と，複製データへのアクセス回送による手法との切り替え・併用制御手法として RM-combination を提案した．提案手法では，データマイグレーションの発動回数を抑えると同時に，データマイグレーション実行時の負荷をアクセス回送率戦略立案に含めることによる要求性能保証を行う．これらについて，実装・実験により大小のアクセス負荷偏りを解消できることを確認した．

今後，複製のみの場合，マイグレーションのみの場合といった環境との詳細評価や，今回設定した各パラメータの妥当性評価等が必要であると考えている．また発展として，負荷が急に変動したときの制御方法，アクセス履歴を用いた予測的閾値変更，ヘテロ環境への適応といったことが課題として挙げられる．

[謝辞]

本研究の一部は，科学技術振興機構戦略的創造研究推進事業 CREST，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 (16016232) 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」の助成により行なわれた．

[文献]

[1] Gerhard Weikum, Axel Mönkeberg, Christof Hasse, and Peter Zabback. Self-tuning database technology and information services: from wishful thinking to viable engineering. In *VLDB*, pp. 20–31, 2002.

[2] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The Google file system. In *SOSP*, pp. 29–43, Boton Landing, NY, Oct 2003.

[3] Joel L. Wolf, Philip S. Yu, and Hadas Shachnai. Disk load balancing for video-on-demand systems. *Multimedia Systems*, Vol. 5, No. 6, pp. 358–370, 1997.

[4] Haruo Yokota. Autonomous Disks for Advanced Database Applications. In *Proc. of DANTE'99*, pp. 441–448, Nov. 1999.

[5] 小林大, 渡邊明嗣, 田口亮, 上原年博, 横田治夫. 負荷分散のためのデータ移動による性能低下を抑制するアクセス回送制御. 信学技報，電子情報通信学会，DE2004-112, pp. 35–40, October 2004.

[6] 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. *DBSJ Letters*, Vol. 3, No. 2, pp. 65–68, Sep. 2004.

[7] Ramakrishna Karedla, J. Spencer Love, and Bradley G. Wherry. Caching strategies to improve disk system performance. *IEEE Computer*, Vol. 27, No. 3, pp. 38–46, 1994.

[8] Nimrod Megiddo and Dharmendra S. Modha. ARC: A self-tuning, low overhead replacement cache. In *USENIX FAST'03*, March 2003.

[9] 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 D1, Vol. 85-D1, No. 9, pp. 877–886, September 2002.

小林 大　Dai KOBAYASHI

平 17 東工大大学院・情報理工・計算工・修士課程了．同大大学院・情報理工・計算工・博士後期課程在学中．日本データベース学会学生会員．

渡邊 明嗣　Akitsugu WATANABE

平 14 東工大大学院・情報理工・計算工・博士前期課程了．同大大学院・情報理工・計算工・博士後期課程在学中．日本データベース学会学生会員．

田口 亮　Ryo TAGUCHI

平 6 慶應義塾大大学院・理工・計測工・修士課程了．同年より NHK 放送技術研究所．映像情報メディア学会会員．

上原 年博　Toshihiro UEHARA

昭 56 慶應義塾大・工・電気工・修士課程了. 昭 59 より NHK 放送技術研究所. 電子情報通信学会，映像情報メディア学会各会員.

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)．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．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．日本データベース学会, 電子情報通信学会，情報処理学会，人工知能学会，IEEE，ACM 各会員．