社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2004－DBS－134 (II) (71)
2004／7／14

分散ストレージ上の複製へのアクセス要求配分を取り入れた負荷均衡化手法

小林 大* 渡邊 明嗣* 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
† 東京工業大学 学術国際情報センター
〒 152-8552 東京都目黒区大岡山 2-12-1
{daik,aki,muu}@de.cs.titech.ac.jp, yokota@cs.titech.ac.jp
‡NHK 放送技術研究所
〒 157-8510 東京都世田谷区砧 1-10-11
taguchi.r-cs@nhk.or.jp, hayashi.n-gm@nhk.or.jp, uehara.t-jy@nhk.or.jp

概要

データマイグレーションは，並列データベースやストレージシステムにおいてアクセス偏り除去のための有効な手段である．しかし同時に，ディスクアクセスやネットワーク消費を伴い，システム性能を低下させ得る．本稿では，複製データへアクセス要求を分配し複数のストレージ装置に分散させることで，データマイグレーション時の性能低下を抑える手法を提案する．これにより負荷集中時の円滑なデータ移動が可能となる．また，我々の提案する自律ディスク上で提案手法の実験を行い，有効性を検証する．

キーワード： 分散ストレージ, 自律管理, 負荷均衡化, マイグレーション, 複製, 自律ディスク

Replica-assisted Migration: a method to handle workload skew on distributed storage systems

Dai KOBYASHI* Akitsugu WATANABE* Munenori YAMAGUCHI*
Ryo TAGUCHI‡ Naoto HAYASHI‡ Toshihiro UEHARA‡ Haruo YOKOTA†,*

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

Abstract

Data migration is an efficient method to handle skew of access request distribution in the field of distributed storage systems. However, it also can cause a performance degradation temporally because it uses system resources such as local I/O or networks. In this paper, we propose a method to decrease performance degradation of data migration, distributing access requests to replica-data. To use the method, we can use data migration even if access requests concentrate on a node. We also show the efficiency of our method by using the result of experiments.

KEYWORD: storage system, autonomic management, load balancing, migration, replication, autonomous disks

1 はじめに

近年，扱うデータの大規模化やそれに伴う管理コスト肥大化により，ストレージシステムの自律管理機能が注目されている．その中でもシステムの持つ資源を最大限に利用するための設定，特にパフォーマンスチューニングはその煩雑さから管理コスト増大の大きな要因であり，自律管理対象として研究がされている．

そのような自律的パフォーマンスチューニングのひとつにアクセス負荷偏り除去のためのデータマイグレーションがある．分散配置されたストレージノードに対して，各データのアクセス負荷を考慮してデータ配置を決定することによりストレージ資源の性能を大きく低下させるアクセス集中を回避することができる．また，システム運用中に動的にデータ再配置を行うデータマイグレーションが並列データベースやストレージシステムにおける自律的な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 [1].

しかし，データマイグレーションによる負荷分散は，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．アクセスパターンの変化が激しい場合や，負荷偏りの評価精度が低い場合，マイグレーションにより負荷を追い出そうとするディスクが既に飽和しており，マイグレーション処理によりサービス性を低下させてしまう恐れがある．これにより，当該ディスク上のデータに対するアクセス要求のレスポンス性が悪化し，システムに要請されるアベイラビリティが満たされなくなる．

一方で，大規模なストレージシステムの多くは障害対策のために複製を保持しており，この複製を負荷分散に用いることが可能である．しかし，複製のみによる負荷分散は，負荷偏りが大きくなるにつれ複製数が増加し容量対性能比が鈍くなる．

よってマイグレーション処理と複製利用による負荷分散を効率よく併用することが肝要である．

本稿では，複製データにアクセスの一部を振り分けることで処理飽和ストレージ装置からデータマイグレーションのためのリソースを確保する手法 ***Replica-assisted Migration***（以下レプリカアシスト）を提案する．レプリカアシストでは，データマイグレーションによる性能低下分のアクセス要求をあらかじめレプリカ間のアクセス分配により他のノードに振り分けることにより，処理飽和による性能低下を一時的に押さえる．

また，障害対策用の複製データを持ちデータマイグレーションによる負荷均衡化を行う分散ストレージ技術である自律ディスクへの適用例を示し，その後自律ディスクの模擬実装上での実験結果を用いて提案手法の有効性を示す．

2 負荷均衡化処理

本章では，アクセス負荷の偏りによる弊害について述べ，その後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の特徴とその問題点，そして複製を用いた負荷分散について述べる．

2.1 アクセス負荷の偏りと弊害

大規模なストレージシステムは，多数のストレージ装置（ノード）をネットワーク結合し構成される．そして必要とするデータ群を固定長分割や意味的分割し格納する．データ断片に対するアクセス分布が偏ると，一部のノードにリクエストが集中し，一般にレスポンスタイムが幾何級数的に遅くなる [3]．また，ディスク装置やテープ装置は単体では並列読み書き不可であるため，レスポンスタイムの低下はスループット低下に繋がる．あるいは過度の即応性低下はシステム障害へと至る．よってアクセス負荷偏りは均衡化する必要がある．

2.2 データレプリケーション

大規模なストレージシステムの多くは障害対策のために複製（レプリカ）を保持しており，ノード障害からのゼロタイム復旧を可能にしている．データ書き込みアクセスへの即応性を考慮すると複製間一貫性は非同期で管理されるべきであるが，近年のス

トレージ利用は読み出しアクセスと追記アクセスに大きく偏っており [2], 複製間同期書き込みを考慮できる．よって，常に一貫性を保った複製が利用可能となり，これをアクセス要求処理に転用することができる．

しかし，複製のみによる負荷分散は，負荷偏りが大きくなるにつれ複製数が増加し容量対性能比が悪くなる．あるいは，複製数が一定である場合，一定量以上の負荷偏りが発生した場合には適用できなくなる．

2.3 データマイグレーションによる負荷均衡化

各ノードの性能を考慮し，格納データに対するアクセス負荷が均等になる様再配置することにより，ノードごとのアクセス負荷偏りを除去可能である．さらに，分散ディレクトリと組み合わせ，索引構造の ACID 性を満たしながらデータを移動することで，システム運用中の負荷均衡化をユーザ透過に行うことが可能となる．

2.4 データマイグレーションによる資源消費

データを移動するためには，ノード上の格納データをメモリ上に蓄え，ネットワークを通じて相手に転送しなくてはいけない．しかし，ディスクアクセスやネットワーク転送は共にストレージ装置において枯渇しやすい（負荷耐性の低い）資源である．よって，負荷が集中してからデータマイグレーションを行うことは難しい．

負荷評価精度の問題や利用者傾向の変化，あるいはノード故障などのシステム構成の変化により，あるノードの負荷が急激に高まることは往々にしてあり得る．一度負荷が偏ると，単純なマイグレーションだとレスポンスタイム低下によりシステム障害に発展し得る．一方，負荷集中前に，つまり負荷の偏りが少ないうちにマイグレーションする場合，マイグレーション頻度が増えることにより資源利用量が増加し，やはりシステム性能を圧迫する．

よって，マイグレーションの頻度を増加させることなく性能低下を抑える手法が必要となる．

3 レプリカアシスト

我々は，負荷が集中してしまった後でも，システム性能低下を抑え，データマイグレーションを行う手法として，***Replica-assisted Migration***（レプリカアシスト）を提案する．本節では，まず提案するレプリカアシストについてその特徴と手順を述べる．さらに，自律ディスクに対しレプリカアシストを適用する例を述べる．

3.1 複製へのアクセス分配による均衡化処理の補助

複製へのサービス振り分けを用いることで，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が行うことを可能とする，レプリカアシストを提案する．

3.1.1 概要

レプリカアシストの概念を図 1 に示す．

マイグレーションにより負荷が上昇するのは移動元及び移動先のノードである．そして，マイグレーション発生時には移動元ノードに負荷が集中している．そこで，障害復旧用の複製を用いて，移動元のディスクの負荷の一部を予めレプリカ側に分配し，移動元のディスク内にマイグレーション用の余力を確保した後マイグレーションをすることで，性能低下を抑える．

3.1.2 手順

提案手法は，マイグレーション発生時に以下に述べる処理を加える．

1. 負荷評価により算出された負荷量から，マイグレーションを行うデータ量を決定する．

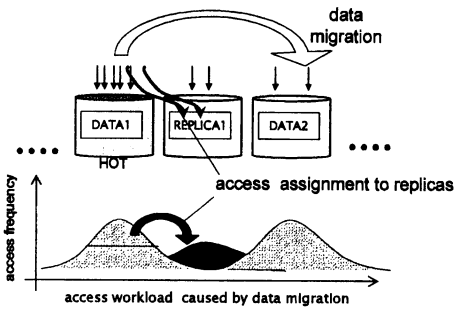

図 1: レプリカアシストの概念：負荷集中ディスクからマイグレーションリソースを捻出

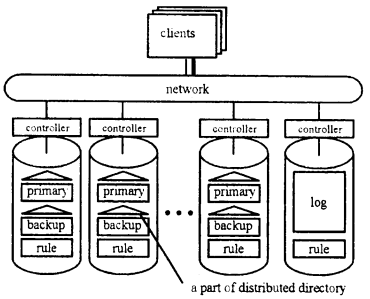

図 2: 自律ディスク

2. 複製との一貫性保持が非同期であれば同期に切り替え，複製に対しまだ適用されていない更新を適用する．

3. 移動元ディスクに対するアクセス要求を以下の式で算出される割合に従い，複製データの存在するディスクへと転送ようにセットする．

$$min\left(1.0, \frac{\text{マイグレーションする負荷量}}{\text{ディスク内プライマリデータの負荷量}}\right)$$

4. データを移動する．その間のアクセス要求は転送率に基き複製へ転送される．

5. データ移動完了後，複製へのアクセス要求転送率を 0 にする．

3.1.3 特徴

特徴として，移動元ノードの負荷量が最大許容量に近い場合もデータマイグレーションに依る負荷均衡化を行うことが可能であることが上げられる．

またアクセス要求をノード単位で管理することで，アクセス要求転送時に無駄にメタデータを読み出す必要がない点が挙げられる．

単純な割合転送により，複製保持ノードの負荷が許容量を超えてしまう可能性があるが，これは今後の課題とする．

3.2 適用例

レプリカアクセスを，実際に複製配置とデータマイグレーションを利用するシステムに適用する例として，我々が提案する分散ストレージ技術である自律ディスクへ適用する場合を述べる．

3.2.1 自律ディスク

自律ディスク [4] は我々が提案している可用性やスケーラビリティに優れたネットワークストレージ技術である．自律ディスクではシステムは図 2 のように，ネットワークに接続されたディスクノードのクラスタにより構成される．システムを運用する上で特別な集中管理サーバは必要としない．

動的データ配置管理などのデータ管理をクライアントから透過的に行うために，値域分割に依るデータ配置分散ディレクトリ構造として分散 Btree 構造を用いることを想定している．また，障害復旧のための複製配置戦略として Chained Declustering[5] を採用している．これはストレージ装置列に対しリング状に複製を配置する，信頼性が高い複製配置戦略である．また値域分割は，データ断片に対する識別子の一意な順序付けに従って並べ，その連続部分範囲を各ストレージに格納する，並列データベースにおいて用いられる手法である．

その他に ECA ルールやトランザクション処理等の機能を持ち，これらを組み合わせることによりデータ分散配置，偏り制御，耐故障性および障害からの

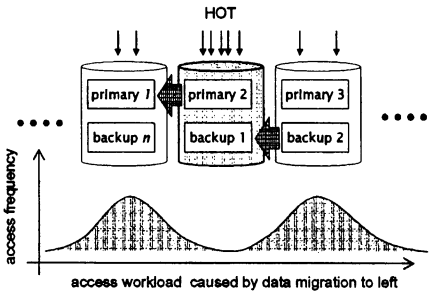

図 3: 左方向マイグレーションと，マイグレーション自体による負荷の上昇

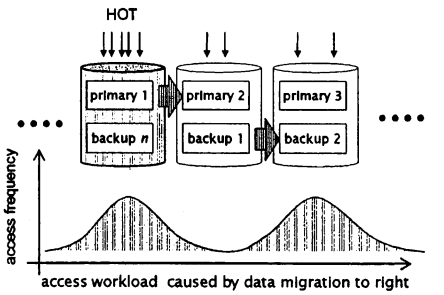

図 4: 右方向マイグレーションと，マイグレーション自体による負荷の上昇

回復といった高度なシステム管理を自律的に行うことができる．

3.2.2 自律ディスクとマイグレーション

自律ディスクではデータマイグレーションにより負荷均衡化を行っている．自律ディスクのデータ配置は値域分割を利用しているため，論理的に隣接するノードのみが移動先対象となる．また現在はリング化していない，両端の存在する Chained Declustering を複製配置戦略に用いているため，データマイグレーションには右方向（データ Key 値正方向）と左方向（データ Key 値負方向）が考えられる．

左方向マイグレーションの場合に負荷の増加するノードを図 3 に示す．複製配置の利用により，プライマリデータ移動元ノードである図中中央のノード内ではプライマリデータをバックアップ領域へと移動するだけであるため，負荷が掛からないようになっており，レプリカアシストを利用する必要はない．

一方，右方向マイグレーションの場合に負荷の増加するノードを図 4 に示す．この場合プライマリデータ移動下ノードである図の左側のノード上でデータ読み出しによる負荷が発生してしまう．一方，その複製データが配置された図の中央のノード内ではバックアップデータをプライマリ領域へ移動するだけであるため，負荷は掛からない．よって，提案するレプリカアシストを用いて移動元ノード上データへのアクセスの一部をバックアップ側へ転送することで移動元ノードの負荷量を一時的に減少させマイグレーションのための資源を捻出することが有効に作用すると考えられる．

4 実験

提案するレプリカアシストの性能面での有効性を示すために，実装し実験を行う．

4.1 実験環境

実験は，我々の提案する分散ストレージ技術である自律ディスクの模擬実装上で行う．これは Linux クラスタ上に Java を用いて模擬実装されている．今回の実験では表 1 に示す構成の PC と十分なバックボーン性能を持つスイッチを用いて，実験環境を構成した．また，クライアントと自律ディスクのインターフェースに HTTP を用いる実装 [7] を利用した．

ユーザに対して透過的なデータ配置を実現する分散ディレクトリには，aB$^+$-Tree[8] を利用した．よって，マイグレーションによるデータ配置の変更毎に全ての自律ディスクノード間でデータ配置情報更新のための通信が発生する．

今回用いた PC は，扱うデータサイズを 1MB 固定長とした場合，1 台辺り 25 個/s のリクエストを処理できる性能を示した．

表 1: ストレージノード・クライアントノード 性能緒元

#nodes	6台 (Storage) + 8台 (Clients)
CPU	AMD Athlon XP-M 1800+ (1.53GHz)
MEM	PC2100 DDR SDRAM 1GB
Network	1000BASE-T
HDD	TOSHIBA MK3019GAX (30GB, 5400rpm, 2.5inch)
OS	Linux 2.4.20
Local File System	ext3 FS
Java VM	Sun J2SE SDK 1.4.2_04 Server VM

また，負荷値としては，最近20秒間のアクセス履歴を基にした単位時間当たりのアクセス数とした，負荷評価値は500msごとに隣接ディスクへと送られるトークンパッシング [6] により収集し，トークンを保持しているディスクがマイグレーションを行う方式が実装されている．

4.2 実験方法

6台構成の自律ディスククラスタに対して，1MBのデータを合計1000個挿入した．

そして，8台のクライアントPCから一定の読み出し負荷をかけた．今回のシステムは最大150(= 25×6)個/sのリクエストを処理可能であるため，8台のクライアントから各最大18個/sのリクエストを生成し送信した．アクセス分布については図5に示すような3通りのパターンを用意した．(a)は偏りのない場合，(b)は偏りが大きく変化の緩やかな場合，(c)は偏りが小さく変化の速い場合を想定している．(b),(c)については図の通り負荷ピークが時間と共に変更するモデルとした．ピーク地の移動速度は30分で1周するスピードとした．

計測時はまず，無負荷時から該当するパターンの負荷を発生させた．その1分後から5分間の総処理リクエスト数を観測することで，急なアクセスパターン変化に伴うマイグレーションコストを再現した．

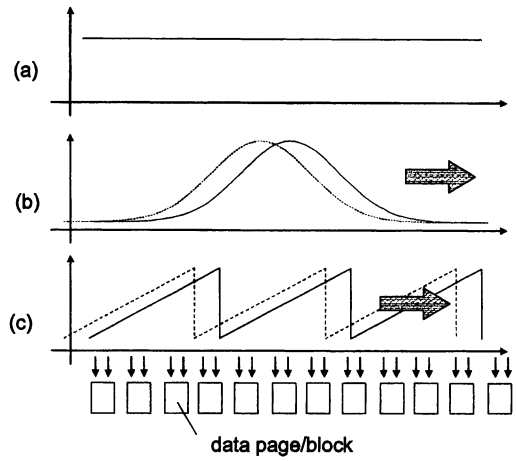

図 5: 実験に用いたアクセスパターン: (a) 一様分布 (b) 正規分布 (c) 櫛型分布

4.3 結果と考察

実験を行った結果を図6に示す．これは最大リクエスト数の固定されたクライアントからのリクエストに対する5分間の平均スループットを示す．つまり，1回1MB要求のリクエストの秒間処理数と同義である．

4.3.1 一定分布

まず偏りのない一定アクセス分布へのスループットについて考察する．グラフよりマイグレーション機能をオフにした場合が一番性能が高く，マイグレーションをオンにした場合，レプリカアシストを実装した場合の性能が順に低くなっている．これは負荷評価の誤差により，本来必要のないデータマイグレーションが発生したためであると考えられる．また，各ノード間の負荷に差がない状態でレプリカアシストを使うと，隣接ノードの負荷値が偏りがないにもかかわらず上がってしまい，より性能が低下したと考えられる．

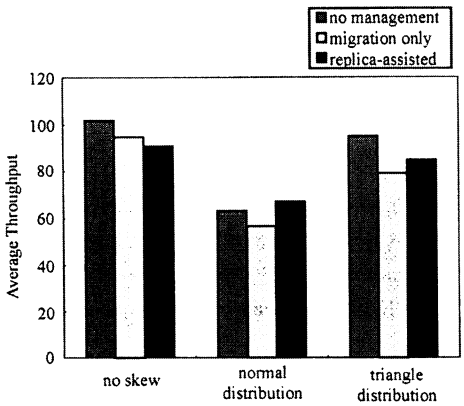

図 6: 各手法実行時の平均スループット

4.3.2 正規分布

正規分布状のアクセス分布は大きな偏りを表している．これにより，マイグレーション機能をオフにした場合は偏りのない場合に比べ大きく性能が低下していることがわかる．これは，ピーク負荷のデータを格納している二つのノードに処理が集中してしまいそのサービス性能 (25 リクエスト/$s \times 2$) と，その他のノードへの僅かなリクエストの合計値が数値として出ている．

また，マイグレーション機能のみを用いている場合は，ピーク負荷格納ノードは転送に必要な能力のうちの幾分かをデータマイグレーションによる負荷偏り除去に利用してしまうため，マイグレーションを行わない場合よりさらに性能が低下してしまっている．

一方，提案手法であるレプリカアシストを用いた場合は，このマイグレーションによる負荷が他の負荷の少ないノードに分散され，性能低下が抑えられていることがわかる．さらに，負荷集中ノードの余力が増えマイグレーションの速度が向上したため，負荷偏り制御を行わない場合よりもよい結果が得られている．

これより，負荷偏りが非常に大きいものであった場合レプリカアシストが有効に作用することがわかる．

4.3.3 櫛状分布

ノコギリの歯のような形状（グラフ中 triangle distribution) のアクセス分布を掛けた場合は，緩やかな負荷偏りのためマイグレーションを行わない場合の性能低下はあまり大きくない．一方，マイグレーションを行ってしまうと，ピーク負荷（三角の山の頂上部分）のデータを格納しているノードでマイグレーションがおこり，その資源をマイグレーションに利用してしまうため，システム性能が低下してしまっている．さらに，アクセス分布は常に変化しているためマイグレーションが停止することはなく，データマイグレーション機能が常にシステム性能を圧迫してしまう．

一方，レプリカアシストを使った場合，この性能低下を幾分か抑えられていることがわかる．これにより，負荷偏りが細かく緩やかな場合でも，マイグレーションによる性能低下量の軽減としてのレプリカアシストは有効であることが確認できる．

しかし，それよりも偏り制御を行わない場合の方がよい性能を示している．これはレプリカアシストによるアクセス分配が複製保持ノードの負荷を平均以上に上昇させてしまっているためであると考えられる．アクセス分配の割合のより詳細な考察については今後の課題とする．

いずれの場合においても，マイグレーションによる性能低下をレプリカアシストにより緩和できていることが確認でき，レプリカアシストの目的とするマイグレーションによる性能低下の緩和が実現できていることが確認された．

5 まとめと今後の課題

本稿では，分散ストレージシステムにおける，性能低下の少ないマイグレーション処理 *Replica-assisted Migration* を提案した．提案手法では，負荷が集中したストレージ装置からデータを移動するために，当該ストレージ装置へのアクセスを複製データへ分配することでデータマイグレーションのための資源を確保する．本提案手法を我々の提案する分散ストレージ技術である自律ディスク上で実現する例を述べた．

さらに，自律ディスクの模擬実装に提案手法を実装し，マイグレーション時の性能低下を抑えることに成功した．

今後の課題として次のことが挙げられる．まず，データマイグレーション自身の実装および負荷評価についてより性能の高いものを用いて提案手法のさらなる評価を行うことが挙げられる．また，本稿では複製へ振り分ける割合をマイグレーションにより移動する負荷割合と同等としたが，この点についてもより詳細に考察すべきである．

さらに，今回は複製間一貫性は同期書き込みにより保持する前提としたが，書き込みアクセスの割合と同期/非同期データの割合に対する考察も必要である．そのほかに，ストレージ装置台数が多い場合に関する実証，複製間アクセス振り分けのみを利用した負荷分散機構との協調動作等が課題として挙げられる．

謝辞

本研究の一部は，科学技術振興事業団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．

参考文献

[1] G. Weikum, A. Moenkeberg, C. Hasse and P. Zabback "Self-tuning Database Technology and Information Services: from Wishful Thinking to Viable Engineering", Proc. of the 28th VLDB Conference, 2002.

[2] S. Ghemawat, H. Gobioff, and S.T. Leung, "The Google File System", 19th ACM Symposium on Operating Systems Principles, 2003. z

[3] H. Simitci, Storage Network Performance Analysis, Wiley Technology Publishing, 2003.

[4] H. Yokota, "Autonomous Disks for Advanced Database Applications", Proc. of International Symposium on Database Applications in Non-Traditional Environments(DANTE'99), pages 441-448, Nov, 1999.

[5] H.I. Hsiao and D.J. DeWitt, "Chained Declustering: A New Availability Strategy for Multiprocssor Database machines", Proceedings of the Sixth International Conference on Data Engineering, 1999.

[6] H.Yokota, Y.Kanemasa and J.Miyazaki, "Fat-Btree: An Update-Conscious Parallel Directory Structure", Proc. of the 15th International Conf. on Data Engineering, pp.448-457, 1999.

[7] 花井 知広, 横田 治夫, "自律ディスクを用いた Web サーバにおける負荷偏りの影響", 情報処理学会研究会報告, データベースシステム, DBS-128-29, 情報処理学会, 2002.

[8] Mong Li Lee, Masaru Kitsuregawa, Beng Chin Ooi, Kian-Lee Tan, and Anirban Modal, "Towards Self-Tuning Data Placement in Parallel Database Systems", Proc. of ACM SIGMOD, pages 225-236, 2000.