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

Consideration of Efficient Data Migration Assisted by a Data Replication

小林 大♢ 渡邊 明嗣♢ 山口 宗慶♢
田口 亮♠ 上原 年博♠ 横田 治夫♣

Dai KOBAYASHI Akitsugu WATANABE
Munenori YAMAGUCHI Ryo TAGUCHI
Toshihiro UEHARA Haruo YOKOTA

ストレージノード間のアクセス負荷偏りに対し，データマイグレーションは有効な手段として並列データベースやストレージシステムにおいて用いられる．しかし同時に，マイグレーションに伴うディスクアクセスやネットワーク転送により，システム性能を一時的に低下させ得る．本稿では，この性能低下を抑えることを目的とし，データマイグレーション時のみ複製データへアクセス要求を一部回送する手法を提案する．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．

Data migration is an efficient method to handle skew of access-request distribution in parallel databases and distributed storage systems. However, it also causes a performance degradation during the migration process temporally because it uses system resources such as local I/O or networks. In this paper, we propose a method of distributing access requests to replica-data to decrease performance degradation of data migration. The method enables data migration after access requests concentrate on a node. We show the efficiency of the proposed method by experiments using the Autonomous disks system we proposed.

1. はじめに

データマイグレーションは並列データベースや分散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．分散配置されたストレージ装置に対して，各データのアクセス負荷を考慮してデータを移動することでデータ配置を変化させ，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．このようなデータマイグレーション技術は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．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タイムが悪化し，システムに要請されるサービス品質が満たされなくなる．

♢ 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,muu}@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

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，静的な複製配置においては負荷偏りの変化に追従できず，動的な複製配置においても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ことや，あるいは複製配置変更時のアクセスによる性能低下といった問題がある．

そこで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を考える．

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，データマイグレーション実行中のみ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，自律ディスクのもつ複製配置とデータマイグレーションに関する制約に焦点を当て述べる．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. 研究背景

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．そしてデータ群に対して固定長分割（ページ，ブロック）や意味的分割（ファイル）を行い，分散格納する．各ノード単体は M/G/1 待ち行列と見なすことができ，レスポンスタイム R はディスクアクセス時間 S, S の分散 $(\sigma_s)^2$, およびリクエスト到着率 λ を用いて，以下の式で表される [3]:

$$R = S + (\text{待ち時間}) = S + \frac{(\sigma)_s^2 + S^2}{2} \cdot \frac{\lambda}{1 - \lambda S}$$

よって，アクセス集中により λ が増加すると，R は非常に大きくなる．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できないため，レスポンスタイムの低下はスループット低下に繋がる．過度の即応性低下によりシステムに求められるサービス品質を満たせなくなる可能性があるため，アクセス負荷偏りは均衡化する必要がある．

2.2 既存手法

アクセス負荷偏りを平坦化するために用いられる手法に，データレプリケーションとデータマイグレーションがある．

データレプリケーションでは，大規模なストレージシステムの多くが障害対策のためにもつ複製（レプリカ）データを，アクセス要求処理にも利用することでノード間偏りを平坦化している．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仮定できるためである．

しかし，静的な複製配置では負荷パターンの変動に対処できず，また動的な配置においても，負荷偏りが大きくなるにつれ，負荷均衡化のための複製数が増加し容量対性能比が悪くなる．さらに，動的な配置を行うレプリケーションは，その配置変更に伴うアクセスにより後述するマイグレーションと同等の問題を抱える．

一方データマイグレーションでは，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データを再配置する．これにより，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から透過に行うことが可能となる．

しかし，データを移動するためには，ノードがそのサービスに利用するディスクアクセスやネットワーク転送能力の一部を利用するため，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一時的にさらに性能を悪化させる．

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ではレスポンスタイム低下によりシステムに求められるサービス品質を満たせなくなる．一方，負荷集中前にマイグレーションを決定する場合，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，負荷集中後においても，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

マイグレーションにより負荷が上昇するのはデータ移動元及び移動先のノードである．そして，マイグレーション実行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データマイグレーション実行中のみレプリカ側に回送し，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，マイグレーション実行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．ここで，データ移動元ノード負荷量 W, 移動負荷量 d に対し，複製へのアクセス回送のための回送率 r を $r = min\,(1.0,\ d/W)$ と決定する．
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回送率 r の割合で，複製データの存在するディスクへと回送するようにセットする．
4. データを移動する．その間のアクセス要求は r に基き複製へ回送される．
5. データ移動完了後, 複製への要求回送率を 0 にする．

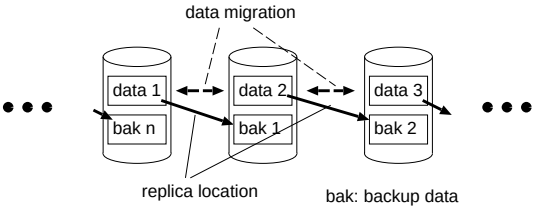

図 1: Chained Declustering と値域分割
Fig.1 Chained Declustering and range pertitioning

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，アクセス要求回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．

今回は単純な回送割合として，d/W を用いた．しかし実際の環境では，複製保持ノードへもアクセス負荷が発生しており，過度なアクセス回送のよって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．このような，他ノードも含めての回送割合の詳細な評価については今後の課題とする．

3.2 適用例

レプリカアシストを，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 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．

自律ディスクでは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，信頼性とアベイラビリティに優れた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．値域分割と Chained Declustering を組み合わせたデータ配置を図 1 に示す．

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，

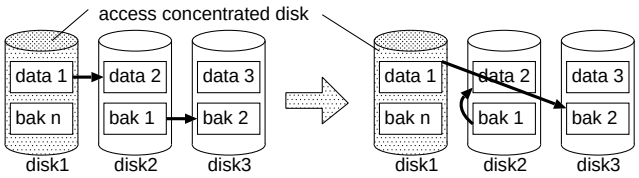

図 2: 右方向マイグレーション
Fig.2 A Migration to right

偏り制御，耐故障性および障害からの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．値域分割を前提とした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．

右方向マイグレーションの場合を図 2 左に示す．disk1 を負荷集中ノードとする．この場合, disk1,2,3 上でマイグレーションに伴うデータ読み出し・書き込み負荷が発生する．そこで，data1 → data2 の移動を disk2 内の bak1 → data2 で代用することで図 2 右のような移動経路に変更でき disk2 へのアクセスを削減できる．さらに，提案するレプリカアシストを用いて disk1 上の data1 へのアクセスの一部を disk2 上の bak1 へ回送することで disk1 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できる．

一方, 左方向マイグレーションの場合，右方向と同様に複製配置を用いたデータ移動経路の変更を行うと，負荷集中ノード内では主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，負荷が掛からず，レプリカアシストを利用する必要はなくなる．

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 実験環境及び実験方法

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ネットワークスイッチを用いて，実験環境を構成した．この PC は予備実験において 1 台辺り 25 個/s の 1MB 読み出しリクエストを処理できる性能を示した．

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[7] を利用したため，データ配置の変更毎に更新情報のブロードキャストが発生する．

また負荷値としては，最近 20 秒間のアクセス数を 20 で割った，1 秒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．

このような環境化の 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 個格納し，8 台のクライアント PC から各最大 18 個/s の読み出し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 3 に示すような 3 通りのパターンを用意した．(a) は偏りが大きく変化の緩やかな場合，(b) は偏りが小さく変化の速い場合を想定している．(a),(b) 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．ピーク時の移動速度は 30 分で 1 周するスピードとした．

表 1: ストレージノード・クライアントノード 性能緒元
Table 1 Spec of PCs used as storage and client nodes

#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

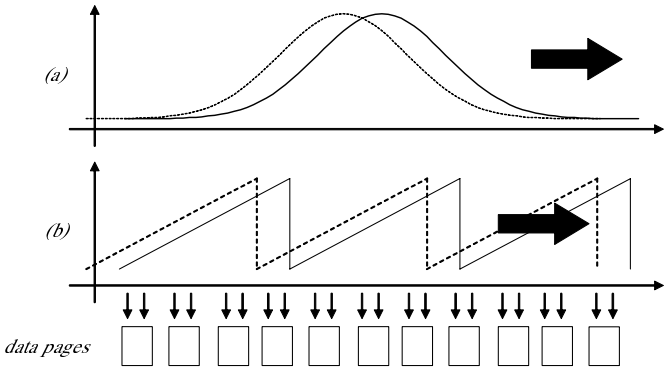

図 3: 実験に用いた負荷パターン: (a) ガウス分布 (b) 櫛型分布
Fig.3 Access patterns:(a)Gaussian Distribution pattern (b) Triangle pattern

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, つまりスループットを観測した．これは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想定している．

4.2 結果と考察

実験を行った結果を図 4 に示す．

ガウス分布状のアクセス分布の場合，マイグレーションのみではアクセス偏りのない場合の 100 リクエスト/s に比べ, 負荷に偏りがあると 62 リクエスト/s とスループットは低下する．これは，負荷が集中している 2 台のノードのサービス性能の和 (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である．マイグレーションを行うと 57 リクエスト/s とさらに低下していることがわかる．一方，提案手法であるレプリカアシストを用いた場合は 67 リクエスト/s と，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．

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．

櫛状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，負荷集中ノード（三角の山の頂上部分）でマイグレーションが起こり，その資源をマイグレーションに利用してしまう

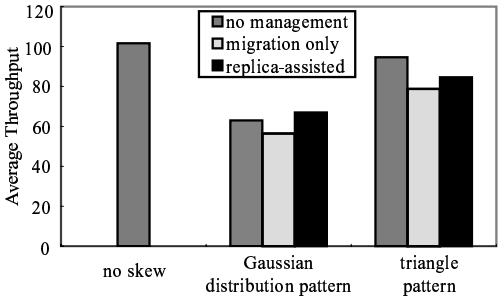

図 4: 各手法実行時の平均スループット
Fig.4 The avelage throughput of each methods

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間，当該ストレージ装置へのアクセスの一部を複製データへ回送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，複製配置に制約を持つストレージ技術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実証した．

今後の課題として，今回の実装ではマイグレーションがトークンのあるノード間のみであったので，並列データマイグレーションが可能な実装を用いた評価を行いたい．また複製へ回送される割合を負荷評価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る必要がある．さらに複製間一貫性制御や，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．

[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 Conf., 2002.

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

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

[4] H. Yokota, “Autonomous Disks for Advanced Database Applications”, Proc. of Intl. Symposium on Database Applications in Non-Traditional Environments(DANTE’99), p.p. 441-448, Nov, 1999.

[5] H.I. Hsiao and D.J. DeWitt, “Chained Declustering: A New Availability Strategy for Multiprocssor Database machines”, Proc. of the 6th ICDE, 1999.

[6] H.Yokota, Y.Kanemasa and J.Miyazaki, “Fat-Btree: An Update-Conscious Parallel Directory Structure”, Proc. of the 15th ICDE, pp.448-457, 1999.

[7] M.L. Lee, M. Kitsuregawa, B.C. Ooi, K. Tan, and A. Modal, “Towards Self-Tuning Data Placement in Parallel Database Systems”, Proc. of ACM SIGMOD conf., pages 225-236, 2000.

小林 大　Dai KOBAYASHI
平 15 東工大・工・情工卒．同大大学院・情報理工・計算工・修士課程在学中．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．

山口 宗慶　Munenori YAMAGUCHI
平 15 東工大・工・情工卒．同大大学院・情報理工・計算工・修士課程在学中．

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．