2005－DBS－137（Ⅱ）（69）
2005／7／14

Adaptive Overlapped Declustering: アクセス負荷分散と容量利用率向上のための複製管理戦略

渡邊明嗣[†]　横田治夫[††]

アクセス負荷分散と容量分散を両立する，値域分割ベースの分散ディレクトリとプライマリ・バックアップ型のデータ冗長化を組み合わせた分散ストレージシステムに適用可能なデータ配置手法 *Adaptive Overlapped Declustering* を提案する．既存手法は，分散システムの各ディスク間におけるアクセス負荷の偏り，または，データ量偏りの片方を抑えるが，提案手法はこの両方の問題を同時に解決する．また，提案手法は信頼性と可用性の向上にも役立つ．我々は，数学的モデルを用いた解析とキューを用いたシミュレーションによって，提案手法のデータ量偏りに対する効果を確認し，また，バックアップ復旧時間を従来手法の半分程度に抑えることを確認した．

Adaptive Overlapped Declustering: Replica Management Strategy for Improving Access Load Balance and Space Utilization

Akitsugu WATANABE [†] and Haruo YOKOTA[††]

This paper proposes a new data-placement method named *Adaptive Overlapped Declustering,* which can be applied to a parallel storage system using a value range partitioning-based distributed directory and primarybackup data replication, to improve the space utilization by balancing their access loads. While some data-placement methods capable of balancing access load or reducing data skew have been proposed, both requirements satisfied simultaneously. The proposed method also improves the reliability and availability of the system. We also shows efficiency of proposed method by using mathematical analysis and queuing simulations.

1. 序　　論

情報を扱うシステムの構築に於いて，ストレージシステムの可用性と規模拡張性は重要な問題である．これらの要求に対する解法として，複数のディスクを用いた分散ストレージ構成が広く利用されている．従って，データを冗長化し，データセットを分割して複数のディスクへ配置する手法が可用性と規模拡張性向上の要となっている．

冗長化戦略については，パリティを用いた戦略が持つ性能上および規模拡張性上の問題のため，大規模メディアを扱うアプリケーションにおいては，データ更新および故障後の復帰処理のコストが低く，しかも，システム再構成の規模拡張性にボトルネックが無い複製を用いた戦略が適しており，その長所を生かすためには複製を用いた戦略における容量効率を向上させることが必要と言える．

論理的なデータ領域を物理的な複数のディスクに割り当てる手法については，一般に，多くのアプリケーションはデータを水平分割する．水平分割戦略は，ラウンドロビン，ハッシュ，値域分割の 3 つに大別される[1)]．いずれも各々長所と短所を併せ持っている．ハッシュ分割では，データ再配置に際して全データのハッシュ値を再計算する必要があるため，システム再構成を考慮した場合の規模拡張性は制限される．一方，値域分割は偏りが生じるという短所を持つが，これはディスク間でデータを移動させることで解消できる．このような動的データ再配置は，アクセスパターンの変化やシステム構成の変化による偏りに対しても効果を発揮する．

アクセス性能の向上と値域分割に対する効率的な動的データ再配置を行うためには，適切な分散ディレクトリが必要である．Fat-Btree は，分散更新とデータ再配置を考慮した値域分割ベースの分散ディレクトリ

† 東京工業大学 大学院 情報理工学研究科 計算工学専攻
Department of Computer Science Graduate School of Information Science and Engineering in Tokyo Institute of Technology
†† 東京工業大学 学術国際情報センター
Global Scientific Information & Computing Center in Tokyo Institute of Technology

構造である[11]．しかし，横田が当初提案した偏り除去アルゴリズムはデータ量の偏りにのみ注目したものであり，アクセス負荷偏りに対して効果的では無かった．他方，アクセス負荷偏りに対する偏り除去手法には，データ量偏りを考慮していないという問題があった[4),7),8)]．

本稿の目的は，アクセス負荷とデータ量の偏りを同時に解消することを目的としたデータ配置戦略 *Adaptive Overlapped Declustering* の提案である．提案手法は複製を用いた手法の特長である高いサービス品質と，値域分割の特長であるレンジクエリやアクセス性能における利点を併せ持ち，また，非同期更新を用いることで更新操作のコストをも抑えている．のみならず，提案手法では，障害回復時のデータ転送量を抑え，さらに，隣接領域が並列に回復動作を行うことで，ディスク故障から速やかに回復する．これらの特長により，提案手法は分散ストレージシステムに高い可用性と規模拡張性を提供する．

2 章では非同期バックアップを用いた場合のアクセス負荷分散について述べる．3 章では，前章の議論を発展させ，新しい配置手法の提案とその特徴について述べる．4 章では相対容量利用率の比較を行う．5 章ではバックアップ復元の所要時間等の比較を行う．6 章では本研究の関連研究について述べる．7 章では本稿の結論と，将来の課題について述べる．

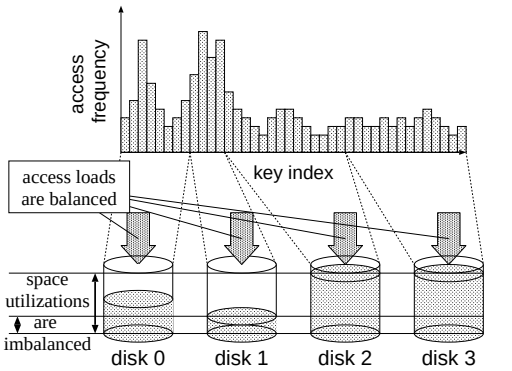
2. アクセス負荷の分散

アクセス負荷と容量利用率の同時均等分配について検討する前に，値域分割のみを用いた場合，および，プライマリ・バックアップ型のデータ冗長化である chained declustering (以下 CD)[5)] のみを用いた場合の負荷分散手法を検討する．

2.1 値域分割とアクセス負荷分散

各ディスクの容量利用率が釣り合っている場合でも，偏ったアクセス分布などの要因によってアクセス負荷は偏りうる．データを高負荷状態のディスクから低負荷状態のディスクに移動させることによってアクセス負荷の釣り合いを取ることができる．値域分割では，データの並び順を保持する必要があるため，データ移動は隣接するディスク間でのみ行われる．

図 1 は，アクセス負荷の釣り合いを取るためのデータ移動を適用した結果，容量利用率の釣り合いが崩れた状態を示している．図の上部はアクセス頻度，下部は容量利用率を表している．中央の点線は値域とディスクとの間の対応を示している．最下部に示した数字は容量利用率を%表記で示したものである．このように，アクセス負荷の釣り合いを取るためのデータ移動は容量利用率の釣り合いを崩す可能性がある．



	disk 0	disk 1	disk 2	disk 3
P	24%	12%	42%	42%
total	24%	12%	42%	42%

図 1 アクセス負荷均衡化によって生じた容量利用率偏り
Fig. 1 Data skews to balance access load for a skewed access pattern

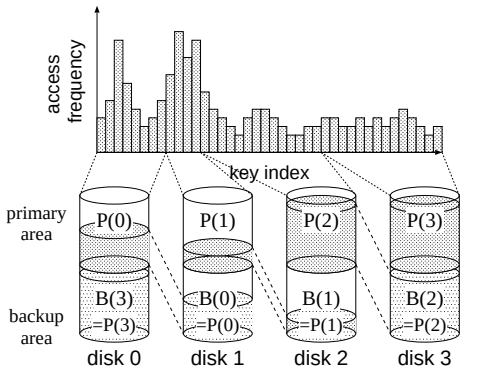
2.2 Chained declustering とアクセス負荷分散

次に，システムの可用性について検討する．高いアクセス性能と単一ディスク故障への耐性を両立させるという目的に対しては，CD ベースの複製を用いた障害対策と値域分割を組み合わせる手法が有効である[10)]．

本稿では，議論を簡単にするため複製数を 2 に限定する．以下の議論では，先行書き込みログを利用した非同期更新が行われることを仮定する．このような非同期更新では，バックアップコピーがいわゆるダーティーコピーとなりうるため，アプリケーションの目的によって各複製へのアクセス頻度が変化しうる．この事を定式化して表すため，本稿ではプライマリコピーから読み出される割合を ψ $(0 < \psi \leq 1)$ とおき，アプリケーションのアクセスに偏りがあり，かつ，ダーティーコピーの利用頻度が低い $(\psi \simeq 1)$ 場合のみを検討する．

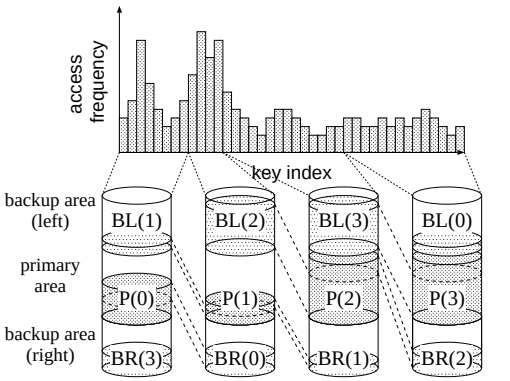
図 2 は，非同期更新を前提とした CD ベースのデータ配置の例である．n 番目のバックアップ領域 B(n) は隣接ディスクに格納された n 番目のプライマリ領域 P(n) の複製である．P(n) と B(n) は異なるディスクに格納されるため，いずれかのディスクが故障した場合でもデータは失われず，サービスを継続することができる．この配置法では隣接する 2 つのディスクが同時に故障しない限り，データ破損が生じない．

データ偏りの観点から見ると，たしかに CD にはデータ偏りを減らす効果がある．しかし，複製なしの場合 (図 1) と CD(図 2) における，同じアクセスパターンにおける容量利用率の違いを示した例にも表れているように，その効果は十分ではない．



	disk 0	disk 1	disk 2	disk 3
P	24%	12%	42%	42%
B	42%	24%	12%	42%
total	66%	36%	54%	84%

図 2 chained declustering におけるデータ偏り
Fig. 2 Data skews under chained declustering



	disk 0	disk 1	disk 2	disk 3
BL	6%	21%	21%	12%
P	24%	12%	42%	42%
BR	21%	12%	6%	21%
total	51%	45%	69%	75%

図 3 適合的バックアップ分割によって均衡化された容量利用率とアクセス負荷
Fig. 3 Data and access load balance with adaptive backup division

3. Adaptive Overlapped Declustering

アクセスが偏っている場合の容量利用率を向上させるため，本稿では *Adaptive Overlapped Declustering* (以下 *AOD*) というデータ配置手法を提案する．提案手法は CD を発展させたもので，データ量の偏りを減らし，バックアップコピーの再生成時間を減らすことで可用性をも高める．

提案手法では，バックアップ領域を 2 つに分割し，隣接する 2 つのディスクに各 1 つずつを格納する．しかる後に，分割されたバックアップ領域間の境界を調整することで，この 2 つのディスク間のデータ量を操作し，容量利用率を均衡化させる．この分割はアクセス偏りおよび容量利用率を均衡させやすくし，しかもバックアップコピーの再生成時間をも減らすものである．

3.1 適合的バックアップ分割

多くのアプリケーションではダーティーコピーを利用しない ($\psi \simeq 1$)．したがって負荷分散の観点から見ると，バックアップの配置はプライマリの配置ほど重要では無い．バックアップをプライマリとは独立に配置することが可能であれば，プライマリの配置を変えることでアクセス負荷を分散し，バックアップの配置を変えることでデータを分散することができるようになるだろう．CD[5] ではプライマリ領域 P(n) のバックアップ B(n) を片方の隣接ディスクにのみ格納していたが，提案手法では， これを BR(n) と BL(n) の 2 つに分割して，それぞれ右と左の各隣接ディスクに格納し，容量利用率の偏りを減らす (図 3)．分割条件 C_a は以下の通りである：

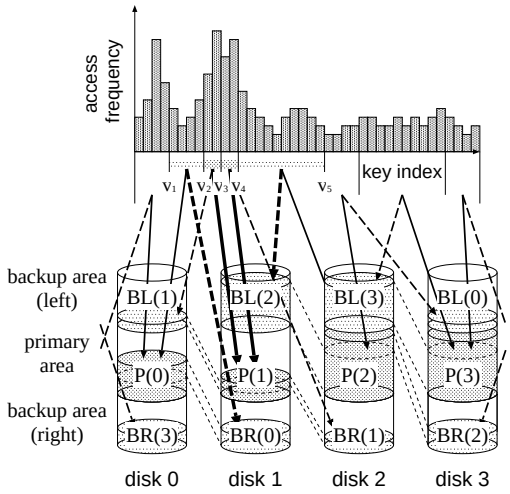


図 4 ディスクに割り当てられた値域とその重なり
Fig. 4 An overlapped continuous range on a disk

(1) $\mathrm{P}(n) \equiv \mathrm{BR}(n) \cup \mathrm{BL}(n)$
(2) $\mathrm{BR}(n) \cap \mathrm{BL}(n) \equiv \emptyset$
(3) BR(n) と BL(n) は $P(i)$ をキー $k(i)$ で分割したもの

この分割によって，BR(n) と BL(n) との間でデータを融通できるようになる．

図 3 は，図 2 と同じ条件のものに対してこのバックアップ分割を適用した例である．この例では，他の手法を用いた場合と同様にアクセス負荷を平均化しつつ，容量利用率も完全に平均化されている．

適合的バックアップ分割は容量利用率の完全な平均化を保証しない．しかし，ほとんどのアクセス分布に対して，この手法は大いに有効である．有効性についての検討は 4 節で行う．

ある 1 つのディスクに格納されているデータアイテムの値域に注目すると，両脇を 2 つのバックアップ領域で挟まれたプライマリ領域という連続した値域が各ディスクに割り当てられることになる（図 4）．これらの連続した値域はディスク間で重なり合っている．そのため，アクセス負荷均衡化のためのデータ移動操作を，対応する 1 対の複製を同一ディスク内のプライマリあるいはバックアップ領域に割り当てなおすという一連の低コストの操作群で代用できる．

この手法は Teradata の interleaved declustering[9] とバックアップを分割するという点で類似しているが，分割されたバックアップの配置方法に大きな違いがあり，そのため，得られるデータの生存性の規模拡張性が大きく異なる．Interleaved declustering は連続する 2 つのディスク故障に対して無防備だが[5]，この手法では CD 同様，隣接する 2 つのディスク故障が起きない限り，データは消失しない．

3.2 データ移動アルゴリズム

データ量およびアクセスパターン変化時にはデータ移動が必要となる．ここでは，容量利用率均衡化のためのディスク間データ移動アルゴリズムとディスク内データ移動アルゴリズム，および，アクセス負荷均衡化のためのアルゴリズムについて述べる．無論，これらのアルゴリズムは併用できる．

3.2.1 負荷移動のためのディスク間データ移動

システムには，データ量およびアクセス負荷の偏りを検出する何らかの機構が既に備わっていると仮定し，本稿ではその詳細については触れない．アクセス負荷偏り検出機構が出力する最も負荷が高いディスクを (h)，期待されるデータの移動先を (d)，必要なデータ移動量を (m) とおく．ただし，配置ルールの制約を考慮した移動先が選ばれるものとする．

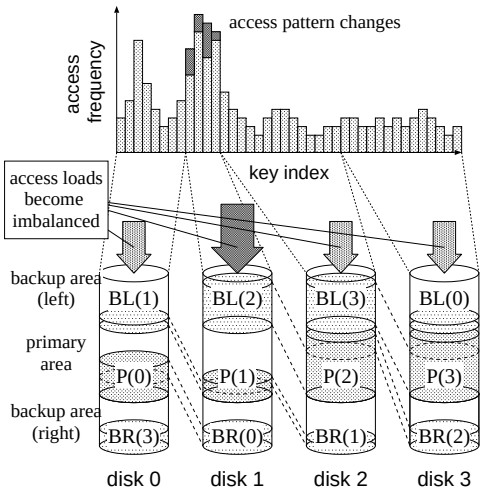
ディスク間データ移動は CD におけるデータ移動と同様のアルゴリズムによって処理される．次節で述べるディスク内データ移動の場合と同様に，移動元のバックアップの量が十分であれば各ディスクの容量利用率は変化しない．

3.2.2 負荷移動のためのディスク内データ移動

ここでは，3.1 節で述べたアクセス負荷均衡化のためのデータ移動操作を，より低いコストで行うアルゴリズムについて述べる．

図 3 の状態から図 5 の状態へとアクセスパターンが変化した場合を例に，図 6 に示したディスク内データ移動アルゴリズムの動作を説明する．P(1) にあるデータアイテムは同じディスク 1 上に格納された BR(1) に再割り当てされ，同時に BL(1) にあるデータアイテムは同じディスク 0 上に格納された P(0) に再割り当てされる．

BL(1) に十分な量のデータアイテムがあるならば，これらの操作はネットワーク帯域を一切消費しない．しかも，これらの操作はデータアイテムのメタデータを書き換えるだけで実現可能なので，実際のデータ

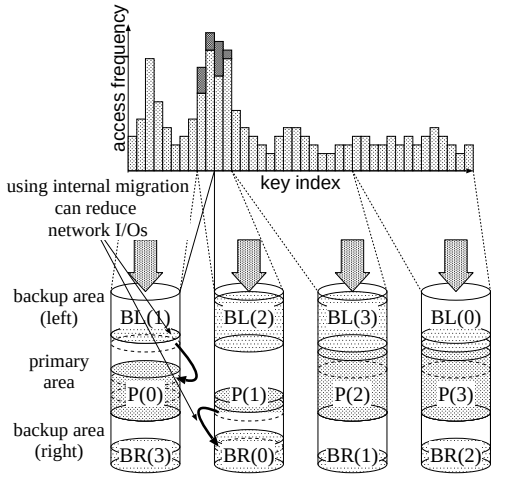


	disk 0	disk 1	disk 2	disk 3
BL	6%	30%	12%	6%
P	24%	12%	42%	42%
BR	30%	18%	6%	12%
total	60%	60%	60%	60%

図 5 偏り除去前のアクセス負荷が偏っている状態
Fig. 5 Load transitions before balancing

```
Intra_Disk_Migrate (h, m, d)
begin
  if d= h− 1 mod N then
    Let i= min (m, cardinality(BL(h));
    Migrate idata items from P(h) to BR(d);
    Migrate idata items from BL(h) to P(d);
    if (i< m) then
      Migrate (m− i) data items from P(h) to P(d);
      Migrate (m− i) data items from BR(h) to BR(d);
  else
    Let i= min (m, cardinality(BR(h));
    Migrate idata items from P(h) to BL(d);
    Migrate idata items from BR(h) to P(d);
    if (i< m) then
      Migrate (m− i) data items from P(h) to P(d);
      Migrate (m− i) data items from BL(h) to BL(d);
end
```

図 6 ディスク内データ移動アルゴリズム
Fig. 6 Intra-disk data migration algorithm



	disk 0	disk 1	disk 2	disk 3
BL	0%	30%	12%	6%
P	30%	6%	42%	42%
BR	30%	24%	6%	12%
total	60%	60%	60%	60%

図 7 ディスク内データ移動によるアクセス負荷分散
Fig. 7 Load balancing by intra-migration

I/O そのものも減らすことができる．図 7 が示すように，この操作は各ディスクのデータ量を変化させない．BL(1) に十分な量のデータが無かった場合は，BL(1) から BR(0) へデータアイテムを移動した後に，BR(1) から BR(0) へデータアイテムを移動させる．この操作はこの 2 つのディスクのデータ量を変化させる．

3.2.3 データ量移動のためのディスク間データ移動

AOD はデータ挿入や削除などによって容量利用率のバランスが変化した場合にも有効に働く．このような場合には，アクセス負荷の均衡を破らないように，プライマリデータセットの配置を維持したまま，バックアップデータセットのみを移動させて容量利用率を調節する．図 8 はこの容量利用率調整アルゴリズムの概要である．

3.3 ディスク故障時の処理

CD 同様 AOD でも，単一ディスク故障でデータアイテムが失われることは無い．また，互いに隣接する 1 組のディスクが同時に故障しない限り，複数のディスク故障に対しても耐性がある．簡単のため，ここでは単一ディスク故障のみを扱うが，非隣接ディスク群の同時故障も同様に取り扱える類の問題である．

本節では，まずディスク故障による負荷分布の変化を考察し，次いでプライマリ-バックアップ対の復元手法を記述する．

Balancing_Data *(*h, m*)*
begin
Let ls *be the list of data amount on each disk;*
Let i= m*;*
Let a= *(*ls*[*h*]* + ls*[*h*−2* mod *N]* + ls*[*h*+2* mod *N])/3 ;*
Migrate min*(cardinality(BL(*h*+1* mod *N)),* a-ls*[*h*−2* mod *N*], i*)*
*data items from BL(*h*+1* mod *N) to BR(*h*+1* mod *N);*
Subtract the number of migrated data items from i*;*
Migrate min*(cardinality(BR(*h*−1* mod *N)),* a−ls*[*h*+2* mod *N*], i*)*
*data items from BR(*h*−1* mod *N) to BL(*h*−1* mod *N);*
Subtract the number of migrated data items from i*;*
end

図 8 データ量移動のためのディスク間データ移動アルゴリズム
Fig. 8 Data volume balancing algorithm

3.3.1 ディスク故障による負荷分布の変化

ディスク 1 が故障した場合を例に考察する．CD の場合，ディスク 2 に格納されたバックアップデータセット B(1) が P(1) の役割を引き継ぐことになる．その結果，ディスク 2 の負荷は元の 200%に跳ね上がる．

AOD のバックアップ分割はこの負荷の急激な変動を抑える．同様の状況の場合，ディスク 0 に格納された BL(1) とディスク 2 に格納された BR(1) が P(1) の役割を引き継ぎ，それぞれの負荷の上昇は 150%に留められる．

プライマリ-バックアップ対を復元する前にディスク 0 または 1 が故障すると，P(1) のデータは失われる．よって，可能な限り短い時間でプライマリ-バックアップ対を復元する必要がある．

3.3.2 プライマリ-バックアップ対の復元

予備ディスクが用意されている場合，故障したディスク 1 は予備ディスクに置き換えら，元の状態に復帰するようにデータが複製される．この場合においては，CD と AOD の間に差異は無い．

予備ディスクがすぐに用意できない場合，故障を免れたディスク群に格納されたデータを用いてプライマリ-バックアップ対を可能な限り速やかに復元しなければならない．このプライマリ-バックアップ対の復元はバックアップの昇格とバックアップの再生成の 2 つの段階を踏んで行われる．

CD のバックアップ昇格段階では，ディスク 2 の B(1) がプライマリに昇格し，P(2) と結合されて，新しい P(2) を構成する．しかる後に，P(2) の昇格した部分がディスク 3 に複製され，B(2) と結合されて，新しい B(2) を構成する．同時に，P(0) がディスク 2 に複製され，新しい B(0) として格納される．複製後，偏り除去機構によりデータ移動が発生する．

AOD では，この昇格がディスク 0 とディスク 2 で

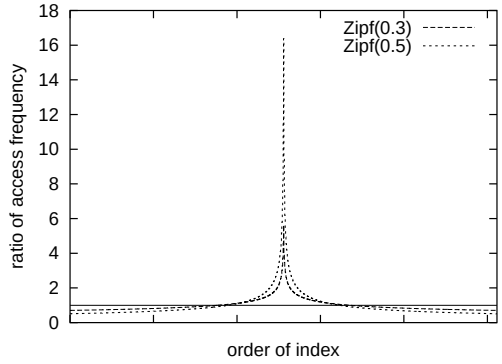


図 9 パターン A
Fig. 9 Access pattern A

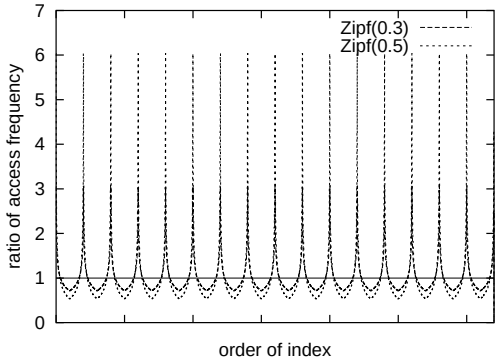


図 10 パターン B
Fig. 10 Access pattern B

並列に実行される．しかも，バックアップ再生成段階においては，BL(0)，BR(0)，BL(2)，BR(2) のコピーもまた並列に行われる．また，再生成段階で偏りを考慮した配置が行えるので，全体のネットワーク帯域利用量とディスク I/O を抑えることができる．

AOD によるプライマリ-バックアップ対復元処理の性能改善については，5 節で検証する．

4. 容量利用率の解析

アクセス負荷を均衡させた状態での容量利用率の均衡化の限界を調べるために，ディスク数 128 のシステムにおける相対未使用領域率を，$\psi = 1$ であるような偏りのあるアクセスパターンについて調べた．指標として用いた相対未使用領域率は，少なくとも 1 つのディスクが満杯になるまでデータを詰め込んだ時の，システム全体の未使用領域率である．

パターン A(図 9) はアクセスが中央の 1 点に集中しているアクセスパターンである．パターン B(図 10) はアクセスが 16 箇所に分散しているアクセスパターンである．いずれのパターンにおいても，アクセス頻度の分布は Zipf 分布[6] に従う．

図 11 は提案手法と CD における相対未使用領域率を比較したものである．偏りの大きさ，分布パターンによらず，提案手法の方がよい結果を収めている．

アクセスパターン A では，偏りが小さい場合に大きな差が見受けられる．アクセスパターン B では，よりはっきりとした差が現れている．偏りが小さい場合，すなわち，偏りの係数が 0.2 未満の場合に，提案手法は容量利用率を完全に均衡させている．

偏りが大きい場合には，AOD も容量利用率を完全に均衡化させることはできないが，そのような場合でも常に未使用領域を従来手法より小さく抑えている．

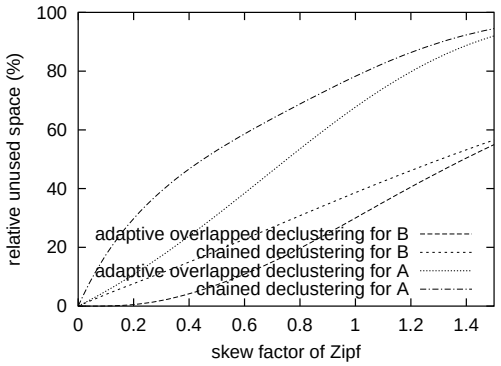


図 11 アクセス負荷偏り除去後の相対未使用領域率
Fig. 11 Relative unused storage space ratio after balancing skews

5. キューモデルに基づいたシミュレーション

本節では，キューモデルに基づいたシミュレーションを用いて測定したプライマリ-バックアップ対の復元に要する時間を CD と AOD の双方について比較する．簡単のため，以下の条件を設ける:

- リクエストは密度関数に従って独立に発生し，他のリクエストの影響を受けない
- 各ディスクはプライマリコピーの数に比例する量のアクセスリクエストを受信する
- 初期状態では，プライマリコピーおよびバックアップコピーのいずれも，各ディスクに均等に割り振られているとする
- キューは十分に長いとする
- サービス時間はある定数で表現されるものとする
- データ移動操作は移動元と移動先で独立に処理されるリクエスト対であるものとして扱う

実験の諸元を表 1 に示す．

図 12 は対故障処理中の平均レスポンス時間を示したものである．プライマリ-バックアップ対の復元を

表 1 実験の諸元
Table 1 Parameters used for the analyses

Parameter	Value
Number of PEs (N):	8
Number of primary copies :	98304
Service time:	0.133sec
Mean time between requests:	0.0625sec
Probability density function of request:	$\lambda e^{-\lambda x}$
Maximum frequency of content migration:	40/min

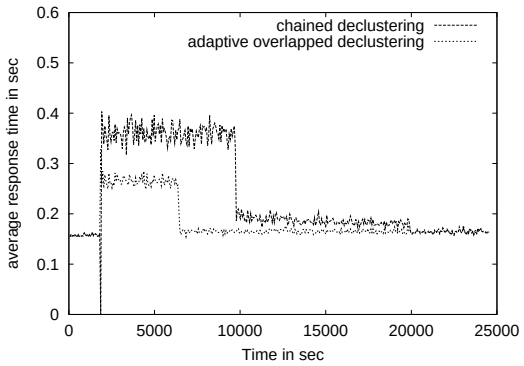


図 12 対故障処理中の平均レスポンス時間
Fig. 12 Average response time during failure handling

行っている期間は，グラフに矩形状の盛り上がりとして現れている．提案手法では，矩形状の盛り上がりの高さは従来手法の半分程度である．これは，復元作業が 2 台のディスクに分割されたことで，復元作業に参加するディスクの負荷が予測どおり軽減されたことを示している．また，提案手法では矩形状の盛り上がりの幅は従来手法の半分より少し大きい程度である．これは，バックアップ領域の分割によって，復元作業に参加するディスクの復元作業に要する総作業量が軽減されたことを示している．

これらの結果が示すように，AOD は CD ベースの手法の半分程度の時間と負荷上昇で，故障から復帰できる．すなわち，提案手法は高い可用性を必要とするシステムに不可欠な特長を備えている．

6. 関 連 研 究

障害からの回復を速めるため，複製を用いた戦略が研究されてきた．Gamma[2] で用いられている chained declustering[5] については前述の通りである．Teradata の interleaved declustering は，ディスク故障からすぐにデータ量とアクセス負荷のバランスをとることができる[9] が，データの生存性に規模拡張性が無いため大きなシステムでは対故障性能を期待できず，また，ハッシュを用いた分割しか利用できないという欠点がある．

動的偏り制御の分野では様々な研究が行われている．Scheuermann らは効果的な偏り除去においてはデータ移動のコストそのものも考慮すべきであると述べている[8]．また，Btree ベースのディレクトリを備えたシステムでは部分木単位での BULK ページ移動による高速なデータ移動手法が可能であることが知られている[7]．RING 手法では，Btree ベースのディレクトリの両端をつなげることで，アクセス負荷均衡に必要なデータ移動量を減らしている[3]．

7. 結　　論

本稿では，アクセス負荷とデータ量の偏りを同時に解消するデータ配置戦略 *Adaptive Overlapped Declustering* を提案した．提案手法では，プライマリ領域を格納したディスクに隣接した 2 つのディスクに適合的バックアップ分割手法によって分割されたバックアップ領域を格納する．その結果，各ディスクは 2 つのバックアップ領域と 1 つのプライマリ領域からなる連続した値域を割り当てられることになる．2 つのバックアップ領域の大きさを調整する適合的バックアップ分割は，アクセス負荷を均衡させつつ領域利用率の偏りを従来手法よりも小さく抑えるという目的に対して効果的である．負荷移動のためのディスク間／ディスク内データ移動とデータ移動のためのディスク間データ移動を組み合わせることで，アクセス分布やデータセットが変化した場合でも，アクセス負荷とデータ量の偏りを同時に解消することができる．また，バックアップを分割したことによって，ディスク故障時のプライマリ-バックアップ対再生成に因する負荷上昇を，従来手法の 200%から 150%に抑制している．これによってプライマリ-バックアップ対再生成の時間短縮が為され，システムの可用性は向上する．

今回行ったシステムの未使用領域の解析は，提案手法がデータ量の偏りを完全に 0 にしてしまうか，あるいは，少なくとも従来手法よりも小さな偏りに抑えることを証明した．また，バックアップ再生成時間の比較実験は，AOD が短時間，かつ，少ないシステム性能に対する悪影響でバックアップを復元できることを示した．

今後の展望だが，Fat-Btree の特性に特化した仕組みを組み上げることでさらなる性能の向上を図り，また，実機実験を行ってより精密な比較を行う予定である．

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，文部科学省科学研究費補助金特定領域研究 (16016232)，情報ストレージ研究推進機構 (SRC)，ＮＨＫ放送技術研究所の助成により行な

われた.

参 考 文 献

1) DeWitt, D. and Gray, J.: Parallel Database Systems: The Future of High Performance Database Systems, *Communications of the ACM*, Vol.35, No.6, pp.85–98 (1992).
2) DeWitt, D. J., Gerber, R. H., Graefe, G., Heytens, M. L., Kumar, K. B. and Muralikrishna, M.: GAMMA - A High Performance Dataflow Database Machine, *VLDB'86 Twelfth International Conference on Very Large Data Bases, August 25-28, 1986, Kyoto, Japan, Proceedings* (Chu, W. W., Gardarin, G., Ohsuga, S. and Kambayashi, Y., eds.), Morgan Kaufmann, pp.228–237 (1986).
3) Feelifl, H. and Kitsuregawa, M.: RING: A Strategy for Minimizing the Cost of Online Data Placement Reorganization for Btree Indexed Database over Shared-nothing Machines, *DASFAA 2001*, IEEE Computer Society (2001).
4) Feelifl, H., Kitsuregawa, M. and Ooi, B.-C.: 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).
5) Hsiao, H.-I. and DeWitt, D.: Chained Declustering: A New Availability Strategy for Multiprocessor Database Machines, *Proceedings of 6th International Data Engineering Conference*, pp.456–465 (1990).
6) Knuth, D. E.: *Sorting and Searching*, Addison-Wesley Publishing Company (1973).
7) Lee, M. L., Kitsuregawa, M., Ooi, B.-C., Tan, K.-L. and Mondal, A.: Towards Self-Tuning Data Plaement in Parallel Database Systems, *SIGMOD Record*, Vol.29, No.2, pp.225–236 (2000).
8) Scheuermann, P., Weikum, G. and Zabback, P.: Adaptive Load Balancing in Disk Arrays, *Foundations of Data Organization and Algorithms*, pp.345–360 (1993).
9) Teradata Corp.: *DBC/1012 Database Computer System Manual Release 2.0*, document no. c100001-02 edition (1985).
10) Yokota, H.: Autonomous Disks for Advanced Database Applications, *Proc. of International Symposium on Database Applications in Non-Traditional Environments (DANTE'99)*, pp.441–448 (1999).
11) Yokota, H., Kanemasa, Y. and Miyazaki, J.: Fat-Btree: An Update-Conscious Parallel Directory Structure, *Proc. of 15th Int'l Conf. on Data Engineering*, pp.448–457 (1999).