社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

オンライン構造劣化モニタを用いたデータベース再編成支援

星野　喬† 　合田　和生†† 　喜連川　優††

† 東京大学大学院 情報理工学系研究科 〒113-0033 東京都文京区本郷 7-3-1
†† 東京大学 生産技術研究所 〒153-8505 東京都目黒区駒場 4-6-1
E-mail: †{hoshino,kgoda,kitsure}@tkl.iis.u-tokyo.ac.jp

あらまし　これまでに，我々はデータベース構造劣化を低オーバーヘッドに測定する手法を構築し，構造劣化をリアルタイムに可視化できるオンライン構造モニタを提案した．当該モニタは管理者による構造劣化の分析及び再編成計画の作成に有用である．本稿では，再編成ポリシに従って構造劣化領域を同定するデータベース再編成支援機能を構築し，構造劣化モニタ上に実装した．

キーワード　データベース構造劣化，データベース再編成

Database Reorganization Assistance with Online Structural Deterioration Monitor

Takashi HOSHINO†, Kazuo GODA††, and Masaru KITSUREGAWA††

† Graduate School of Information Science and Technology, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-0033, Japan
†† Institute of Industrial Science, University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505, Japan
E-mail: †{hoshino,kgoda,kitsure}@tkl.iis.u-tokyo.ac.jp

Abstract　So far, we made a method to measure structural deterioration of database with low overhad, and online structural deterioration monitor that can visualize structural deterioration in real time. The monitor is very useful for database administrator to analyze structural deterioration and create database reorganization plan. In this paper, we constructed an assistance function for database reorganization, which detects structurally-deteriorated portions following a given reorganization policy and implemented it into the monitor.

Key words　Database, Structural Deterioration, Database Reorganization

1. はじめに

近年，様々な分野において運用されているデータベースが大容量化している．そのため，データベースシステムの管理はより困難になり，人件費等のコストが増大している．データベースシステム常時運用ニーズもまた増大しており，状態監視および管理タスク実行に費やすことのできる時間やシステムリソースが少なく，データベースシステム管理をさらに難しいものにしている．また，システムにおける人的操作ミスが，甚大な被害を引き起こす事例も増えており，人間の管理者のみによるきめの細かい管理に限界が見えてきた．この問題を解決するため，管理の自立化を目指す試みが増えている [3].

我々は データベース再編成管理に着目し，その自立化を目的としている．データベース表空間内のデータは，更新操作を繰り返すことによりその構造が劣化し，本来想定している配置と大きく乖離した状態となり，性能を大幅に低下させる場合がある (図 1). このため，データベース管理者は再編成を行うことにより，セカンダリストレージ内のデータを再配置し，性能低下を予め防ぐ必要がある．現状では，構造劣化を同定し再編成を実施するべきか否かの判断は難しく，また，再編成自体にかかる時間は膨大であることから，再編成は高度な管理業務と考えられており，再編成を自立化させることの意義は大きい．

我々はこれまでに，構造劣化を，低オーバヘッドで監視することのできるデータベースオンライン構造劣化モニタを構築した [9]〜[11] ．当該モニタを用いることで，構造劣化に起因する性能低下を他の性能低下要因と切り分けることができるだけで

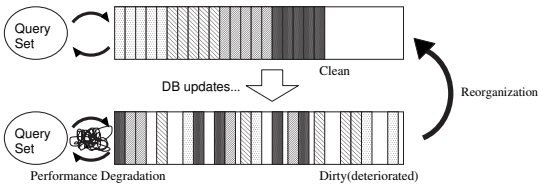

図 1　構造劣化と再編成

なく，管理者の構造劣化分析および再編成計画作成を容易化できる．

再編成管理においては，構造劣化の状況を把握した後に，性能要求などの再編成ポリシに合わせて，再編成対象領域および再編成実施時期などを決定する必要がある．

再編成管理をより容易化するため，また，再編成スケジュール作成の自立化に向けて，本稿では，オンライン構造劣化モニタを用いた再編成支援機能を構築した．本支援機能は，ユーザから与えられた再編成ポリシを用いて，構造劣化が許容範囲を越えるデータベース領域を同定し，オンライン構造劣化モニタに再編成対象領域として表示する機能を持つ．本支援機能を用いることで，単純な再編成ポリシによる管理者の再編成意思決定支援が可能になり，再編成管理がより容易になることが期待される．

本稿は以下のように構成される．まず，第 2 章でオンライン構造劣化モニタについて述べ，次に，第 3 章で再編成支援機能について述べる．その後，第 4 章でケーススタディを用いた評価を行う．第 5 章で関連研究について述べる．第 6 章で結論と今後の課題を述べる．

2. オンライン構造劣化モニタ

本節では，オンライン構造劣化モニタが監視する対象である構造劣化度分布について，その定義，集約による解像度変更手法，及び，その差分計算手法を述べる．

2.1 構造劣化度分布

構造劣化を定量的に表す指標として，構造劣化度を定義し，その分布について述べる．

まず，論理的なレベルにおけるデータベースアクセスパタンを仮定する．例えば，ある属性における範囲を指定してデータレコードを取得する範囲検索などが挙げられる．

論理アクセスパタンとデータベース構造が与えられたとき，現在の状態におけるアクセスコストを考える．また，データベース再編成によって，データ再配置が行われた理想的な状態におけるアクセスコストを考える．両者の比を，当該論理アクセスにおける構造劣化度と定義する．例えば，100,000 レコードか成る表があり，ある時点の構造状態で 0 から 30,000 までのレコードを取得するために 30 秒かかるとする．再編成直後の理想的な状態において，15 秒で済むことが分かったとする．このとき，当該時点での 0 から 30,000 のデータに対する範囲検索に関する構造劣化度は，2 である．同様に，30,000〜100,000 までのデータに対して，それぞれ 50 秒，20 秒である場合，当該データの構造劣化度は 2.5 である．

細粒度のアクセスに対する構造劣化度を各データに対して把握できた場合，データベース論理空間全体に対してこれを分布として扱うことが出来る．これを構造劣化度分布と呼ぶ．例えば，B+木構造における索引構造を利用した範囲検索に関する構造劣化度分布は，B+木構造の葉ページの鍵順序に見たものを論理空間とし，当該論理空間における各ページの範囲走査時の IO コストと，シーケンシャルスキャン時の IO コストとの比を各ページに対する構造劣化度をすると，空間全体に対する構造劣化分布が得られる．ちなみに，このケースでは，構造劣化のない状態とは，ある論理アクセスを仮定した論理空間におけるデータの論理順序が，セカンダリストレージにおける実際のデータの物理順序と一致している状態を指す．この場合，特に今日のハードディスクドライブを用いたセカンダリストレージにおいては，ランダムアクセスに比べて最大で数十倍も高速なシーケンシャルアクセスが期待できる．ただし，各データベース構造は，特定の論理アクセスパタンを高速化するために特化されている場合が多いため，すべての論理アクセスパタンに対して構造劣化のない状態は一般には存在しない．例えば，B+木構造に二次索引が存在するとき，二次索引の属性から見ると，定義からは構造劣化がない状態とは言えない．このため，ここでは，データベース構造が高速化のために想定する論理アクセスパタンを考えることとする．

上記で例に挙げた B+木構造における範囲検索に対する構造劣化度分布は以下のように定義することが出来る．範囲検索時における根ページから枝ページの走査に必要なアクセスコストは，範囲幅が大きい範囲検索時においては小さいため無視する．各葉ページを p_i $(0 \leqq i < N)$ とする．p_i はセカンダリストレージ上の物理位置すなわちアドレス a_i，保持しているデータレコードの中での最小鍵値 k_i を持つ $(k_i \geqq k_{i-1})$．各ページに対する構造劣化度は，範囲検索を想定しているため，p_i が読み込まれる直前は p_{i-1} が読み込まれるものと期待し，論理シーク量 $a_i - a_{i-1}$ に対するセカンダリストレージにおける IO コスト c_i をストレージ性能モデル ST を用いて $c_i = ST(a_i - a_{i-1})$ のように計算し，ページ p_i の構造劣化度分布 c_i/c_{seq} を得る．ここで，c_{seq} は，シーケンシャルスキャン時のページの IO コストである．論理空間は，i(もしくは k_i) で表され，物理空間は，a_i で表わされる．詳しくは，論文 [9] に述べている．

2.2 構造劣化度分布の集約

先述した構造劣化度分布を用いることで，細粒度に構造劣化を把握することが出来る．ただし，当該分布を詳細に把握しようとすればするほど，当該分布を保持するストレージ容量は増大し，その分析コストは高くなる．上記で定義した構造劣化度分布は可能な限り最も高解像度に分布を取得するモデルである．このため，データベースサイズが増加すれば，構造劣化度分布の保持に必要なストレージも線形的に増加する．この場合，解像度を下げてでも，ストレージ容量の節約や，分析の高速化を図る必要が生じる．例えば，大容量データベースを監視したいが，様々な分析を高速に行いたい場合は，構造劣化度分布を集約してメインメモリ上で保持すれば良い．

ここでは，例として B+木の範囲検索における構造劣化度分布の集約手法について述べる．$\{p_i\}$ において，i を軸にした論理空間で連続するページをまとめて一つの要素とし，その構造劣化度を算出する．当該要素を logical bulk (lb) と呼ぶことにし，以下のように定義する．$j(0 \leqq j < N_{lb})$ 番目の lb を

lb_j とし，ページ p_{i_j} から連続する n'_j ページ分の情報を集約したものとする．n'_j は必要な解像度に合わせて自由に設定できる．$i_j = \sum_{j_0=0}^{j_0=j-1} n'_{j_0}$ ただし $i_0 = 0$ であり，$N = \sum_{j=0}^{N_{lb}-1} n'_j$ である．lb_j は，IO コストの和 $c'_j = \sum_{i=i_j}^{i_j+n'_j-1} c_i$ ，最小鍵値 $k'_j = k_{i_j}$，及び，n'_j を保持する．このとき，lb_j における構造劣化度は，$c_j/(n_j c_{seq})$ である．

$\{lb_j\}$ の分布は，$\{p_i\}$ の分布の集約であり，粗い情報を保持していると言える．全ての j に対して $n_j = 1$ である場合，$\{lb_j\} = \{p_i\}$ である．

2.3 構造劣化度分布の差分計算

前節で集約された構造劣化度分布が与えられたとき，データベース構造の更新差分を用いて当該分布をインクリメンタルに最新の状態に追随させる手法について，詳しくは論文 [10] に述べているため，ここでは概要のみ述べる．

B+木の範囲走査を対象にしている場合，葉ページにおけるページ分割時のページ追加，及び，ページ結合時のページ削除において，データベース構造が変化する．構造劣化度分布をメンテナンスするために，ページ追加もしくは削除されたページ p_{midd} およびその両隣のページ p_{prev}, p_{next} の鍵値 k とアドレス a が更新差分情報として必要である．

$\{p_i\}$ において，構造変化時に IO コスト c_i が変化するのは，p_{midd} と p_{next} のみである．これらの差分計算に必要なページアドレスは，$a_{prev}, a_{midd}, a_{next}$ である．鍵値 k の値は，構造劣化分布の中で，該当する p_i もしくは，lb_j を同定するために用いる．

$\{lb_j\}$ に対する差分計算は，どの lb に各ページ $p_{prev}, p_{midd}, p_{next}$ が含まれているかで変わる．例えば，3 つ全てがある lb に含まれているならば (これは鍵値 k の比較で判定できる)，全ての差分計算を当該 lb に対して行えば良い．具体的に，lb_j 内に含まれている，すなわち，$k'_j \leqq k_{prev}$ かつ $k_{next} < k'_{j+1}$ である場合，ページ追加時は，

$$\begin{aligned} c'_j \quad += \quad & -ST(a_{next} - a_{prev}) \\ & +ST(a_{midd} - a_{prev}) \\ & +ST(a_{next} - a_{midd}) \\ n'_j \quad += \quad & 1 \end{aligned}$$

，ページ削除時は，

$$\begin{aligned} c'_j \quad += \quad & -ST(a_{midd} - a_{prev}) \\ & -ST(a_{next} - a_{midd}) \\ & +ST(a_{next} - a_{prev}) \\ n'_j \quad -= \quad & 1 \end{aligned}$$

のように差分計算で求めることが出来る．

ただし，同じページが連続して分割され，ある lb_j における n'_j が極端に大きくなってしまった場合，監視したい解像度よりも低い解像度しか得られない現象が局所的に発生してしまう．解像度をある程度の範囲内に抑えるために，各 lb_j の n'_j の大きさを特定の範囲に制限することが必要になる．例えば基準となる値 n'_{base} を与えたとき，全ての j に対して，$n'_{base}/2 \leqq n'_j \leqq 2n'_{base}$ を保つようにするなどである．解像度が高すぎる場合は，隣り合う lb をさらに集約すれば良いが，解像度が低すぎる場合は，詳細情報を再取得する必要が生じる．この詳細情報をデータベースシステムから取得することも原理的には可能であるが，オーバーヘッドが大きくなってしまう．このため，集約された $\{lb_j\}$ の情報はオンコアで保持し，高速な構造劣化度分布の把握を可能にしておき，同時に，$\{p_i\}$ の情報はセカンダリストレージに保持しておき，必要に応じて $\{lb_j\}$ の解像度を部分的に上げる運用が考えられる．

上記で述べた差分計算手法によってデータベースシステム稼働中に最新のデータベース構造を把握するための余分な IO がほとんど発生しないため，構造劣化度分布のオンライン監視が可能になる．

3. データベース再編成支援

本節では，前節で述べた構造劣化度分布を分析することにより，与えられた再編成ポリシに従って再編成対象領域を同定する手法について述べる．

3.1 再編成ポリシ

データベース再編成は，構造劣化が著しく進行することでデータベースのアクセス性能が性能要求を満たさなくなることを防ぐために実施するものである．このため，再編成ポリシは許容できる性能閾値などを与えることが想定される．その他，時間と資源 (CPU パワー，IO リソース) を与えて可能な限りの再編成を行い，次回の再編成出来るだけ延ばす，などのポリシも考えられる．

オンライン構造劣化モニタは，構造劣化に起因する性能低下量の推定値を，構造劣化度として表現しているため，本稿では，必要な性能を満たすための構造劣化度閾値をユーザもしくは管理者が指定することにより，再編成対象領域を同定させる最も基本的なアプローチを取る．

例えば，B+木の範囲検索を対象とした場合，構造劣化のない状態のアクセスコストのおおむね 3 倍以内に抑えたいときには構造劣化度分布閾値を 3 に設定することで，3 倍を越える部分を同定することが出来，当該領域を再編成することで，再編成にかかる時間および投入する資源を最小限にしながら，性能要求を満たすことが可能になる．

統合的な運用のために，本来ならば，再編成を行うのに必要な時間および資源を考慮に入れて，再編成を早めに実施する必要があるが，それは再編成スケジュールの問題であるため，議論は他稿に譲り，本稿では閾値を越える構造劣化部分の同定に注力する．

3.2 再編成対象領域の同定

構造劣化度分布は論理空間に対して必ずしも連続でないため，単純に構造劣化度分布のなかで，与えられた閾値を越える lb_j もしくはページ p_i を同定すると，各連続領域の大きさが非常に小さい集合が得られることが予測される．

しかし，データの物理順序を再配置により改善する再編成は，ある粒度以上のまとまった連続論理領域で実施しないと効果がない．例えば B+木の範囲検索を対象とする構造劣化に対して，論理空間で連続する数ページを部分再編成 [7] したとき，その最初のページは，直前のページと物理的に不連続であるため，構造劣化が改善されない．このため，これら境界のページにおける構造劣化が無視できるようにある程度まとまった論理連続

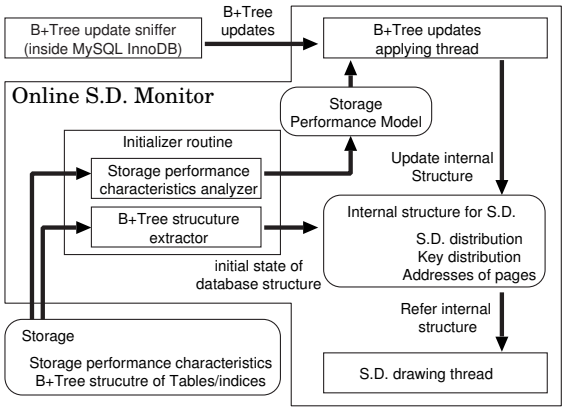

図 2　オンライン構造劣化モニタ試作機の構成

ページを対象に再編成を実施する必要がある．

今日のディスクドライブにおいては，一般にランダムアクセスコストがシーケンシャルアクセスコストの数十倍であるため，境界の影響を無視できるようにするためには，最低でも 100〜1000 ページ程度の連続領域を対象にする必要がある．

そのため，再編成対象アドバイザは，再編成により十分効果が期待できる論理空間サイズを指定して，構造劣化度分布をローパスフィルタにかけることで，指定したサイズ以上の連続構造劣化領域のうち，平均的に構造劣化度が閾値を越える領域を同定する．

ローパスフィルタに必要な計算量は移動平均を計算するコストであるため，論理空間の大きさ N に対して $O(N)$ で計算することが出来る．

大容量データベースに対しては，オンコアに集約された構造劣化度分布を持ち，それに対してローパスフィルタをかけることで，同定領域は多少荒くなるが，そもそも再編成に必要な最低粒度が大きいため，十分詳細な再編成対象領域を同定することができる．

ローパスフィルタを用いると，指定されたサイズより小さい連続領域は，実際に構造劣化していても同定漏れが起こる．ただし，このような細粒度の構造劣化領域のみを再編成しても，上述の通り十分な再編成効果が見込めない．これらの細粒度構造劣化を改善するためには，それらの領域を含み，指定されたサイズ以上の再編成領域を確保する必要があるため，再編成する領域に比べて再編成によって回復する性能が小さい．このため，より大きな範囲に渡って構造劣化している領域に比べて再編成優先順位は低い．よって本モニタは，これらの細粒度構造劣化領域を再編成対象領域としてユーザーに示唆しない．

4. 試作機によるケーススタディ

先述の再編成対象領域同定器を，オンラインデータベース構造劣化モニタ [11] に実装し，ケーススタディを用いてその有用性を検証した．

4.1 試作機の構成

図 2 に，オンライン構造劣化モニタ試作機の構成を示した．描画および分析を高速化するために，logical bulk を用いた集約情報をオンコアで持ち，完全なデータはセカンダリストレージ上に保持している．再編成対象領域同定器は，オンコアの構造劣化度分布を用いて，再編成対象領域を同定し，可視化して構造劣化度分布描画に重ねて描画する．

4.2 環境と設定

実験環境として，データベースサーバとオンライン構造劣化モニタサーバを用意した．データベースサーバは Linux 2.4 が動作する PC で，ファイバチャネル経由のディスクドライブ 4 台を SoftwareRAID0 化し raw デバイスとして使用した．ディスクドライブは，Cheetah 10K 18GB [4] を用い，チャンクサイズは 64KB とした．オンライン構造劣化モニタサーバには，Linux 2.6 が動作する PC を用い，データベースサーバと 1Gbps のイーサネットで接続した．

まず，データベースサーバにて，MySQL 5.0 InnoDB データベースエンジンを用いて 1GB の表空間領域を確保した．ページサイズは標準の 16KB とした．

上記の表空間に人工的なクラスタ表 T1 を作成した．InnoDB においては，クラスタ表は B+木を用いて構成される．スキーマは `T1(int id primary key, int id2, char str[255])` とし，主鍵 `id` を，$(0 \leqq$ `id` $< 1000000)$ として，重複がないように 100 万行ロードした．1 ページあたり，約 50 行格納できる．データロード後は 表空間内で 300MB 程度の大きさを占めた．

ワークロードは，主鍵に対して連続する 1000 レコードの範囲をバルクとしてランダムに選択し，削除し，一定時間後に同じレコードを挿入する操作を一単位とし，主鍵分布の中で $200,000 \leqq$ `id` $\leqq 400,000$ すなわち 20〜40%の範囲に，4 並列で計 400 回，$600,000 \leqq$ `id` $\leqq 800,000$ すなわち 60〜80%の領域に 2 並列で計 200 回投入した．

ワークロードを投入中にオンライン構造劣化モニタを動作させ，ワークロード投入後に，閾値を許容構造劣化度 2 として設定し，また，再編成対象とする最小連続サイズを 16MB(1024 ページ) とし，再編成対象領域同定器を動作させた．その後，再編成対象領域同定器の示唆に従い，再編成をアンロード，リロードする形で行った．

4.3 結　　果

まず，データロード直後の構造劣化度分布を図 3 に示す．上図が主鍵順の葉ページ番号を用いた論理軸における，下図がページアドレスを用いた物理軸における構造劣化度分布である．主鍵の範囲を区切って色付け表示した．スケールは 1/32 であり，構造劣化モニタの表示 1 ドットがデータベース 32 ページ分の情報を表示している．スケールはモニタからリアルタイムに変更可能である．初期状態では論理順序と物理順序が一致していることを確認できる．

ワークロード投入後，再編成アドバイザを動作させた結果を図 4 に示す．構造劣化度 2 のところにバーがあるが，これをドラッグアンドドロップすることにより，閾値となる構造劣化度を変更して分析することが出来る．分析の結果，構造劣化した領域の背景が灰色として表示され，当該領域の構造劣化度の平均値がバーで示される．このことから，左側の構造劣化領域の構造劣化度が約 4，右側の構造劣化領域の構造劣化度が約 2.5 であることが確認できる．物理軸においては，ワークロードによって更新された領域が，新たに確保された領域も占有してい

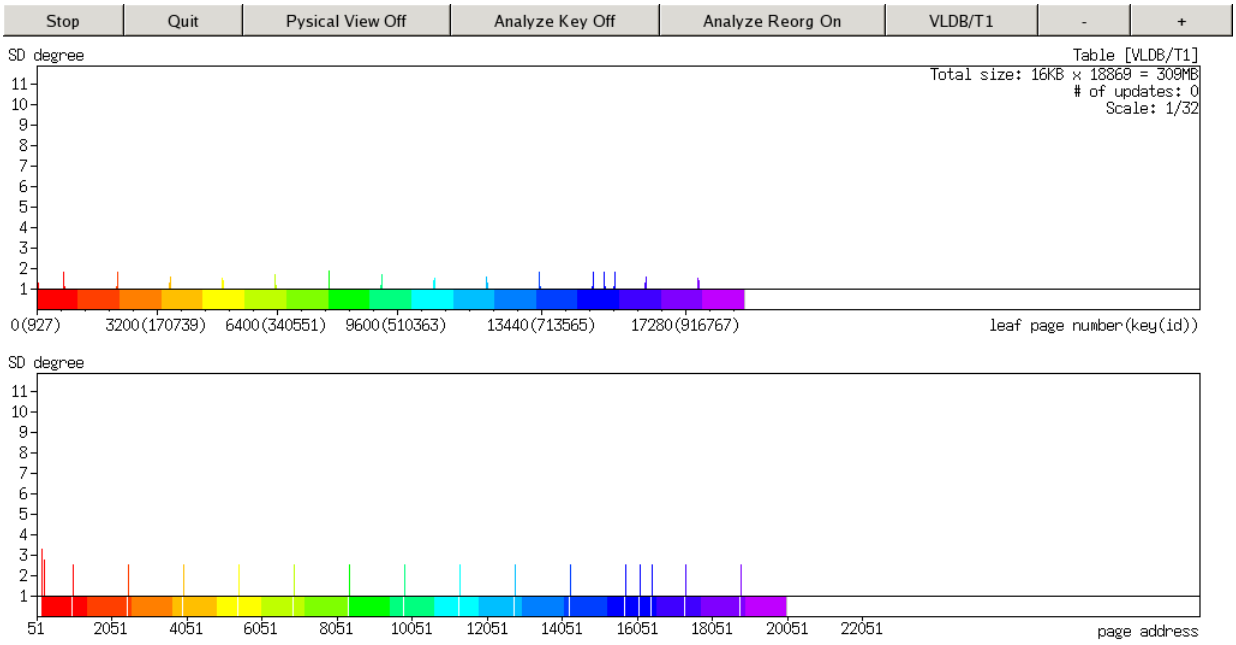

図 3　データロード後の構造劣化度分布

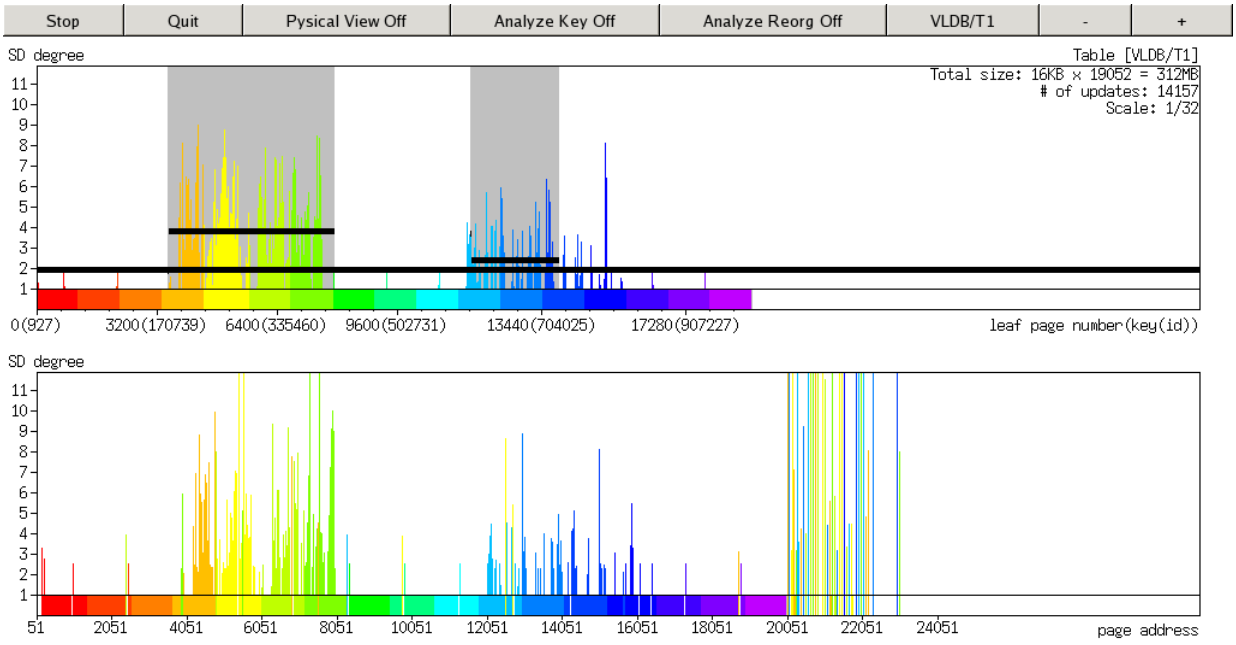

図 4　ワークロード投入後の構造劣化度分布
再編成対象領域示唆

ることが確認できる．

アドバイザの示唆通りに再編成を実施した結果を図 5 に示す．構造劣化がほぼなくなっていることが確認できる．一部再編成されずに構造劣化が閾値を越えている領域が存在するが，いずれもサイズの小さい連続領域であり，まとまった範囲検索における影響は再編成を実施した領域に比べて非常に小さいものと推察される．

以上の結果から，本稿で構築した再編成支援ツールは，実用的に再編成対象領域をアドバイスできることが分かった．

5. 関 連 研 究

データベース再編成に関する研究には，実行方式の高度化及び実行スケジューリングの最適化を中心に行われてきた．前者については，近年オンライン再編成方式についての研究が行われてきている．[2], [8]．後者については，再編成スケジュールの最適化に関する研究が行われてきた [1], [6]．再編成契機判断の自立化は，後者に属する．これらの研究では，表空間を構成するストレージの性能特性は考慮せず，比較的簡単なデータベー

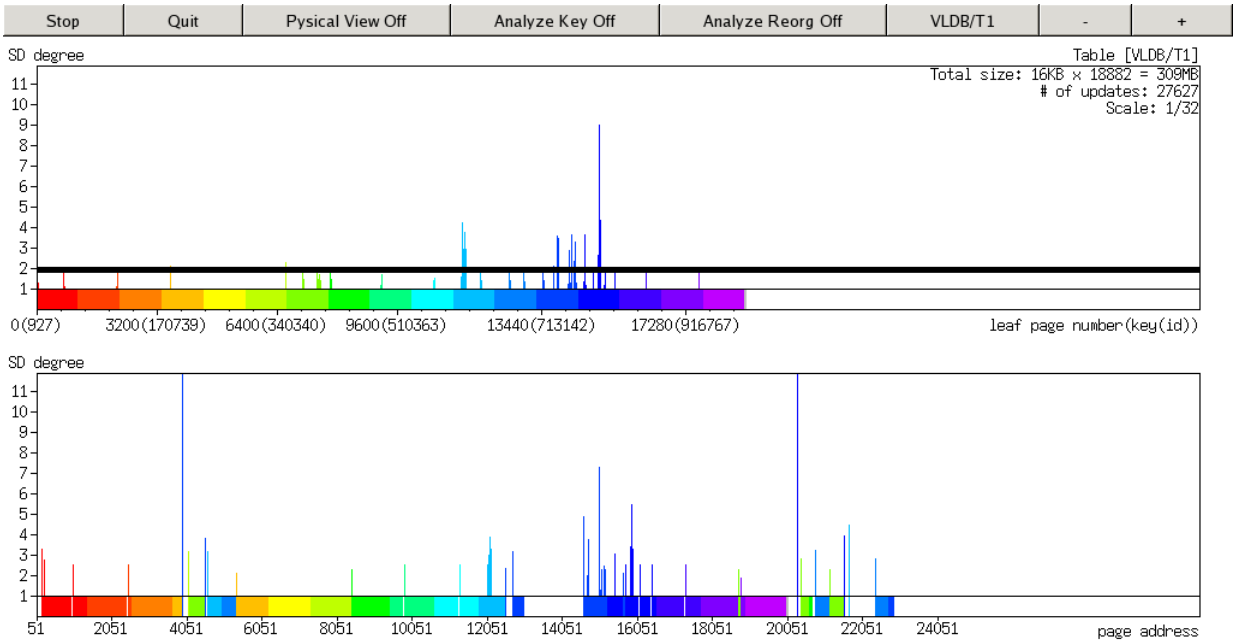

図 5　再編成実施後の構造劣化度分布

スモデルに基づいて，近未来の性能低下量を予測し，運用時間あたりの再編成コストを最小化する再編成スケジューリング手法が考察された．

IO 性能モデルに関して，文献 [5] は，データベースバッファ等の影響を考慮して，B+木索引と表を走査するのに必要な IO 数をコストとしてモデル化している．この論文では，データがクラスタ化されていない状態を仮定しており，クラスタ化の度合いを構造劣化度として扱う本研究とは扱う対象が異なる．

文献 [7] において，部分再編成手法が提案された．それまで再編成対象となる最小粒度は，表単位であるか，もしくは表データが配置されている表空間が分割されている場合は分割された表空間であった．この論文では，大容量の表に対して，表空間が分割されているかどうかに関係なく，構造劣化部分のみを再編成することにより，再編成時間を削減しつつ，構造劣化がほぼ回復する．本研究では，構造劣化監視においてリアルタイムに構造劣化部分の同定を行うことが可能であるため，部分再編成対象の同定に貢献する．

6. おわりに

本稿は，オンライン構造劣化モニタを用いた再編成支援ツールを構築し，ケーススタディを用いてその有用性を確認した．今後の課題として，再編成ポリシの充実や，オンライン再編成スケジューラの構築が挙げられる．

謝　　辞

本研究の一部は，文部科学省リーディングプロジェクト e-society 基盤ソフトウェアの総合開発「先進的なストレージ技術」の助成により行われた．協力企業である株式会社日立製作所より多くの有益なコメントを頂戴した．深謝する次第である．

文　　献

[1] Don S. Batory. Optimal file designs and reorganization points. *ACM Trans. Database Syst.*, 7(1):60–81, 1982.

[2] (Ed.)D.Lomet. Special Issue on Online Reorganization. *IEEE Data Eng. Bull.*, 19(2):1, 1996.

[3] Sam Lightstone, Berni Schiefer, Danny Zilio, and Jim Kleewein. Autonomic Computing for Relational Databases: The Ten Year Vision. In *Proceedings of Workshop on Autonomic Computing Principles and Architectures (AUCOPA2003)*, August 2003.

[4] Seagate Technology LLC. *Cheetah 18LP FC Disk Drive Product Manual Volume 1*, 1999.

[5] Lothar F. Mackert and Guy M. Lohman. Index Scans Using a Finite LRU Buffer: A Validated I/O Model. *ACM Trans. Database Syst.*, 14(3):401–424, 1989.

[6] Ben Shneiderman. Optimum data base reorganization points. *Commun. ACM*, 16(6):362–365, 1973.

[7] 合田 和生，喜連川 優．構造劣化の局所性を活かしたデータベース部分再編成の提案．In **電子情報通信学会 第 *17* 回データ工学ワークショップ** *(DEWS2006)*, 2006.

[8] 合田和生，喜連川優．データベース再編成機構を有するストレージシステム．**情報処理学会論文誌データベース**, 46(SIG 8(TOD 26)):pp.130–147, 2005.

[9] 星野 喬，合田 和生，喜連川 優．関係データベース再編成契機決定のための性能劣化同定方式．In **電子情報通信学会 第 *16* 回データ工学ワークショップ** *(DEWS2005)*, 2005.

[10] 星野喬，合田和生，喜連川優．データベース更新差分を用いた範囲検索の IO コスト推定．**日本データベース学会論文誌** *(DBSJ Letters)*, 4(2):37–40, 2005.

[11] 星野喬，合田和生，喜連川優．データベースにおけるリアルタイム構造劣化監視機構の試作．**日本データベース学会論文誌** *(DBSJ Letters)*, 5(2):37–40, 2006.