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

Database Partial Reorganization for Exploiting Spatial Locality of Structural Deterioration

合田 和生♠　喜連川 優♠

Kazuo GODA　Masaru KITSUREGAWA

データベースにおいては一般に更新操作が繰り返されるにつれて，例えば，レコードの並びが乱雑化し，連続したキー値を有するレコードの走査に要するコストが大きく増加するなど，格納構造が劣化することによりアクセス性能が低下する現象，即ち構造劣化が発生する．多くのデータベースにおける更新は局所性を有していることから，構造劣化は記憶空間のうち一部の限られた領域に発生する場合が多い．本論文では上記の特性に着目し，構造が劣化した局所空間に対して再編成を行うことにより，再編成実行時間を大きく削減し，空間全体を再編成する場合とほぼ同程度の構造回復効果を実現する部分再編成手法を提案し，構造劣化が局所性を有する２つの代表的なケーススタディを示す．また，試作機において実装した部分再編成機構の性能評価を示すことにより，その有効性を明らかにする．

As updates are repeatedly performed in database, the data structure may gradually deteriorate and then the data access performance may also degrade. Specifically speaking, a number of record manipulations could lead to scrambled placement, resulting in much larger cost of record scanning. Such a phenomenon is called structural deterioration. As database has usually access locality, its data structure may also deteriorate in limited parts of the storage space. Exploiting this characteristic, the paper proposes a new reorganization method, partial reorganization. The method is able to reorganize only locally structurally deteriorated space in the database. Partial reorganization need not reorganize the entire database, but can remove most of the structural deterioration. Thus, the reorganization time can be significantly reduced, while the structural efficiency can be recovered similarly to the conventional method. This paper presents two typical case studies and presents experimental evaluations with an implemented prototype, which confirm the effectiveness of the proposal.

1. はじめに

記憶装置上のデータは，更新によってデータ構造の編成が乱れ，データアクセスの性能が著しく低下する恐れがあり，当該現象を構造劣化(structural deterioration)と呼ぶ[1]．とりわけ，二次記憶装置のアクセス性能はディスクドライブの機械的特性に依存するため，構造劣化による性能低下は，膨大なデータを二次記憶装置上に格納するデータベースシステムにとって，最も深刻な問題の一つである．

当該問題を解決すべく，従来より，記憶装置上のデータを再配置することにより，劣化した構造を回復し性能を改善するデータベース再編成(database reorganization)[2]をデータベースシステムの一機能として実現する試みが行われて来ており，データベース再編成は今日では，データベースシステムに不可欠な機能となっている．一般に，再編成はデータベース空間中の全データの再配置を行うため，膨大なIOを発行する必要があり，その負荷は極めて大きく，処理は長時間に渡る．このため，データベース再編成の高度化，並びにデータベース再編成の管理に関して，多くの様々な研究がこれまで行われて来た．

一方で，一般にデータベースの更新はデータベース空間全体に対して一様に行われることはなく，局所性を有していることが広く知られている．即ち，データベース空間の一部の限られた領域に多くの更新操作が集中するアクセス特性がしばしば見られる．このため，データベースの更新操作によって発生する構造劣化に関しても，同様に，データベース空間の一部の限られた領域において発生する場合が多い．本論文では，上記のような構造劣化の局所性に着目し，構造が劣化した局所空間に対して再編成を行うことにより，再編成実行時間を大きく削減し，空間全体を再編成する場合とほぼ同程度の構造回復効果を実現する部分再編成(partial reorganization)手法を提案する．更に，本論文では，構造劣化が局所性を有する２つの代表的なケーススタディを用いて，構造劣化空間の同定方式，並びに当該同定空間の再編成方式に関して論じる．また，自己再編成ストレージ[1]の試作機において実装した部分再編成機構の性能評価結果を示すことにより，その有効性を明らかにする．提案手法はとりわけ大規模なデータベースに対して極めて有効な方式といえる．著者の知る限り，これまでに，構造劣化の局所性に着目し，データベースの部分再編成方式を明らかにし，その有効性を示す提案はなされていない．

本論文の構成は以下の通りである．2.においては，関連研究をまとめる．3.においては，本論文の提案する部分再編成に関して一般的な説明を行い，4.においては，局所的な構造劣化を２例示し，これをケーススタディとして，より具体的な部分再編成の検討を行う．5.においては，自己再編成ストレージにおける部分再編成の設計，及び実装を示し，これを用いたケーススタディの検証を行う．6.においては，本論文をまとめるとともに，今後の課題を示す．

2. 関連研究

データベース再編成の高度化を目指す研究は，サービス継続中に再編成を実行可能とするオンライン再編成[3]に関する研究を中心として行われて来た．これらの研究は，トランザクション処理とデータベース再編成処理を並行，若しくは並列に動作させ，トランザクション処理性能への副作用を最小限としながら再編成時間を短縮させる目的を以って，主に，同時実行制御方式，並びにIO最適化方式に関する議論を中心として行われて来た．対して，本論文で提案する部分再編成は，データベースの構造劣化の局所性に着目し，構造劣化空間を同定し，当該空間のみの再編成を可能とする再編成方式を導入することにより，再編成時間の短縮を目的としており，

♠ 正会員　東京大学生産技術研究所
{kgoda, kitsure}@tkl.iis.u-tokyo.ac.jp

着眼点は大きく異なる．

構造劣化の定量的な把握方式を明らかにする試みに関しては，これまで，データベース再編成の最適化，若しくは自律化を目指す研究の中心的課題として行われて来た[4-8]．これらの研究は，本研究と同様に構造劣化の定量的な把握に基づく再編成の高度な管理を目指しているものの，再編成を時間軸上で制御しようとしている．対して，本研究の提案する部分再編成は，構造劣化の空間的な局所性を活用し，再編成を空間軸上で制御することを目指しており，目的を大きく異にする．

既に幾つかのデータベースベンダは，そのデータベースシステム製品において，類似の名称を有する部分再編成なる機能を実装しているが[9,10]，これらの機能が予め，データベース空間を複数の区画に分割して，再編成を区画単位で実施するものであるのに対し，本論文は，データベース空間内で構造が劣化した任意の部分空間を同定し，当該空間の再編成を可能とする手法を提案していることから，その学術的意義は大きい．

3. データベース部分再編成

部分再編成(partial reorganization)は，データベースにおいて，構造が劣化した局所空間に対して再編成を行うことにより，再編成実行時間を大きく削減し，空間全体を再編成する従来的な手法と比較してほぼ同程度の構造回復効果を実現することを目指す．

一般に，従来的な再編成においては，データベース空間全体を再編成する，若しくはユーザが指定したオブジェクトや空間を再編成する．対して，本論文の提案する部分再編成は，データベース空間のうち構造が劣化した部分空間，即ち構造劣化空間を再編成することから，その処理は，構造劣化空間の同定，及び当該空間の再編成の二手順によって実施される．即ち，第一の手順によって，部分再編成の対象空間とするべき構造劣化空間を同定し，第二手順によって，当該同定空間の再編成を実施し，構造劣化の解消を行う．本論文では，紙面の関係から上記2つの手順の一般的な手法に関して焦点を当てた議論は[11]に譲り，次章以降においてケーススタディを用いて議論を進める．

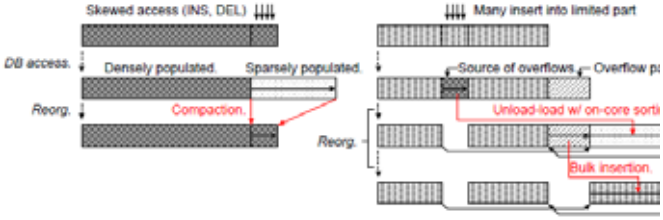
4. ケーススタディによる部分再編成の検討

本章では，構造劣化が局所的に発生する典型的な例を２つ示し，ケーススタディとして，部分再編成における構造劣化空間の同定方式，並びに当該空間の再編成方式について，より具体的な議論を行う．

4.1 ケーススタディ１：局所的疎空間の縮退

通常，非クラスタ化表，即ち，レコード格納順序を意識しない関係表においてレコードが削除されると，格納構造においては当該レコードに削除フラグが立てられるものの，その空間は即座には回収されない．上記の実装を有する多くのデータベースシステムにおいては，図１(a)に示すように，非クラスタ化表に対してレコードの挿入と削除が繰り返される場合，表の一部の空間の充填率が低下し，著しくデータの格納効率の悪い疎な空間が発生し，全表検索の性能を低下させる．このようなアクセス特性は，一般にWeb上の電子商取引のバスケットシステムなどに頻繁に見られ，検索等に係る走査の効率が低下することから，全表検索などを伴う問い合わせ処理の性能が低下する恐れがある．

このような局所的に疎な空間が発生する構造劣化を解消



(a) Case study #1.　(b) Case study #2.
図１　ケーススタディの概要.
Fig.1 Overview of case studies.

するためには，当該疎空間中の空き空間を回収し，縮退させることにより，空間配置の効率性を高める必要がある．従来的には，非クラスタ化表の再編成は，データベース中の表から全レコードをアンロードし，再度レコードをロードすることにより，上記の縮退を実施する．しかし，一般的に疎空間は局所的に発生することから，上記のアンロード・ロード戦略による再編成は必ずしも効率的でない．むしろ，局所的な疎空間を把握し，当該空間のみをアンロード・ロードすることにより，全体をアンロード・ロードする場合と比較して，大幅に再編成の実行時間を削減するとともに，同程度の構造劣化の解消効果を達成することが可能である．

当該ケーススタディにおける構造劣化空間の同定は，構造劣化度として，データベース空間を構成する基本空間単位(ページ，若しくはエクステント)毎の空き率を計測することにより，可能となる．これにより，例えば，データベース空間中の各ページに対して空き率を計測することにより，空き率閾値を下回った部分空間に関して，これを構造劣化空間と同定し，部分再編成の対象とすることが可能となる．

同定された構造劣化空間に関しては，当該空間のアンロード・ロードにより，空間の縮退を行う．即ち，まず，構造劣化空間を走査して有効なレコードをアンロードし，その後，アンロードされたレコードを同じ空間にロードする．この際，レコードは詰めて配置される．一般に，同じ空間へのアンロードとロードは並行して実行することができないが，この場合，空間を縮退させる操作であることが予め判明していることから，再編成を更に高速化されるためのテクニックとして，アンロード操作とロード操作の間に適切な処理バッファを置くことにより，アンロードとロードをパイプライン的に実行することが可能となり，これにより，高速に空間の縮退を行うことが期待される．

4.2 ケーススタディ２：オーバーフローレコードの解消

多くのデータベースシステムはクラスタ化表，即ち，指定されたクラスタ鍵によって格納順序を決定する関係表を有しているが，その実装は，データベースシステムにより多岐に渡る．ここでは，図１(b)に示すような，クラスタ索引ベースのクラスタ化表に焦点を当てて述べる．クラスタ化表へのレコード挿入については，クラスタ鍵によりクラスタ索引を辿り，挿入先ページが決定され，当該ページへレコードが格納される．この際，当該ページが満杯で挿入が失敗する場合，近傍ページへレコードの格納が試みられるが，それも失敗した場合，データベース空間内に新たなページが確保され，当該ページにレコードが格納される．この際，新たに割り当てられるページは，本来格納されるべきページから物理的に離れている可能性がある．本論文では，上記において，新規に割り当てられたページをオーバーフローページ，当該ペー

ジへ格納されたレコードをオーバーフローレコード，オーバーフローレコードが本来格納されるべきであったページをオーバーフロー元ページと呼ぶ．

クラスタ鍵による範囲検索を行う場合，鍵順にレコードを走査する必要があり，一般にクラスタ化表では当該走査がデータベースの物理空間のシーケンシャルアクセスとなることが期待されるが，オーバーフローレコードに関しては，本来格納されるべきページから離れたページに格納されることから，走査がランダムアクセスとなり，範囲検索の性能が低下する恐れがある．特に，一部の限られたクラスタ鍵範囲に挿入が繰り返される場合，上記のオーバーフローレコードが多発し，著しい性能低下を招く．このようなアクセス特性は，日付や時間をクラスタ鍵として利用する財務会計アプリケーション等に頻繁に見られる．

オーバーフローレコードを解消するには，再編成が必要であるが，大きなクラスタ化表を再編成する場合，多数のランを伴う外部ソートが必須となり，再編成の時間が長くなる．このような場合，オーバーフローレコード以外の空間は殆んど構造は劣化していないため，部分再編成として，オーバーフローレコードの解消のみを行うことにより，効率的な再編成を実施することができる．

当該ケーススタディにおける構造劣化空間の同定は，クラスタ鍵に対するレコード格納順序不正率の変動と，データベース空間の物理アドレスに対するクラスタ鍵の分布を計測することにより，可能となる．前者は，クラスタ化索引の葉ページから計測され，当該葉ページに対応するクラスタ鍵範囲のレコードの格納順序の不正率を意味する．一方，後者は，クラスタ化表のページから，当該ページに格納されているレコード群の鍵の平均値，並びに標準偏差を計測し，これに基づき，データベース空間におけるページに対するレコードの鍵分布を求める．

レコード格納順序不正率の変動と，クラスタ鍵の分布を計測することにより，以下の手順により，構造劣化空間を同定することが可能となる．まず，レコード格納順序不正率の高いクラスタ鍵分布範囲を同定し，その後，クラスタ鍵分布統計により，当該クラスタ鍵分布範囲に対する物理アドレス空間を同定する．このとき，同定された物理アドレス空間におけるクラスタ鍵範囲が広い空間をオーバーフローページ，それ以外をオーバーフローレコードの発生源ページ(オーバーフロー元ページ)とする．

同定された構造劣化空間，即ち，オーバーフローページとオーバーフロー元ページに対して，オーバーフローレコードの解消を行う部分再編成は，以下の２ステップの手順で実施する．

（１）まず，オーバーフロー元ページをアンロードし，レコードをクラスタ鍵順に整列し，新たな空間にロードする．オーバーフロー元ページでは，レコードは殆んど整列化されている，若しくは本来格納されるべきページの近傍に格納されていることが期待されるため，一定のウィンドウ内で隣接するページ群内で整列操作を行うことにより，レコードを確実に整列することができる．即ち，多くの場合において，整列は外部整列ではなくオンコアで行うことが可能である．

（２）次に，オーバーフローページをアンロードし，レコードをクラスタ鍵順に整列し，先にロードされた空間にバルク挿入を実施する．

以上の部分再編成により，クラスタ化表全体の外部整列を避け，必要な整列のみを実施することにより，オーバーフローレコードを解消することが可能となる．即ち，再編成時間の大幅な高速化が期待できる．

5. 自己再編成ストレージ試作機を用いた部分再編成の有効性検証

本章では，オンラインデータベース再編成機能を有する高機能ディスクアレイである自己再編成ストレージ試作機[1]を用いた部分再編成の実装，並びにそれを用いた有効性の検証に関して述べる．

著者らは前章で示したケーススタディの部分再編成を，自己再編成ストレージの試作機において実装した．この際，データベースシステムとしてHiRDB[12]を対象とした．なお，紙面の制約から，本論文では部分再編成の有効性を示すことに焦点をあて，実験の主な結果を報告する。試作機実験システムの詳細に関しては文献[1]を，部分再編成の実装に関しては文献[11]を参照されたい。

5.1 ケーススタディ１の検証

代表的なデータベースベンチマークであるTPC-Hのデータセットを利用して，ケーススタディ１における構造劣化空間の同定，並びに構造劣化空間の再編成の検証を行った．

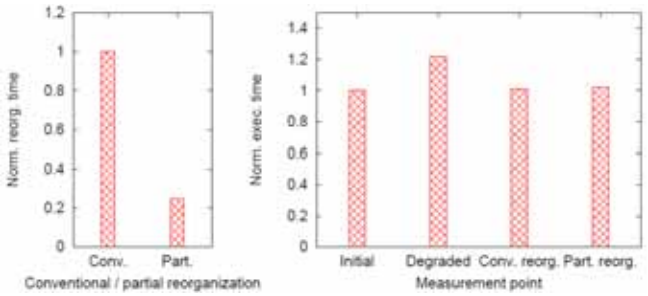
局所的な疎空間の生成を模擬するために，表空間にTPC-Hにおけるスケールファクタ(SF)を1としたORDERS表(150万行)を初期データとして非クラスタ化表に格納し，当該表に対して，150万レコードの挿入及び削除を実施した．この際，挿入及び削除は当該表の主鍵であるO_IDについて，[95%, 5%]の局所性を有することとした．即ち，レコードの挿入及び削除のうち，95%はO_IDのうち最大値から5%の範囲内の鍵を対象に行われることを意味する．その後，ケーススタディ１において検証した部分再編成を実行した．

図２には，上記において部分再編成を実施した場合の実行時間，並びにその構造劣化の解消効果を，従来的にデータベース全体を再編成する場合と比較して，まとめる．図２(a)に表空間全体の再編成と部分再編成の実行時間を比較する．図２(b)には，初期状態，再編成前状態，並びに従来的な再編成，部分再編成それぞれに関して，再編成後状態の全表検索の実行時間を示す．データベースの更新により疎空間が発生し，全表検索の実行時間が2割超悪化したが，部分再編成によって十分に実行時間を回復することができたことが分かる．また，部分再編成により，従来的な再編成と比較して77%の再編成実行時間の削減が可能となったことが分かる．

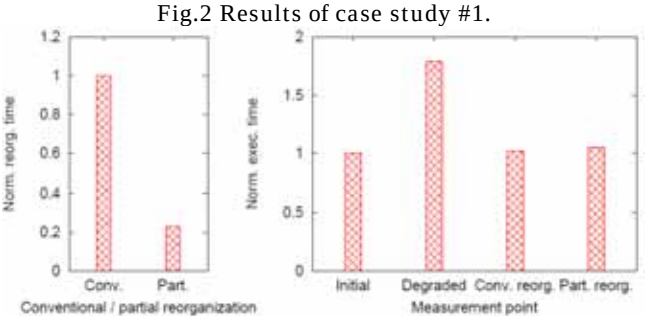
5.2 ケーススタディ２の検証

前節と同様に，ケーススタディ２における構造劣化空間の同定，並びに構造劣化空間の再編成の検証を行った．局所的なクラスタ鍵空間への挿入を模擬するために，表空間にTPC-Hにおけるスケールファクタ(SF)を1としたORDERS表(150万行)を初期データとして充填率70%でクラスタ化表に格納し，さらに，当該表に対して150万行の挿入を実施した．挿入は当該表の主鍵であるO_IDについて，[95%, 5%]の局所性を以って行った．即ち，挿入のうち，95%はO_ID}のうち最大値から5%の鍵を対象に行われる．その後，ケーススタディ２において検証した部分再編成を実行した．

図３には，上記において部分再編成を実施した場合の実行時間，並びにその構造劣化の解消効果を，従来的にデータベース全体を再編成する場合と比較して，まとめる．図３(a)に表空間全体の再編成と部分再編成の実行時間を比較する．図３(b)には，初期状態，再編成前状態，並びに従来的な再編成，部分再編成それぞれに関して，再編成後状態の範囲検索の実行時間を示す．この際，実行時間はデータ量により正



(a) Reorganization time. (b) Query execution time.
図２ ケーススタディ１の結果.
Fig.2 Results of case study #1.



(a) Reorganization time. (b) Query execution time.
図３ ケーススタディ２の結果.
Fig.3 Results of case study #2.

規化されている．データベースの更新によりオーバーフローレコードが発生し，範囲検索の実行時間が約8割悪化したが，部分再編成によって十分に実行時間を回復することができたことが分かる．また，部分再編成により，従来的な再編成と比較して75%の再編成実行時間の削減が可能となったことが分かる．

以上，2つのケーススタディの検証により，部分再編成により，従来的な再編成と比較して大幅な再編成実行時間の削減が可能となる一方，ほぼ同等の構造劣化の解消効果が達成可能であることが分かり，その有効性が確認された．

6. まとめ

本論文では，データベースの更新が局所性を有することから，その構造劣化も局所的に発生することに着目し，データベース空間の一部のみを再編成することにより，再編成実行時間を大きく削減し，空間全体を再編成する従来的な手法とほぼ同程度の構造回復効果を目指す部分再編成を提案した．部分再編成を実施するための，構造劣化空間の同定並びに再編成の方式に関して，典型的なデータベースの更新例を２つ示し，ケーススタディとして検討した．また，自己再編成ストレージ試作機において，当該２例の部分再編成を実装し，その有効性を検証した．ケーススタディにおいては，部分再編成により再編成に係る時間を75-77%削減可能であり，かつ，同等の再編成効果を得られることを示した．

本論文は部分再編成の研究に関する第一ステップであると考えている．今後は，構造劣化空間の同定に関してストレージにおける構造劣化度の計測コストの検証を行う他，時間軸，空間軸双方における再編成管理の高度化に関して研究を進めたい．

[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．感謝する次第である．

[文献]

[1] 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，Vol. 46, No. SIG 8(TOD 26), pp. 130–147, 2005.
[2] Gary H. Sockut and Robert P. Goldberg. Database Reorganization - Principles and Practice. ACM Comput. Surv., Vol. 11, No. 4, pp. 371–395, 1979.
[3] David Lomet, editor. IEEE Data Eng. Bull.: Special Issue on Online Reorganization., Vol. 19. IEEE Computer Society, 1996.
[4] B. Shneiderman. Optimum Data Base Reorganization Points. Comm. ACM, Vol. 16, No. 6, pp. 362–365, 1973.
[5] S. B. Yao, K. S. Das, and T. J. Teorey. A Dynamic Database Reorganization Algorithm. ACM Trans. Database Syst., Vol. 1, No. 2, pp. 159–174, 1976.
[6] K. Maruyama and S. E. Smith. Optimal Reorganization of Distributed Space Disk Files. Comm. ACM, Vol. 19, No. 11, pp. 634–642, 1976.
[7] D. S. Batory. Optimal file designs and reorganization points. ACM Trans. Database Syst., Vol. 7, No. 1, pp. 60–81, 1982.
[8] 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 Letters, Vol. 4, No. 2, pp. 37–40, 2005.
[9] BMC Software. REORG PLUS for DB2 General Information, 1997. ARUPMG093097.
[10] IBM Corp. Information Management System, Utilities Reference: Database and Transaction Manager, Version 8., 2004. SC27-1308-02.
[11] 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．電子情報通信学会第 17 回データ工学ワークショップ／第４回日本データベース学会年次大会(DEWS2006)，4C-o4，2006.
[12] 日立製作所. Hitachi HiRDB Version 7. http://www.hitachi.co.jp/Prod/comp/soft1/hirdb/.

合田 和生 Kazuo GODA
東京大学生産技術研究所特任助手．2000 東京大学工学部卒業，2005 同大学院情報理工学系研究科博士課程単位取得満期退学．博士（情報理工学）．並列データベースシステム，ストレージシステムの研究に従事．本会，情報処理学会，ACM，USENIX 会員.

喜連川 優 Masaru KITSUREGAWA
1978 東京大学工学部電子工学科卒業．1983 同大大学院工学系研究科情報工学博士課程修了．工学博士．同年同大生産技術研究所講師．現在，同教授．2003 より同所戦略情報融合国際研究センター長．データベース工学，並列処理，Web マイニングに関する研究に従事．現在，本会理事，情報処理学会，電子情報通信学会各フェロー．ACM SIGMOD Japan Chapter Chair，電子情報通信学会データ工学研究専門委員会委員長歴任．VLDB Trustee (1997-2002), IEEE ICDE, PAKDD, WAIM などステアリング委員．IEEE データ工学国際会議 Program Co-chair(99), General Co-chair(05).