

# データベース構造劣化による OLTP 性能低下に関する一考察

## A Study on OLTP Performance Degradation by Structural Deterioration of Database

星野 喬<sup>\*</sup> 合田 和生<sup>\*</sup> 喜連川 優<sup>\*</sup>

Takashi HOSHINO Kazuo GODA  
Masaru KITSUREGAWA

データベース更新が繰り返されると、二次記憶装置内の物理データ格納構造が劣化する。このような構造劣化は性能低下を引き起こす。オンライントランザクション処理(OLTP)は、電子金融取引や電子商取引などに不可欠なデータ処理方式である。従来のデータベースシステムにおける OLTP 高速化技術は、構造劣化による性能低下を防ぐという観点からはあまり行われてこなかった。近年、業務やサービスの電子化に伴い、データベース常時運用ニーズも高まってきたため、今後、構造劣化を監視し、性能低下を自動的に予防する技術の必要性は、ますます増大すると考えられる。本論分では構造劣化が OLTP 性能に与える影響をベンチマークを用いて解析し、負荷に適応的な、構造劣化の進行を抑制するデータ更新戦略について考察する。

Database updates disorganize data stored physically in secondary storage, which is called structural deterioration and causes performance degradation. Online Transaction Processing (OLTP) is an essential data processing scheme for e-finance, e-commerce and so on. Conventional researches to improve OLTP performance lack a view from the aspect of preventing performance degradation due to structural deterioration. Recently, more and more business operations and services are being digitized then 24-hours-a-day operation of database is required. Needs for techniques to monitor structural deterioration and prevent performance degradation are increasing more and more. In this paper, we analyze how structural deterioration effects performance degradation in OLTP workload by using a benchmark. We also consider strategies of database mutations which are adaptive to workload and prevent structural deterioration from being accelerated.

### 1. はじめに

オンライントランザクション処理(Online Transaction Processing: OLTP)は、電子金融取引や電子商取引などにおいて、ACID特性を保証するために不可欠なデータ処理方式

<sup>\*</sup> 学生会員 東京大学大学院情報理工学系研究科  
[hoshino@tkl.iis.u-tokyo.ac.jp](mailto:hoshino@tkl.iis.u-tokyo.ac.jp)

<sup>\*</sup> 正会員 東京大学生産技術研究所  
[kgoda.kitsure@tkl.iis.u-tokyo.ac.jp](mailto:{kgoda.kitsure}@tkl.iis.u-tokyo.ac.jp)

である。そのため、OLTP の高速化は今も重要な課題である。加えて、近年、業務やサービスの急激なデジタル化が進んだため、データベースの常時運用を実現する技術が求められている。

OLTP の高速化には二つのアプローチがある。一つは、データベース構造や処理方式の改善により、高速化を達成しようとするものである。もう一つは、データベース更新による構造劣化による性能低下を防ぐ取り組みである。データベース構造劣化とは、データベース更新が繰り返されると進行するデータ格納構造の非効率化現象である。

前者に関して、これまで、IO を意識したデータベースの格納構造の最適化方式[1, 2], CPU 命令やキャッシュを意識したデータ処理高速化方式[3], そして、並列化によるデータ処理高速化方式[4] などがOLTP 高速化のために研究されてきた。また、ハードウェア構成、データベーススキーマなどを負荷特性に応じて自動設定、自動調整することで高速化を目指す研究もなされている[5, 6]。後者に関して、一般に、OLTP においてはデータ更新が大量かつ継続的に行われるため、構造劣化の必要以上の進行を阻止し、OLTP 性能を確保することが不可欠である。このため、従来、管理者はデータベース再編成を実施し、データの再配置を行うことで、構造劣化を除去し、継続的なデータベース運用における性能低下を防いできた。再編成は負荷の高い処理であり、データベース常時運用においてその負荷を分散するために、OLTP と小規模な再編成を同時に実行する技術[7], ストレージ技術を用いてOLTP 負荷に影響を与えずにバックグラウンドで再編成を実施する技術[8] などの、オンライン再編成技術、また、構造劣化を継続的に監視し、再編成契機を自動的に決定するための技術[9, 10] が近年研究されている。以上のように、再編成によって構造劣化を除去する視点からの研究はなされているが、データ更新時に構造劣化を抑制するための技術、とりわけ、様々な負荷状況に応じて適応的に構造劣化を抑制する方法論はこれまで研究されてこなかった。

本論文は、データベースにおけるOLTP 負荷において、構造劣化に起因する性能低下現象を、ベンチマークを用いて分析し、データ更新パターンに適応的に構造劣化を抑制するデータ更新戦略について議論する。トランザクションあたりの構造劣化量を抑制し、性能低下を最小限に抑えることで、OLTP 処理の高速化とともに、再編成負荷の低減にも貢献すると期待される。

本論文は以下のように構成される。まず、第2章で、OLTP と構造劣化の関係について述べ、性能低下の原因となる現象を説明する。次に、第3章で、ベンチマークを用いて構造劣化によるOLTP 性能低下の分析を行い、第4章で、構造劣化を抑制するデータ更新戦略について議論する。第5章で結論と今後の課題を述べる。

### 2. OLTP と構造劣化

本章では、OLTP 負荷の特性について述べ、その後、起こりうる構造劣化およびその性能への影響について述べる。

OLTP 負荷と構造劣化の関係を一概に述べることは難しい。なぜなら構造劣化の性質はデータベース構造およびアクセスパターンに依存するからである。以後、本論文で扱う典型的な OLTP 負荷を例に説明する。TPC-C ベンチマーク[11]は、卸売り業者での処理を模擬し、OLTP ベンチマークの標準として扱われている。5つのトランザクションと、9つの表が定義されている。New-order トランザクションによって

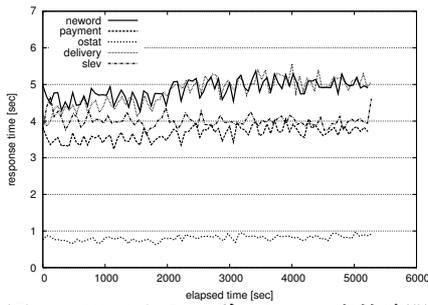
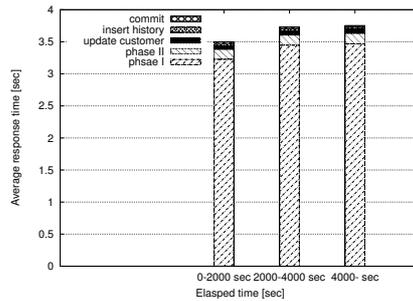
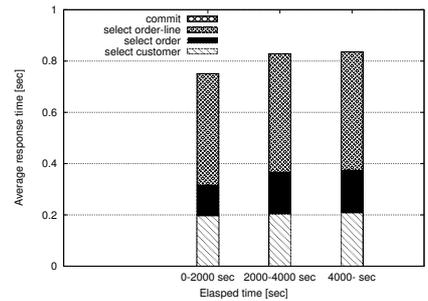


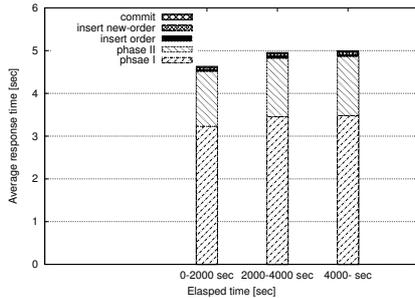
図 1 TPC-C トランザクション応答時間  
Fig.1 Response time of TPC-C



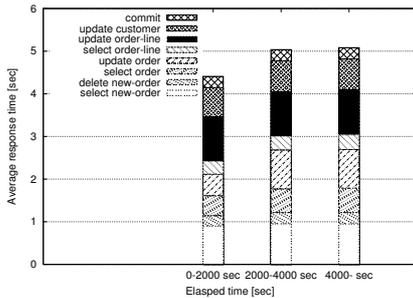
(b) Payment transaction



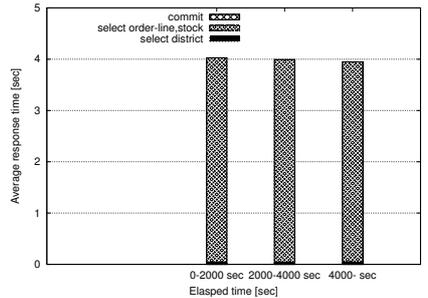
(d) Order-status transaction



(a) New-order transaction



(c) Delivery transaction



(e) Stock-level transaction

図 2 各 TPC-C トランザクション応答時間の内訳  
Fig. 2 Breakdown of response time of each TPC-C transaction

新規注文がデータベースに格納され、配達待ちになる。Delivery トランザクションは、配達待ちの注文を処理する。Payment トランザクションは、入金処理を行う。Order-status および Stock-level トランザクションは統計処理を行うが一切のデータ更新を行わない。トランザクション実行時のアクセスパターンは、各表から見て必ずしもランダムアクセスに近似できるわけではなく、主鍵順でのアクセス、すなわちシーケンシャルレコードアクセスも存在する。例えば、order 表、order-line 表、new-order 表におけるレコードは、一連のトランザクションによって order ID (o\_id)順に挿入およびアクセスされる。order ID は、主鍵を構成する複合鍵に含まれるため、一連のトランザクションによる物理格納構造における連続的なアクセスが期待される。すなわち、アクセスの局所性が高い表が存在する。これは、一般的な OLTP データベースにおいても同様であると考えられる。

構造劣化現象は、高水準位無効化、クラスタ化度低下、レコード断片化、充填率低下など様々である。例えば、データベース格納構造として B+木を考える。これは、MySQL InnoDB データベース[12]のクラスタ表や、Oracle 索引構成表に採用されている構造であり、二次索引にも利用されている。レコードは、B+木の葉ページ内に主鍵(クラスタ鍵)順に格納され、範囲検索を含む、主鍵を用いた高速レコードアクセスが可能である。B+木構造におけるクラスタ化度低下、充填率低下は多くのデータベースで発生しうる。一般に OLTP 負荷においては、クラスタ化度低下による性能低下は発生しないため、本論文では、充填率低下に焦点を絞って分析を行う。

データページ内でレコード格納に使用できるサイズに対する、実際に格納されているデータレコード合計サイズの比を、ページ充填率と呼ぶ。充填率が高ければ、データあたりに必要なページ数は少なく、特にリード系アクセスが高速になる。しかし、データベース更新により、充填率が低下する

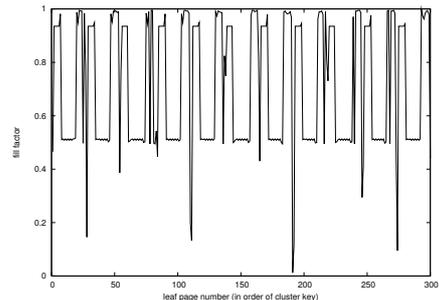


図 3 OLTP 負荷投入後の Order 表における各葉ページの充填率  
Fig. 3 Fill factor of each leaf page after issuing OLTP workload

と、データあたりのページ数が増え、必要なページアクセス数が増えるため、性能が低下する。

B+木構造において、満杯のページに対してレコード挿入を行おうとすると、ページ分割が発生し、充填率が低下する。MySQL の場合、ページ分割の方式は 2 種類存在する。一つは、ページ内で、鍵値が最大であるレコード挿入が連続して起きている場合、シーケンシャル挿入が行われていると認識し、ページ分割時に、新しく挿入されるレコードのみ新ページに格納する。すなわち、充填率はほぼ 100%で格納され、今後続くと予想されるシーケンシャル挿入に対して、充填率ほぼ 100%を保つことができる。これは、主にデータロード時に発生するアクセスパターンを想定している。もう一つは、ページ分割時に、半分ずつ新旧ページにレコードを格納するものである。すなわち、それぞれ約 50%の充填率になる。これは、ランダムレコード挿入に対して、さらなるページ分割を出来るだけ遅らせるために有効である。何らかのアクセスパターンによって、多くのページにおいて充填率が低下した場合、性能低下の恐れが高まる。

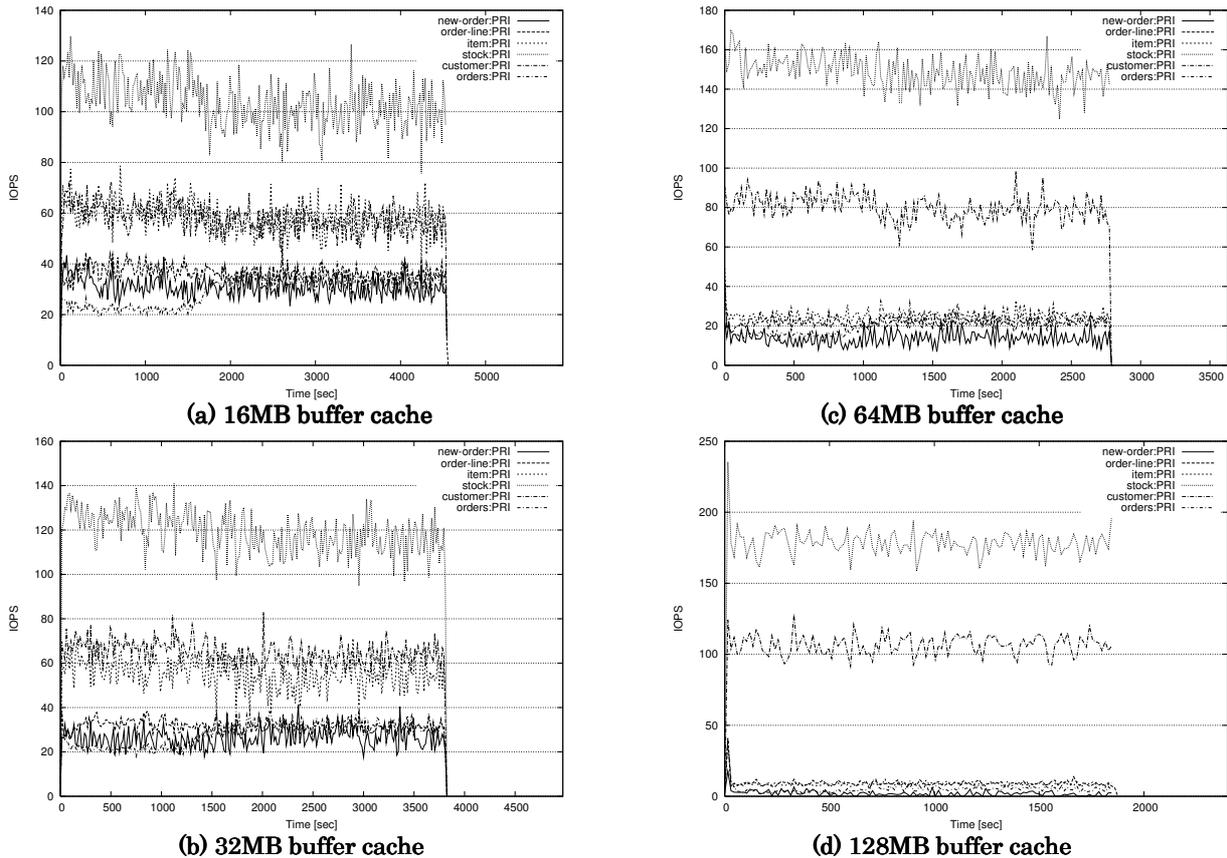


図 4 TPC-C ワークロードにおける IOPS  
Fig. 4 IOPS on TPC-C workload

### 3. 構造劣化による OLTP 性能低下分析

本章では、OLTP 負荷における充填率低下の発生メカニズム、およびその性能への影響を実験によって分析する。

#### 3.1 環境と設定

実験環境として、Linux PC 上でデータベースソフトウェア MySQL 5.0 InnoDB をバッファサイズ 16MB で、OLTP ベンチマークとして TPC-C Rev. 5.6 を使用した。データベース空間として、4GB を raw デバイスを用いて確保し、全ての表は、クラスタ表で構成し、レコードは固定長とした。ウェアハウス数は 16 とし、約 1.6GB がデータロード後に使用された。トランザクション比率は New-order:45%, Payment:45%, Order-status:2%, Delivery:6%, Stock-level:2% とした。データロード後、各ウェアハウスにつき 5 並列で、思考時間 0 にて計 100,000 トランザクションを実行し、応答時間を計測した。また、トランザクション比率を変え(New-order:47%, Payment:47%, Delivery:6%), データベースシステムのバッファサイズをパラメータとし、同様の実験を行い、IOPS を測定した。

#### 3.2 結果

最初の実験において、平均スループットは、1138tpm (transaction per minutes)であった。図 1 に全トランザクションの応答時間の推移を示す。Stock-level トランザクションを除く全ての応答時間が OLTP 負荷投入後 2,000 秒から 2,500 秒付近で増加していることが確認できる。図 2 において、各トランザクションの応答時間の内訳を、2,000 秒毎に示した。各内訳は個々の SQL を表し、insert, delete などの

操作と対象表を示している。図 2(a)において、Phase I は、select warehouse, select customer, select district, update district から成り、Phase II は、select item, select stock, update stock, insert order-line から成る。同様に、図 2(b)において、Phase I は update warehouse, select warehouse, update district, select district から成り、Phase II は、select customer から成る。New-order, Payment トランザクションにおける応答時間増加は、それぞれ主に Phase I の増加時間であり、warehouse, district 表は小さい表であるため、他の性能低下によって引き起こされた排他待ち時間の増加が原因であると考えられる。Delivery, Order-status トランザクションにおいて、order 表へのアクセス応答時間が主に増加しているのが確認できる。図 3 に、実験後の order 表における各葉ページの充填率を、クラスタ鍵順に示した。order 表の主鍵は warehouse ID (w\_id), district ID (d\_id), order ID (o\_id) から成る複合鍵である。データロード後は、各ウェアハウス、ディストリクト毎に 1 から 3,000 まで 3,000 個の o\_id を持つレコードが格納されている。その後、New-order トランザクションにより、3,001 の o\_id から順に挿入される。データロード時は、充填率約 100% で格納されているが、New-order トランザクションによって挿入された部分は、充填率が約 50% になっている。複合鍵からなる表の中途位置に発生するシーケンシャルレコード挿入であるため、現在の機構ではシーケンシャル挿入であると検知できず、通常の 50% ページ分割が実行されたものと考えられる。Delivery トランザクションは、2,101 の o\_id を持つレコードから順に処理する。このため、最初の 900 レコード処理と、その後の処理に

において、充填率に2倍の開きがある領域をアクセスすることになり、IOPSが増えることにより性能が低下すると思われる。次の実験において、図4に、各表のIOPSの推移を示した。この結果から、order表のIOPSが約2倍になることが確認された。ただ、バッファサイズが大きくなると、IOPSの増加はほとんど全体の性能に影響を与えなくなることも確認された。これは、注文済みかつ配達待ちに関するレコード群が頻繁にアクセスされるホットスポットであり、このホットスポットが全てバッファキャッシュ上に保持できるほどバッファキャッシュサイズが大きければ性能には直接影響しないものと考えられる。ただ、充填率低下により実質的なホットスポットが大きくなり、メモリの無駄遣いをしていることが推察される。TPC-Cベンチマークはホットスポットが比較的小さいため、実社会のOLTPデータベースにおいては、充填率低下の影響がより大きくなる恐れがある。

#### 4. 構造劣化抑制のためのデータ更新戦略

紙面の制約から詳細は省くが、前章の結果を考慮し、シーケンシャルレコード挿入検知を工夫することにより、複合鍵クラスタ表においても、シーケンシャルレコード挿入を検知し、ページ分割時の適切な充填率調整を行うことが可能であると考える。TPC-Cベンチマークの場合、order表においては、シーケンシャルレコード挿入及びレコード更新が行われるため、固定長レコード設定の場合は、ページ分割時の充填率をデータロード時と同様の100%にすることにより、前章で確認された充填率低下および性能低下は発生しない。また、データ増加レートなどの情報を用いて、より詳細なアクセスパターンを把握し、近未来のアクセスパターン予測が出来た場合に、長期的にページ分割数が最小になるように充填率調整を行うことで、充填率を高めることが可能になり、再編成に必要なコストを削減できると期待される。

#### 5. おわりに

本論文では、データベース構造劣化のOLTP性能低下への影響について実験を通して分析し、充填率低下がホットスポット拡大、しいては性能低下を引き起こすことが明らかになった。これらの結果から、充填率低下を抑制するデータ更新戦略を考察し、ページ分割時充填率調整手法について考察した。今後の課題として、より一般的なアクセスパターン認識による充填率調整手法について研究を進めたい。

#### 【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。深謝する次第である。

#### 【文献】

- [1] Theodore Johnson and Dennis Shasha. B-trees with inserts and deletes: Why free-at-empty is better than merge-at-half. *Journal of Computer and System Sciences*, Vol. 47, No. 1, pp. 45–76, 1993.
- [2] David B. Lomet. Simple, robust and highly concurrent b-trees with node deletion. In *ICDE*, pp. 18–28, 2004.
- [3] Richard A. Hankins and Jignesh M. Patel. Effect of node size on the performance of cache-conscious b+-trees. In *SIGMETRICS*, pp. 283–294, 2003.

- [4] David DeWitt and Jim Gray. Parallel database systems: the future of high performance database systems. *Communications of the ACM*, Vol. 35, No. 6, pp. 85–98, 1992.
- [5] Sanjay Agrawal, Vivek R. Narasayya, and Beverly Yang. Integrating vertical and horizontal partitioning into automated physical database design. In *SIGMOD Conference*, pp. 359–370, 2004.
- [6] Daniel C. Zilio, Jun Rao, Sam Lightstone, Guy M. Lohman, Adam Storm, Christian Garcia-Arellano, and Scott Fadden. Db2 design advisor: Integrated automatic physical database design. In *VLDB*, pp. 1087–1097, 2004.
- [7] Chendong Zou and Betty Salzberg. On-line reorganization of sparsely-populated B+-trees. In *Proc. ACM SIGMOD Int. Conf. Management of Data*, pp. 115–124, 1996.
- [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*, Vol. 46, No. SIG 8(TOD 26), pp. pp.130–147, 2005.
- [9] 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)*, Vol. 5, No. 2, pp. 37–40, 2006.
- [10] 星野喬, 合田和生, 喜連川優. 関係データベースにおける構造劣化監視機構を用いた再編成スケジューラの提案. *日本データベース学会論文誌(DBSJ Letters)*, Vol. 5, No. 1, pp. 101–104, 2006.
- [11] TPC: Transaction Processing Performance Council. TPC BENCHMARK C Standard Specification. <http://www.tpc.org/>.
- [12] MySQL: The World's Most Popular Open Source Database. <http://www.mysql.com/>.

#### 星野 喬 Takashi HOSHINO

東京大学大学院情報理工学系研究科博士課程在学中。日本学術振興会特別研究員 DC. 2003 年東京大学工学部電子情報工学科卒業。2005 年東京大学大学院情報理工学系研究科修士課程修了。データベースシステムの研究に従事。本会、情報処理学会、ACM、IEEE 学生会員。

#### 合田 和生 Kazuo GODA

2000 東京大学工学部電気工学科卒業、2005 同大学院情報理工学系研究科電子情報学専攻博士課程単位取得満期退学。博士(情報理工学)。現在、東京大学生産技術研究所特任助教。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、IEEE CS、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 同大学院工学系研究科情報工学専攻博士課程修了。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)。