

ロックフリー索引のトライ木化による高速化に関する研究

井戸 佑^{1,a)} 杉浦 健人^{1,b)} 石川 佳治^{1,c)} 陸 可鏡^{1,d)}

概要：代表的な索引構造である B+木は様々なデータを格納可能な汎用性の高い索引である．一方で，扱うキーに制限を加え，索引構造を最適化させることで性能を向上させる研究も行われてきた．本研究では，著者らの研究室で開発しているロックフリー B+木に対し同様の拡張および性能改善を行う．具体的には，扱うキーをバイナリ比較可能なものに制限することでトライ木の構造を適用し，空間利用効率の向上およびそれに伴う検索性能の向上を図る．

1. はじめに

ムーアの法則の終焉により，CPU のコア単体性能は限界に達しつつある．一方で，IT 技術の発展に伴い管理すべきデータは爆発的に増えつつある．この状況に対処するため，現在のコンピュータ技術は複数のコアを用いて処理を行うマルチスレッド処理が主流である．データベース分野においても例外ではなく，近年メニーコアなどを前提としたインメモリデータベースの研究が進んでいる．データベースの構成要素の 1 つである索引技術も同様に，メニーコア・大容量メモリに適合させる必要がある．

代表的な索引構造である B⁺ 木 [4] では，ロックを用いた同時実行制御が行われている．しかし，マルチスレッド処理においてロックによる同時実行制御は多数の待ちスレッドが発生するため，スケラビリティが悪化する．そこで，B⁺ 木をロックフリー化させた索引として Bw 木 [2] や Bz 木 [1]，著者らの研究室で開発しているロックフリー B+木 (B^c 木) が提案されている．

また，インターネットの普及に伴い URL の管理や EC サイトにおける文字列検索など，特定のキーに対し効率的に処理する索引構造が求められている．Mass 木 [3] は，文字列型や整数型などのバイナリ比較可能なキーに特化した索引構造の 1 つである．B⁺ 木を階層的に作成することにより，キャッシュ効率を改善している．

本研究の B^c-forest では，著者らの研究室で開発しているロックフリー B+木 (B^c 木) に対し，Mass 木と同様の拡張および性能改善を行う．特に，本論文ではその構造およ

び操作について述べる．

本稿の構成は以下の通りである．2 章では，関連研究としてロックフリー索引やバイナリ比較可能なキーに対し最適化した索引について概説する．次に，3 章で B^c-forest の構造について説明し，4 章および 5 章で B^c-forest の操作について述べる．最後に，6 章で本稿のまとめと今後の方針を述べる．

2. 関連研究

2.1 B^c 木

2.2 Masstree

3. B^c-forest の構造

4. B^c-forest のノード操作

4.1 書き込み

4.2 読み取り

5. B^c-forest の構造変更操作

5.1 統合

5.2 分割

5.3 新層作成

6. おわりに

本稿では B^c 木に Mass 木と同様のトライ木構造を適応させた B^c-forest について提案し，その構造および操作を紹介した．今後は提案した索引構造を実装するとともに，Mass 木や近年提案されている Bw 木や Bz 木といったロックフリー索引との性能を比較検証する．

謝辞 本研究は JSPS 科研費 JP20K19804，JP21H03555，JP22H03594，JP22H03903 の助成，および国立研究開発法人新エネルギー・産業技術総合開発機構 (NEDO) の委託

¹ 名古屋大学大学院情報学研究科
Graduate School of Informatics, Nagoya University

a) ido@db.is.i.nagoya-u.ac.jp

b) sugiura@i.nagoya-u.ac.jp

c) ishikawa@i.nagoya-u.ac.jp

d) lu@db.is.i.nagoya-u.ac.jp

業務（JPNP16007）の結果得られたものである。

参考文献

- [1] Arulraj, J., Levandoski, J., Minhas, U. F. and Larson, P.: BzTree: A High-Performance Latch-free Range Index for Non-Volatile Memory, *PVLDB*, Vol. 11, No. 5, pp. 553–565 (2018).
- [2] Levandoski, J., Lomet, D. and Sengupta, S.: The Bw-Tree: A B-tree for New Hardware Platforms, *Proc. ICDE*, pp. 302–313 (2013).
- [3] Mao, Y., Kohler, E. and Morris, R. T.: Cache Craftiness for Fast Multicore Key-Value Storage, *Proc. EuroSys*, pp. 183–196 (online), DOI: 10.1145/2168836.2168855 (2012).
- [4] 北川 博之: データベースシステム, 昭晃堂 (1996).