

データベースチューニングとしての B-Tree 索引構造の特性解析

佐藤 英幸 石井 茂雄

1. 背景と目的

データベースから高速にデータを検索する強力な手段として B-Tree 索引がある。だが B-Tree 索引は、常にデータベースのパフォーマンス向上に繋がるとは限らない。B-Tree には以下に述べる問題点がある為である。

- (1) 追加、削除が多いときには効率低下を招く。
- (2) 索引用の固定記憶領域が大きくなる。

Read/Write の回数と B-Tree のブロックサイズとはトレードオフの関係が生じる。この問題は性能チューニングでは特に重要となる。B-Tree を改良した索引構造として B*-Tree がある。B-Tree との相違点は、連続したアイテムの読み込みの効率化である。リーフブロック間を、フタを用いてリンクを繋げる手段を用いている。本研究では B-Tree、B*-Tree 索引構造における特性を検証するため、B-Tree、B*-Tree 索引構造を実現したプログラムを試作し、各索引構造の特性を解析すると共に、相違点による有用性の変化を検証することを目的とする。

2. 研究内容

2.1. 特性による効率性

B-Tree 索引構造のもつ効率性を左右する特性として次のものがある。

- (1) ブロックを構成するアイテム数
(以降、ブロックサイズと称する)
- (2) 登録するアイテムの総数
- (3) 検索 / 書き込み処理の割合

これらをパラメータとしてチューニングする。トレードオフ分析を用いて、最適なブロックサイズを求める。

2.2. B-Tree と B*-Tree 索引構造の比較

連続したアイテムの読み込みのアクセス回数及び、索引サイズを測定する。独立変数として、ブロックサイズと連続して読むアイテム数を用いる。

3. 実験結果及び考察

測定はアイテムを 10 万件を登録した状態で行う。キーは 0~99,999 を順に登録する。独立変数であるブロックサイズは 2 の乗数を設定する。

3.1. B-Tree 索引構造の持つ特性による効率性の変化

図 1 よりブロックサイズを増加させることで、平均アクセス回数とその変化規模が減少するのが分かった。登録件数を変えた場合、変化が変わらず約 2, 3 回のアクセスで落ち着く。トレードオフの関係により、128~256「アイテム/ブロック」が最適なブロックサイズと言える。

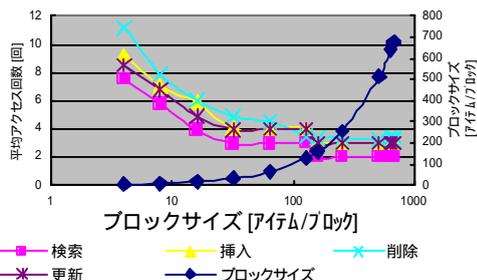


図 1. ブロックサイズと平均アクセス回数

3.2. B*-Tree と B-Tree の相違

図 2 よりブロックサイズの設定が小さい時には、大きな差が見られる。平均アクセス回数の変化が落ち着いたあたりから、両者の差はなくなった。総ブロック数が減少した為、差が無くなったと考えられる。結果、ブロックサイズは平均アクセス回数だけでなく B*-Tree 索引のサイズにも大きく影響することがわかった。

図 3 では、全体の 100% を連続で読んでみる。図 3 を見ると、連続したアイテムの読み込みでは、B*-Tree のほうが有利なことが明らかである。この実験で索引サイズだけでなく、連続読み込みでのアクセス回数もブロックサイズの増加に伴って差が縮む事が分かった。50%, 10% でも同様の結果が得られた。索引の階層が浅くなることによって、B-Tree での処理が緩和される為と考えられる。

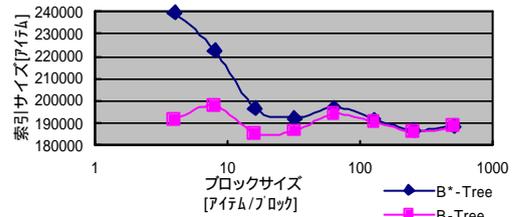


図 2. ブロックサイズと索引サイズ

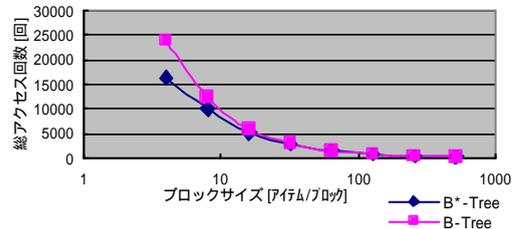


図 3. アイテムの連続読み込み

3.3. 索引サイズの変化

ブロックサイズが増加するに伴って、索引サイズも増加する傾向にあると考えられた。だが、図 2 より索引サイズが増加するとは限らないことが分かる。このような波形のグラフになる理由として、索引のアイテム密度が変化するためと考えられる。しかし、アイテム密度は登録する手順によっても大幅に変わる。そのため、ブロックサイズの設定だけで最適なポイントを見つけることは無理である。B-Tree 索引への登録手順が重要になると考えられる。

4. まとめ

本研究により、ブロックサイズはアクセス回数だけでなく、B*-Tree 索引のサイズにも影響することが分かった。索引サイズの縮小はコスト削減に繋がる為、今後の課題として注目できる内容と言える。

参考文献

Mark Gurry, Peter Corrigan 著: 「Oracle パフォーマンスチューニング第 2 版」オライリージャパン (1998)