Juergen Klonk

Elkartallee 11
3000 Hannover 1
W-Germany
7/27/1982

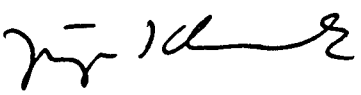
To SIGMOD NEWSLETTER editor
Tom Cook
Tektronix,Inc.
P.O. Box 500
M/S:50-384
Beaverton, OR 97077

Dear Mr. Cook,

Enclosed you find some comments on 'optimality in B-trees', which you may publish in SIGMOD-NEWSLETTER if you find them of sufficient interest.

I wonder whether SIGMOD can influence the editorial policy of TODS. It seems to me that the editors of TODS are basing their decisions more and more on formal and mathematical quality and less on practical relevance. This is a develop-ement which I sincerely regret. After all data bases are part of the real world, and not just a motivation for formal definitions.

Sincerely, Yours



Juergen Klonk

Comments on Optimality of B-Trees

Juergen Klonk

Elkartallee 11
Hannover, West Germany

In a recent paper /3/ A.L. Rosenberg and L.Snyder investigate two measures for optimality of B-trees, namely 'Space-optimality' and 'Time-optimality'. A B-tree of a given degree M and a given number n of keys is defined to be time-optimal, if it has a minimal average number of node-accesses, and space-optimal, if it has a minimal number of nodes. Rosenberg/Snyder show that the criterium of time-optimality is not very useful, but they propose space-optimality to be used in the processing of real B-tree files.

On this I want to make the following comments:

- Space-optimality is extremely volatile.
- The cost of maintaining space-optimality is very high.
- The problem itself is only of academic relevance.

1.) Persistence of Space-Optimality

One of the main results of Rosenberg/Snyder is that as many as possible nodes on the lowest levels should be completely filled. This is contrary to practionars beliefs. The following arguments show that space utilization decreases severely after few insertions into such a tree.

Let T1 be a space-optimal B-tree of order M with n keys. We make the following simplifying assumptions:

- n and M are not small, say $M > 20$ and $n > 10000$.
- We consider only the lowest level of the tree. (Since it containes the vast majority of the keys.)
- All nodes in the lowest level of the tree are full, i.e. contain (M-1) keys.

The number v of nodes is therefor:

$$v = n / (M-1) \qquad \text{equ. (1)}$$

Now a sequence of d keys is inserted into tree T1 in random order, resulting in tree Td. Every full node where a key is to be inserted splits. d shall be much smaller than n. It is therefor justified to assume that every node splits at most once. Let sp(d) be the number of nodes in T1 which have split after these d inserts. Using Rosenberg/Snyder's definition of space-usage NU, we get:

$$NU(Td) = (n+d) / ((M-1) * (v + sp(d))) \qquad \text{equ. (2)}$$

With the following small statistical model we can derive the expectation for sp(d):
There are v nodes, and d experiments to hit those nodes. The probability that a single node is missed in all d experiments is:

$$(1 - 1/v)^d$$

The probablity that a node splits is the probability that it is hit. This is equal to:

$$1 - (1 - 1/v)^d$$

This applies for all v nodes, and therefore the expectation of the number of node-splits is:

$$sp(d) = v * (1 - (1 - 1/v)^d) \qquad \text{equ. (3)}$$

Rosenberg/Snyder use instead of the number d the degree of change alpha:

$$d = n * (1 - alpha) \qquad \text{equ. (4)}$$

Combining equations (1) to (4) yields as expected node utilization ENU for tree Td:

$$ENU(Td) = \frac{2 - alpha}{2 - (1 - \frac{M-1}{n})^{n*(1-alpha)}}$$

Using the rules of L'Hospital we get for big n:

$$LENU(Td) = \lim_{n \to \infty}(ENU(Td)) = \frac{2 - alpha}{2 - e^{M*(alpha-1)}}$$

The following table lists a number of values of this function:

M	alpha	LENU
10	0.99	0.922
75	0.99	0.661
150	0.99	0.568
150	0.996	0.692
150	0.999	0.879

Two observations can be made:

- For M=150 the tree is nearly half empty after only one percent inserts.
- The space-utilization becomes worse with increasing M. Concentrating on M=3 (as Rosenberg/Snyder do) therefore

gives an over-optimistic picture of space-optimality.

2.) Cost of Maintaining Space-Optimality

Rosenberg/Snyder present an algorithm which re-establishes space-optimality in a B-tree, and they suggest that this algorithm is executed 'during' the daily backup of the file.

There seems to be a confusion of 'backup' and 'reorganisation': Rosenberg/Snyder's algorithm is in fact an in-place reorganisation, and as such it is dangerous to the security of the file. In-place reorganization should always be preceded by backup, and should not replace backup.

Furthermore is the daily backup normally performed with programs which are typically faster than reorganization by a factor of 10.

3.) Relevance of the Problem

In their first B-tree paper /1/ Bayer and McCreight showed that a space utilization of 85% can be expected if a simple refinement of the node-splitting algorithm is employed. In /2/ this figure is confirmed using analytical methods. The 'potential' for saving of space is therefore around 15%. In a typical situation the index (the B-tree) may need about 15% of the total space of the file. This leaves a potential of just 2% of the total space which can be saved.

Consider now an average programer allocating space for his file. He adds some space for possible miscalculations on his side (say 20%) and for future growth of the file (say another 30%). He thus leaves about one third of the total allocated space unused. So: why do we bother about two percent, if 33 percent are wasted elsewhere? The answere is simple: Only the 2%-problem can be tackled with formal methods and only formal methods are adequate for computer scientists.
And who cares for the practical relevance of the problem.

/1/ Bayer,R. and McCreight,E.: Organization and Maintenance of Large Ordered Indexes. Acta Informatica 1, pp. 173-189 (1972)

/2/ Klopprogge, M.R. and Qutizow, K.H.: Space Utilization and Access Path Length in B-Trees. Information Systems Vol. 5, pp.7-16 (1980

/3/ Rosenberg,A.L. and Snyder,L.: Time- and Space-Optimality in B-Trees. ACM Trans. Database Syst. Vol. 6, pp. 174-193 (1981)