
Fringe Analysis of 2–3 Trees with Lazy Parent Split

ANTIGONI MANOUSAKA AND YANNIS MANOLOPOULOS

*Department of Informatics, Aristotle University, Thessaloniki 54006, Greece
Email: manolopo@delab.csd.auth.gr*

B-trees with lazy parent split (lps) are B-tree variants, according to which parent splits are postponed until a future access of the latter node. This way, the number of splits during an insert is decreased and the number of locks is also decreased. Consequently, better concurrency is achieved. In this paper 2–3 trees with lps are studied. Fringe analysis is used to obtain bounds on some performance metrics of 2–3 trees with lps. The performance metrics of 2–3 tree with lps are compared with those of the classical 2–3 trees. The conclusions are that 2–3 trees with lps have slightly better performance, more keys in the fringe, larger storage utilization and a slightly shorter path length from the root to the leaves.

Received 19 May 1998; revised 12 September 2000

1. INTRODUCTION

In a multi-user database environment, a basic task is to perform concurrent operations on the underlying access methods. For this reason, locking is used to avoid inconsistency of data. In the past, several variants and algorithms have been proposed to support locking for the structures of the B-trees family, which is the most important access method used for storing alphanumeric data in any commercial database management system [1, 2, 3, 4].

A B-tree with lazy parent split (lps) is a B-tree variant aiming at improving concurrency [5, 6, 7, 8], which is an important parameter in database management systems and file organizations for multi-user environments. The lps technique is used to decrease the number of node splits during an insert operation; therefore, the number of locks is decreased and as a consequence enhanced concurrency is achieved. It has been shown that B-trees with lps achieve better storage performance [6]. However, a theoretical proof of this remark was lacking from the literature. On the other hand, fringe analysis is a tool used to derive expressions for various tree characteristics [9, 10, 11, 12]. In the present paper, fringe analysis will be used to study the performance of 2–3 trees (e.g. B-trees with degree $d = 1$) with lps. Generally, an improvement of one parameter of a structure, such as concurrency, can only be achieved at the cost of other parameters. We shall prove that in 2–3 trees with lps the gain in concurrency is accompanied by larger storage utilization, more keys in the fringe etc.

The rest of this work is organized as follows. In Section 2, the structure of B-trees with lps is defined, and insertion examples are given to demonstrate the basic mechanism. In Section 3 the fringe analysis method is briefly described. In Section 4 the model of a 2–3 tree with lps suitable for fringe analysis is formalized and fringe analysis is performed. In Section 4.2 the lps technique is restricted to the lowest level.

In Section 4.3 the lps technique is used in any level, except in the case of Theorem 14 where it is assumed that the lps technique is restricted to the two lowest levels. Finally, the last section contains a discussion on various issues and carries out a comparison with standard 2–3 trees.

2. B-TREES WITH LAZY PARENT SPLIT

A B-tree of degree d is a balanced tree with the following properties:

- every node except the root has at least d keys and at most $2d$ keys,
- the root is either a leaf or has at least one key and at most $2d$ keys,
- every k -key node except the leaves has $k + 1$ children, and
- all leaves appear at the same level.

Insertion of a new key into a node containing less than $2d$ keys is an easy task. To insert a new key into a $2d$ -key node, the latter node is split into two d -key nodes, the middle key is inserted into the parent and the insertion process is repeated one level up. For $d = 1$ the B-tree is called a 2–3 tree.

A B-tree with lps of degree d is a balanced tree with the following properties [5, 6]:

- every node except the root either has at least d keys and at most $2d$ keys or is a double node,
- the root is either a leaf or has at least one key and at most $2d$ keys or is a double node,
- a double node consists of two sibling nodes,
- every k -key node except the leaves has $k + 1$ children, where no more than one child can be a double node,
- a node is called a flagged node if one of its children is a double node, when a flagged node is accessed due to any operation, a clean-up operation is performed, and
- all leaves appear at the same level.

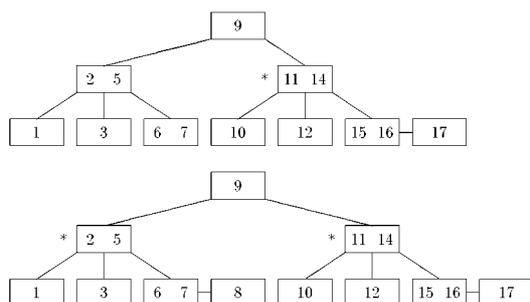


FIGURE 1. Insertion of key 8 into a 2-3 tree with lps (symbol * shows the flagged node).

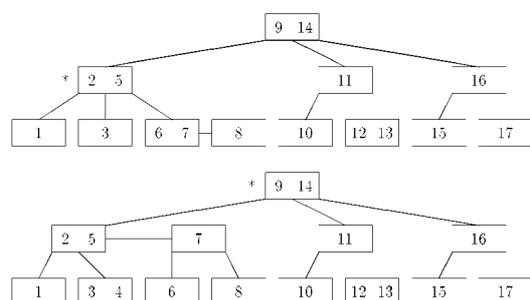


FIGURE 2. Insertion of keys 13 and 4 into the last instance of Figure 1.

In [6] insertion and deletion algorithms have been given. When inserting a new key into a node containing less than $2d$ keys, it is placed among the other keys. When inserting a new key into a $2d$ -key node, the parent is checked first. If the parent has less than $2d$ keys, then the node is split into two d -key nodes and the middle key is placed among the parent node keys. If the parent has exactly $2d$ keys, then first the node is temporarily split into two d -key nodes, and then the middle key is inserted into the first of the two nodes to form a double node. In this last case, no new key is posted to the parent but a flag is set to indicate the child which is a double node. In Figure 1 a simple insertion example is shown for a 2-3 tree with lps, where sibling nodes are connected with a horizontal pointer.

The next time that the flagged node is accessed due to any operation, a clean-up operation will be performed before any subsequent operation takes place below the specific point. More specifically, the double node will become two ordinary nodes and the insertion process will continue by posting the middle key (temporarily stored in the left sibling) up to the parent node. This means that the flagged node, the left node of its double node child, and perhaps the parent of the flagged node (if it has less than $2d$ keys), will be changed, and thus they have to be locked. This way, the number of locked nodes is minimized and improvement in concurrency is achieved, while the clean-up operation is performed during the top-down traversing towards the leaf level. The example in Figure 2 demonstrates this case.



FIGURE 3. The closed tree collection of 2-3 trees of height 1.

As a consequence of the lps technique, paths from the root down to leaves do not have the same length. Every time a right sibling node is accessed the path length is increased by one, although a level change is not considered to be performed. Classical fringe analysis considers only insertions [11]; some attempts to include deletions in the analysis have been made for AVL-trees and B-trees. However, the random model does not apply after deletions, and a different model must be used [9]. Such a model, called the ‘ghost’ model, was used in [13]. The term ‘ghost’ has been used for the reason that, although an item may have been deleted, it destroys randomness and affects the probability distribution of subsequent insertions. For this reason, no further explanations are given about how clean-up is performed in the case of deletions.

3. FRINGE ANALYSIS

Fringe analysis is a method first introduced by Yao in 1978 [12] and formalized by Eisenbarth *et al.* in 1982 [11]. The interested reader may find a recent review on fringe analysis in [9].

A tree with N keys divides all possible key values into $N + 1$ intervals. An insertion into the tree is said to be a random insertion if it has an equal probability of corresponding to any of these $N + 1$ intervals. A random tree is a tree constructed by random insertions into an initially empty tree. Fringe analysis considers random insertions and random trees. We call the analysis of the lowest level of a tree ‘first order analysis’ and, generally, the analysis of the k lowest levels ‘ k -order analysis’. The fringe of a tree during k -order fringe analysis is obtained by deleting all nodes at a distance greater than k levels from the leaves.

The isomorphic subtrees of the fringe are considered as one tree type, so that a tree collection C consisting of all possible tree types, which are numbered sequentially, can be derived. We say that a tree collection C is ‘closed’ if [11]:

1. for all T in C an insertion into T always leads to one or more members of C , and
2. the effect of an insertion on the composition of the fringe of a random tree is determined only by the subtree of the fringe in which the insertion is performed.

Figure 3 shows the closed tree collection of 2-3 trees with height equal to 1. Fringe analysis assumes a closed tree collection. The probability that an insertion into one tree leads to another tree depends only on the types of the two trees involved.

The composition of the fringe can be described via the probability that a randomly chosen leaf of the tree belongs to each member of the corresponding closed tree collection.

We call a leaf a ‘type i leaf’ if it belongs to a tree of type i . The probability that a leaf is of type i in an $N + 1$ -leaves tree is

$$p_i(N) = (\text{Expected number of leaves of type } i \text{ in a tree with } N \text{ keys})(N + 1)^{-1}. \quad (1)$$

Let $A_i(N)$ be the expected number of trees of type i in a random search tree with N keys. Let L_i be the number of leaves of the type i tree. Equation (1) can be written as

$$p_i(N) = \frac{A_i(N)L_i}{N + 1}. \quad (2)$$

From [11] it is known that the probability of an insertion occurring in each of the subtrees of the fringe can be obtained from the steady-state solution of a matrix recurrence relation in a Markov chain. Let m be the number of members of the closed tree collection C . Let $p(N)$ be an m component column vector consisting of the probability values $p_k(N)$ as in Equation (1). Then

$$p(N) = \left(I + \frac{H}{N + 1} \right) p(N - 1) \quad (3)$$

where I is the $m \times m$ identity matrix and H is an $m \times m$ matrix called the transition matrix of the fringe analysis problem. The elements in the diagonal of H represent the number of leaves lost due to an insertion minus one, and off-diagonal elements represent the number of leaves obtained for each type times the probability that each type is reached in a transition. In general,

$$h_{ij} = \frac{1}{L_j} \sum_{k=1}^{L_j} l_{ij}(k) - \delta_{ij}(L_j + 1) \quad (4)$$

where δ_{ij} is the Kronecker symbol, L_j is the number of leaves of a j -type tree and $l_{ij}(k)$ is the number of leaves of type i that will be generated at an insertion into the k leaf ($k \in [1, \dots, L_j]$) of a j -type tree.

A fringe analysis is called ‘connected’ if there is an $l \in [1 \dots m]$ such that $\det(H_u) \neq 0$, where H_u is matrix H with the l th column and l th row deleted. Let $\lambda_1, \dots, \lambda_m$ be the eigenvalues of H . They can be ordered so that $\lambda_1 = 0$ and $0 > \text{Re } \lambda_2 \geq \text{Re } \lambda_3 \geq \dots \geq \text{Re } \lambda_m$.

Let x_1 be the right eigenvector of H corresponding to $\lambda_1 = 0$. Then there is a c such that for every vector $p(N)$

$$|p(N) - cx_1| = O(N^{\text{Re}\lambda_2}) \quad (5)$$

where $p(N)$ is defined by Equation (3).

Consider the $m \times m$ transition matrix H of a connected fringe analysis case. Equation (5) says that $p(N)$, the m -component column vector solution of Equation (3), converges to the solution of

$$Hq = 0 \quad \text{as } N \rightarrow \infty \quad (6)$$

where q is also an m -component column vector which is independent of N , and

$$p(N) = ax_1 + O(N^{\text{Re}\lambda_2}) \quad (7)$$

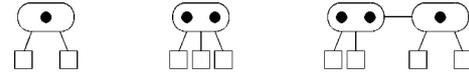


FIGURE 4. The first three steps.

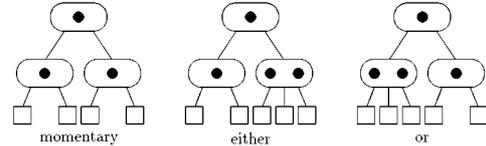


FIGURE 5. The fourth step.

where x_1 is the right eigenvector of H corresponding to eigenvalue $\lambda_1 = 0$, and a is a constant.

In the next section, fringe analysis will be used to examine the performance characteristics of 2–3 trees with lps.

4. ANALYSIS OF 2–3 TREES WITH LPS

4.1. Notation

In Figure 4 keys are indicated by dots, and the evolution of an initially empty 2–3 tree with lps, where 3 keys are inserted, is shown. Evidently, this is the tree collection for 2–3 trees with lps of height 1. If a new key has to be inserted in the rightmost instance of the latter figure, then according to the clean-up phase the leftmost instance of Figure 5 is derived. Afterwards one of the two remaining structures of the same figure will be produced.

The performance metrics which will be computed are defined as follows.

1. Let $\bar{f}(N)$ be the expected number of keys in the fringe of a tree.
2. Let $\bar{n}(N)$ be the expected number of ‘equivalent single nodes’ in a 2–3 tree with lps after the random insertion of N keys into an initially empty structure. During computation of $\bar{n}(N)$, every double node is counted as two nodes.
3. Let $\text{Pr}\{0 \text{ splits}\}$ be the probability that 0 splits occur on the $(N + 1)$ th random insertion into a structure with N keys.
4. Let $\bar{h}(N)$ be the expected path length of a tree with N keys. The length of a path in a B-tree with lps is the number of levels plus the number of right sibling nodes across the path. The length of any path in a B-tree is the tree height or the number of tree levels.

4.2. Second-order analysis

In second-order analysis the two lowest levels are examined. The lps technique is restricted to the lowest level to have a closed tree collection. The total number of 2–3 trees with lps of height 2 is 24 distinct structures. These 24 trees are grouped into 11 tree types so that symmetrical structures (i.e. structures with the same subtrees of level 1) belong to the same type. Figure 6 shows the two-level closed

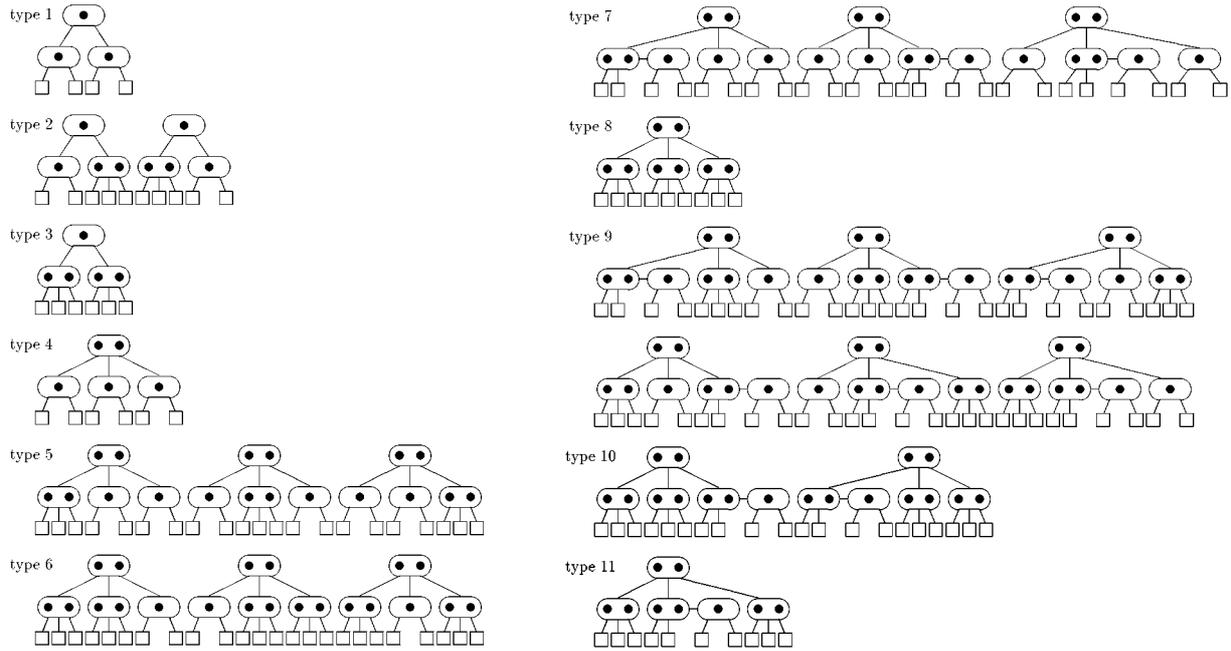


FIGURE 6. Closed tree collection 2-3 trees with lps of height 2.

-5	0	0	0	0	0	4	0	20/9	12/5	0
5	-6	0	0	0	0	5	0	40/9	2	5
0	12/5	-7	0	0	0	0	0	4/3	12/5	12/5
0	18/5	0	-7	0	0	0	0	2	0	18/5
0	0	7	7	-8	0	0	0	0	21/5	0
0	0	0	0	32/7	-9	0	0	0	0	0
0	0	0	0	24/7	0	-9	0	0	0	0
0	0	0	0	0	9/4	0	-10	0	0	0
0	0	0	0	0	27/4	0	0	-10	0	0
0	0	0	0	0	0	0	20/3	0	-11	0
0	0	0	0	0	0	0	10/3	0	0	-11

FIGURE 7. Transition matrix for the closed tree collection 2-3 trees with lps of height 2.

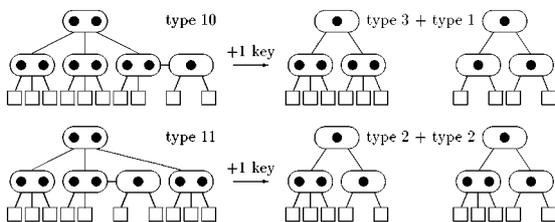


FIGURE 8. Insertions in type 10 and type 11 trees.

tree collection, whereas Figure 7 shows the corresponding 11×11 transition matrix.

Although types 10 and 11 seem symmetrical cases, they are treated as different because an insertion into any type 10 tree gives different tree types to those produced by an insertion into a type 11 tree. For instance, as depicted in Figure 8, any insertion into the left tree belonging to type 10

will produce one type 3 tree and one type 1 tree. On the other hand, any insertion into the right tree of type 10 will produce one type 1 and one type 3 tree. However, these tree types cannot be produced by any insertion into a type 11 tree.

Using Equation (6), Table 1 is obtained. The eigenvalues of H of Table 1 are

$$0, -6.55 \pm 6.25i, -7, -9, -9.23 \pm 1.37i, -10, -11, -11, -13.44.$$

According to Equation (7) the asymptotic values of $p(N)$ are approximated to $O(N^{-6.55})$.

THEOREM 1. *The expected number of keys in the fringe of a 2-3 tree with lps containing N keys is*

$$\bar{f}(N) = (N + 1) \frac{514,927}{615,307} + O(N^{-6.55}). \quad (8)$$

Proof. Let K_i be the number of keys of the type i tree as depicted in the closed tree collection of Figure 6. From

TABLE 1. Probability values $p(N)$ for a random 2–3 tree with lps of height 2.

p_1	$= 8496/87,901$	$\simeq 0.09665$
p_2	$= 17,880/87,901$	$\simeq 0.20341$
p_3	$= 55,152/615,307$	$\simeq 0.08963$
p_4	$= 7128/55,937$	$\simeq 0.12743$
p_5	$= 1575/7991$	$\simeq 0.19710$
p_6	$= 800/7991$	$\simeq 0.10011$
p_7	$= 600/7991$	$\simeq 0.07508$
p_8	$= 180/7991$	$\simeq 0.02253$
p_9	$= 540/7991$	$\simeq 0.06758$
p_{10}	$= 1200/87,901$	$\simeq 0.01365$
p_{11}	$= 600/87,901$	$\simeq 0.00683$

Figure 6 it is noted that $K_i = L_i - 1$ for all trees. From Equation (2) one obtains

$$\begin{aligned} A_i(N) &= (N+1) \frac{p_i(N)}{L_i} \Rightarrow \\ \bar{f}(N) &= \sum_i A_i(N) K_i = \sum_1^{11} \frac{(N+1)p_i K_i}{L_i} \\ &= (N+1) \sum_1^{11} \frac{p_i(L_i-1)}{L_i} \\ &= (N+1) \left(1 - \sum_1^{11} \frac{p_i}{L_i}\right). \end{aligned}$$

From Table 1 and Figure 6 one obtains

$$\begin{aligned} \bar{f}(N) &= (N+1) \frac{514,927}{615,307} + O(N^{-6.55}) \\ &\simeq 0.83686(N+1). \quad \square \end{aligned}$$

THEOREM 2. *The expected number of ‘equivalent single nodes’ in a random 2–3 tree with lps containing N keys is bounded by*

$$\begin{aligned} \frac{414,273}{615,307} N - \frac{213,239}{1,230,614} + O(N^{-6.55}) \\ \leq \bar{n}(N) \leq \frac{464,463}{615,307} N - \frac{150,844}{615,307} + O(N^{-6.55}) \end{aligned}$$

or, to five decimal places, by

$$0.67328N + 0.17328 \leq \bar{n}(N) \leq 0.75485N - 0.24515.$$

Proof. Let n_f be the number of equivalent single nodes in the fringe. Let n_a be the number of nodes above the fringe. Let N_i be the number of nodes of a tree of type i . Then

$$\bar{n}(N) = \bar{n}_f + \bar{n}_a.$$

The tree above the fringe is a 2–3 tree. Therefore, based on the 2–3 tree definition, it easily follows that

$$\frac{n_l - 1}{2} \leq n_a \leq n_l - 1$$

where n_l is the number of leaves of the tree above the fringe (or, equivalently, the number of the trees of the fringe),

$$\begin{aligned} \bar{n}_l &= \sum_1^{11} \frac{(N+1)p_i}{L_i}, \\ \bar{n}_f &= \sum_1^{11} \frac{(N+1)p_i N_i}{L_i}. \end{aligned}$$

Then

$$\begin{aligned} \frac{1}{2} \sum_1^{11} \frac{(N+1)p_i}{L_i} + \sum_1^{11} \frac{(N+1)p_i N_i}{L_i} - \frac{1}{2} \\ \leq \bar{n}(N) \leq \sum_1^{11} \frac{(N+1)p_i}{L_i} + \sum_1^{11} \frac{(N+1)p_i N_i}{L_i} - 1. \end{aligned}$$

This inequality, Table 1 and Figure 6 lead to the theorem. \square

THEOREM 3. *The probability that no split occurs on the $(N+1)$ th random insertion into a 2–3 tree with lps containing N keys is:*

$$\Pr\{0 \text{ splits}\} = \frac{272,644}{615,307} + O(N^{-6.55}).$$

Proof. An insertion into the type 1 tree shown in Figure 6 causes no split, an insertion into the type 2 tree causes no split two-fifths of the time, and so on. The following expression is obtained by observing Figure 6:

$$\begin{aligned} \Pr\{0 \text{ splits}\} &= P_1 + \frac{2}{5}P_2 + P_4 + \frac{4}{7}P_5 + \frac{2}{8}P_6 \\ &= \frac{272,644}{615,307} + O(N^{-6.55}) \simeq 0.44310. \quad \square \end{aligned}$$

THEOREM 4. *The expected path length of a 2–3 tree with lps containing N keys is upper bounded by*

$$\bar{h}(N) \leq \log_2(N+1) - 0.57795.$$

Proof. Let k_a indicate the number of keys above the fringe. The height h_a of the classic 2–3 tree above the fringe is bounded by

$$\lceil \log_3(k_a + 1) \rceil \leq h_a \leq \lfloor \log_2(k_a + 1) \rfloor$$

where the lower and the upper bound on the height are obtained when each node of the 2–3 tree contains 2 keys and 1 key respectively. Let h_f indicate the path length of the fringe. The following expression is obtained by observing Figure 6:

$$\begin{aligned} \bar{h}_f &= 2(p_1 + p_2 + p_3 + p_4 + p_5 + p_6 + p_8) \\ &\quad + \left(2\frac{6}{8} + 3\frac{2}{8}\right)p_7 + \left(2\frac{7}{9} + 3\frac{2}{9}\right)p_9 \\ &\quad + \left(2\frac{8}{10} + 3\frac{2}{10}\right)(p_{10} + p_{11}) \\ &\Rightarrow \bar{h}_f = \frac{179,132}{87,901} + O(N^{-6.55}) \simeq 2.03788. \end{aligned}$$

Then

$$\bar{h} \leq E[\lfloor \log_2(k_a + 1) \rfloor + h_f] \leq E[\log_2(k_a + 1) + h_f].$$

Using Jensen's inequality one obtains

$$\bar{h} \leq \log_2(E[k_a + 1]) + \bar{h}_f. \quad (9)$$

But

$$E[k_a] = N - \bar{f}(N).$$

From Theorem 1 it is known that

$$\bar{f}(N) = (N + 1) \frac{514,927}{615,307}$$

and

$$E[k_a] + 1 = (N + 1) \frac{100,380}{615,307}.$$

Then

$$\begin{aligned} \bar{h} &\leq \log_2 \left((N + 1) \frac{100,380}{615,307} \right) + \frac{179,132}{87,901} \\ &\leq \log_2(N + 1) - 0.57795. \quad \square \end{aligned}$$

THEOREM 5. Let t_i be the probability that a tree of the fringe is of type i in an $(N + 1)$ -leaves tree. Then

$$t_i = \frac{p_i}{L_i} \frac{1}{\sum_j (p_j/L_j)}$$

where j ranges from 1 to the number of tree types.

Proof. Let S be the number of trees of the fringe. The number of leaves of type i is $t_i S L_i$ or, equivalently, $p_i(N + 1)$. Then

$$\frac{t_i L_i S}{p_i(N + 1)} = 1 \Rightarrow \frac{t_i}{p_i/L_i} = \frac{1}{\sum_j (p_j/L_j)}.$$

Thus the theorem follows. Table 2a gives the probability values that a tree of the fringe is of type i in an $(N + 1)$ -leaves tree for the case examined. \square

THEOREM 6. Let np_i be the probability that a node of the fringe belongs to a tree of type i in an $(N + 1)$ -leaves tree. Then

$$np_i = \frac{N_i p_i}{L_i} \frac{1}{\sum_j N_j (p_j/L_j)} \quad (10)$$

where j ranges from 1 to the number of tree types.

Proof. Similar to that of Theorem 5. Table 2b gives the probability values that a node of the fringe of a random 2-3 tree with lps is of type i , for the case examined. \square

THEOREM 7. Let 'space utilization', su , be the ratio of the total number of keys to the number of keys if all nodes were full. The expected value of space utilization of the fringe of a 2-3 tree with lps of height 2 computed to five decimal places is

$$\bar{su} = 0.70716.$$

TABLE 2. Random 2-3 trees with lps of height 2. (a) probability that a tree of the fringe is of type i , (b) probability that a node of the fringe belongs to a tree of type i .

(a)	(b)
$t_1 \cong 0.14812$	$np_1 \cong 0.12251$
$t_2 \cong 0.24937$	$np_2 \cong 0.20626$
$t_3 \cong 0.09157$	$np_3 \cong 0.07574$
$t_4 \cong 0.13019$	$np_4 \cong 0.14357$
$t_5 \cong 0.17259$	$np_5 \cong 0.19034$
$t_6 \cong 0.07671$	$np_6 \cong 0.08456$
$t_7 \cong 0.05753$	$np_7 \cong 0.07931$
$t_8 \cong 0.01534$	$np_8 \cong 0.01692$
$t_9 \cong 0.04603$	$np_9 \cong 0.06345$
$t_{10} \cong 0.00837$	$np_{10} \cong 0.01154$
$t_{11} \cong 0.00418$	$np_{11} \cong 0.00577$

Proof. The space utilization of a node of type i is $K_i/2N_i$. According to Theorem 6 the probability that a fringe node is of type i is

$$np_i = \frac{N_i p_i}{L_i} \frac{1}{\sum_j N_j (p_j/L_j)}.$$

The expected value of the space utilization is

$$\bar{su} = \sum_1^{11} np_i \frac{K_i}{2N_i} = \frac{C}{2} \left(1 - \sum_1^{11} \frac{p_i}{L_i} \right)$$

where

$$C = \frac{1}{\sum_1^{11} N_j (p_j/L_j)}.$$

Table 1 and Figure 6 lead to the theorem. \square

THEOREM 8. Let $\bar{n}(N)/(N/2)$ be the storage used by a 2-3 tree. Then

$$\begin{aligned} &1.34656 + 0.34656N^{-1} + O(N^{-6.55}) \\ &\leq \frac{\bar{n}(N)}{N/2} \leq 1.50970 - 0.49030N^{-1} + O(N^{-6.55}) \end{aligned}$$

where $N/2$ represents the number of nodes when all the nodes of the tree contain 2 keys exactly.

Proof. From Theorem 2 it follows that

$$\begin{aligned} &\frac{414,273}{615,307}N - \frac{213,239}{1,230,614} + O(N^{-6.55}) \\ &\leq \bar{n}(N) \leq \frac{464,463}{615,307}N - \frac{150,844}{615,307} + O(N^{-6.55}). \end{aligned}$$

Thus the theorem follows. \square

4.3. An alternative approach

In this section, the aim is a two-level analysis of 2-3 trees with lps, where double nodes may appear in the two lowest levels. The difficulties of such an analysis will be shown and bounds will be derived.

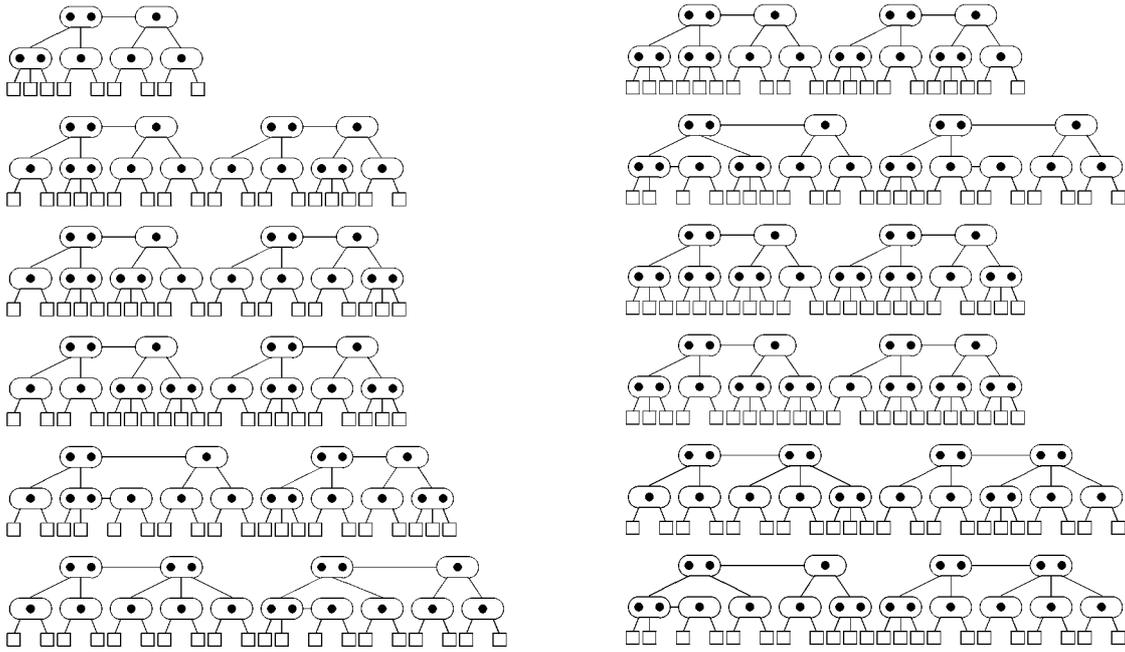


FIGURE 9. Collection of 2–3 trees of height 2 with lps and a double node as root.

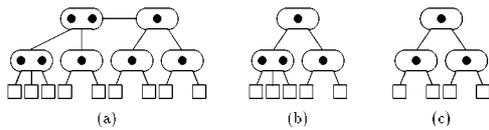


FIGURE 10. (a) A tree with a double node as root; (b) and (c) the two trees that will be produced just before a key insertion in (a).

The collection of trees consists of those presented in Figure 6 and Figure 9. The insertion of a new key into any of the trees of type 7, 9, 10 or 11 of Figure 6 will cause the creation of either two separate trees or one tree with a double node as root, depending on the parent node one level up. It is clear that the second necessary condition for a tree collection to be closed¹ is not satisfied. Therefore, a closed tree collection that is required by fringe analysis has to be achieved.

Here, we are going to transform the original tree collection to a closed one. It is known that a key insertion into any of the trees of Figure 9 will change the double root node to two separate nodes before the new key is inserted into the appropriate leaf. Figure 10 shows a tree with a double node as root, and the two separate trees that will be produced just before a key insertion. This first step before a key insertion allows us to break the double root node into two nodes and to use the two new trees instead of the original one to form a closed tree collection. The right trees are already members of the tree collection of Figure 6. The left

trees, shown in Figure 11, are new trees that will be added to the tree collection of Figure 6. The type these trees belong to is also shown in Figure 11. We choose to group the left part of the original tree in the same tree type with the separate tree produced by an insertion to the right part of the original tree. A key insertion in the right part of a tree of Figure 9 will cause the left part of the tree to be changed. The left part of the original tree and the separate left tree produced by the key insertion belong to the same tree type. Therefore, the change that occurs does not affect fringe analysis. Figure 12 shows a tree with a double node as root and the two trees that are used in fringe analysis instead of it. An insertion in the tree (c) of Figure 12 will cause the tree (b) of Figure 12 to be changed to the tree (b) of Figure 10.

The types of trees studied in this subsection are the same as those of the previous one, although more trees are included in types 1, 2, 3, 4 and 5. All the trees of the same tree type have the same number of leaves² and the same number of nodes, but may differ in the number of keys and/or in the mean path length. The transition matrix is that of Figure 7, whereas the probability values of Table 1 are also valid.

THEOREM 9. *The probability p_s that a tree is not a separate tree connected to a parent node one level up, but the right part of a tree whose root is a double node, is bounded by*

$$0 \leq p_s \leq 0.39592.$$

Proof. The probability that a tree of Figure 6 or Figure 11 belongs to a tree of Figure 9 is less than or equal to the sum of the probabilities of the tree types that compose these trees

¹As mentioned in Section 3, the effect of an insertion on the composition of the fringe of a random tree is determined only by the subtree of the fringe in which the insertion is performed.

²It is a necessary condition to apply fringe analysis.

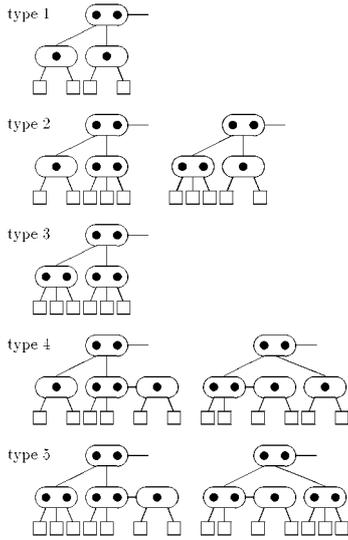


FIGURE 11. Left part of trees with double node as root.

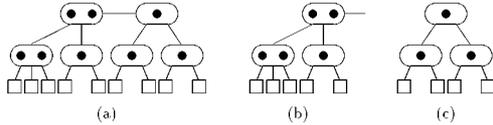


FIGURE 12. (a) a tree with a double node as root; (b) and (c) the two trees we study in fringe analysis instead of (a).

(types 1, 2, 3, 4 and 5). The probability of being a right or a left part is equal. According to Table 2a it holds that $p_s \leq 0.39592$. \square

THEOREM 10. *The expected number of keys in the fringe of a 2-3 tree with lps containing N keys is bounded by*

$$0.83686(N+1) \leq \bar{f}(N) \leq 0.90145(N+1).$$

Proof. The number of keys in the fringe is the number of keys if p_s is 0, plus one key per right part of the tree. According to Theorem 1, in the case that when $p_s = 0$, the number of keys in the fringe is $0.83686(N+1)$. The number of trees of the fringe when counting every tree with a double node root as two trees is

$$\sum_i A_i(N) = \sum_1^{11} \frac{(N+1)p_i}{L_i} \simeq 0.16314(N+1).$$

The number of trees that are a right part of the tree is $0.16314(N+1)p_s$. The number of keys in the fringe is

$$\bar{f}(N) = 0.83686(N+1) + 0.16314(N+1)p_s. \quad (11)$$

From Theorem 9 one finds that

$$0.83686(N+1) \leq \bar{f}(N) \leq 0.90145(N+1). \quad \square$$

THEOREM 11. *The expected number of 'equivalent single nodes' in a random 2-3 tree with lps containing N keys is bounded by*

$$0.64099N + 0.14099 \leq \bar{n}(N) \leq 0.75485N - 0.24515.$$

Proof. The proof is similar to that of Theorem 2. Let n_f and n_a be the number of equivalent single nodes in the fringe and the number of nodes above the fringe, respectively. Then

$$\bar{n}(N) = \bar{n}_f + \bar{n}_a,$$

$$\frac{n_l - 1}{2} \leq n_a \leq n_l - 1,$$

where n_l is the number of leaves of the tree above the fringe (or, equivalently, the number of the trees of the fringe). It holds that

$$\bar{n}_l = (1 - p_s) \sum_1^{11} \frac{(N+1)p_i}{L_i},$$

$$\bar{n}_f = \sum_1^{11} \frac{(N+1)p_i N_i}{L_i}.$$

Then

$$\frac{1 - p_s}{2} \sum_1^{11} \frac{(N+1)p_i}{L_i} + \sum_1^{11} \frac{(N+1)p_i N_i}{L_i} - \frac{1}{2} \leq \bar{n}(N)$$

and

$$\bar{n}(N) \leq (1 - p_s) \sum_1^{11} \frac{(N+1)p_i}{L_i} + \sum_1^{11} \frac{(N+1)p_i N_i}{L_i} - 1.$$

These inequalities, Table 1, Figure 6 and Theorem 9 lead to Theorem 11. \square

THEOREM 12. *The storage $\bar{n}(N)/(N/2)$ used by a 2-3 tree is bounded by*

$$1.28198 + 0.28198N^{-1} + O(N^{-6.55})$$

$$\leq \frac{\bar{n}(N)}{N/2} \leq 1.50970 - 0.49030N^{-1} + O(N^{-6.55}).$$

Proof. By dividing the inequalities of Theorem 11 by $N/2$, the above inequalities are obtained. \square

THEOREM 13. *The expected value of space utilization su of the fringe of a 2-3 tree with lps of height 2 is bounded by*

$$0.70716 \leq \bar{su} \leq 0.81632.$$

Proof. The space utilization of a node of type i (where $6 \leq i \leq 11$) is $K_i/2N_i$. The space utilization of a node of type i (where $1 \leq i \leq 5$) is greater than or equal to the space utilization computed for a node of the trees of type i of Figure 6 and less than or equal to the space utilization computed for the nodes of the trees of type i of Figure 11. According to Theorem 6 the probability that a fringe node is of type i is

$$np_i = \frac{N_i p_i}{L_i} \frac{1}{\sum_j N_j (p_j/L_j)}.$$

Let u_i be the space utilization of a node of tree of type i . The expected value of the space utilization of the fringe is

$$\bar{su} = \sum_1^{11} np_i u_i.$$

Table 1, Figure 6 and Figure 11 lead to the theorem. \square

THEOREM 14. *The expected path length of a 2–3 tree with lps to the two lowest levels, containing N keys, is bounded by*

$$\bar{h}(N) \leq \log_2(N+1) - 0.57795.$$

Proof. The proof is similar to that of Theorem 4. The notation is also the same. To estimate the height of the tree above the fringe it is assumed that the lps technique is restricted to the two lowest levels. Let t_{li} indicate the probability that a tree of the fringe is a left sibling tree of type i , and t_{ri} indicate the probability that a tree of the fringe is a right sibling tree of type i . Let p_{li} be the probability that a leaf belongs to a left sibling tree of type i , and p_{ri} be the probability that a leaf belongs to a right sibling tree. If p_r indicates the probability that a leaf belongs to a right sibling tree then $p_r = p_{r1} + p_{r2} + p_{r3} + p_{r4} + p_{r5}$. From Theorem 9, p_s indicates the probability that a tree of the fringe is a right sibling tree. This is equal to the probability that a tree of the fringe is a left sibling tree.

The expected path length \bar{h}_f of the fringe is computed by the path length of the trees of the fringe increased by the probability that a leaf belongs to the right part of a tree with a double node as root. The following expression is obtained by observing Figure 6 and Figure 11:

$$\begin{aligned} \bar{h}_f &= 2(p_1 + p_2 + p_3 + (p_4 - p_{l4}) + (p_5 - p_{l5}) + p_6 + p_8) \\ &\quad + \left(2\frac{6}{8} + 3\frac{2}{8}\right)p_7 + \left(2\frac{7}{9} + 3\frac{2}{9}\right)p_9 \\ &\quad + \left(2\frac{8}{10} + 3\frac{2}{10}\right)(p_{10} + p_{11}) + \left(2\frac{4}{6} + 3\frac{2}{6}\right)p_{l4} \\ &\quad + \left(2\frac{5}{7} + 3\frac{2}{7}\right)p_{l5} + p_r \end{aligned}$$

or

$$\bar{h}_f \simeq 2.03788 + \frac{2}{6}p_{l4} + \frac{2}{7}p_{l5} + p_r. \quad (12)$$

In proportion to Theorem 5

$$t_{li} = \frac{p_{li}}{L_i} \frac{1}{\sum_{j=1}^{11} (p_j/L_j)}.$$

It is known that $\sum_{i=1}^5 t_{li} = p_s$. Therefore

$$\sum_{i=1}^5 \frac{p_{li}}{L_i} = p_s \sum_{j=1}^{11} \frac{p_j}{L_j}$$

then

$$\frac{p_{l4}}{6} + \frac{p_{l5}}{7} \leq p_s \sum_{j=1}^{11} \frac{p_j}{L_j}.$$

In the same way,

$$\sum_{i=1}^5 \frac{p_{ri}}{L_i} = p_s \sum_{j=1}^{11} \frac{p_j}{L_j}$$

whereas since $L_5 \geq L_i$ (where $1 \leq i \leq 5$) and

$$\sum_{i=1}^5 p_{ri} = p_r,$$

it follows that

$$p_r \leq L_5 \sum_{i=1}^5 \frac{p_{ri}}{L_i} = L_5 p_s \sum_{j=1}^{11} \frac{p_j}{L_j}.$$

One level above the fringe the number of full nodes is greater than or equal to the number of right sibling trees, which is $0.16314(N+1)p_s$. To calculate the maximum height h_a of the classic 2–3 tree above the fringe we use the number of keys k_a where

$$\begin{aligned} E[k_a] &= N - \bar{f}(N) - 0.16314(N+1)p_s \Rightarrow \\ E[k_a] + 1 &= 0.16314(N+1)(1 - 2p_s). \end{aligned}$$

Using Equation (11) and Inequality (9) one finds that

$$\bar{h} \leq \log_2(0.16314(N+1)(1 - 2p_s)) + 2.03788 + 1.46824p_s.$$

The sum $\log_2(1 - 2p_s) + 1.46824p_s$, where $0 \leq p_s \leq 0.39592$, is negative or 0. Then

$$\bar{h} \leq \log_2(N+1) - 0.57795.$$

5. CONCLUSIONS

In this paper, a second-order fringe analysis is presented for a 2–3 tree variation, which uses a lazy parent split technique to enhance concurrency; bounds on some performance metrics of 2–3 trees with lps have been achieved. Table 3 summarizes the derived expressions for a random 2–3 tree with lps restricted to the lowest level, and depicts a comparison of these performance metrics to those of the classical 2–3 tree, as derived in [11], where the results are approximated to $O(N^{-6.55})$ for $N \rightarrow \infty$. As easily extracted from the table, in the fringe of 2–3 trees with lps in comparison to the fringe of the classical 2–3 trees, the expected number of keys in the fringe is 2.3% larger, the upper and lower bounds of the expected number of nodes are 5% and 4% lower respectively, the storage utilization is 6% larger and the expected path length is slightly shorter. The results derived for a random 2–3 tree with lps restricted to the lowest level are bounds to the random 2–3 tree with lps as Table 4 presents.

In conclusion a 2–3 tree with lps has a better performance than a 2–3 tree with respect to space utilization. This, along with the improvement in concurrency, makes the 2–3 tree with lps a useful and noteworthy B-tree variant. Note also that since fringe analysis does not consider deletions, details on how B-trees with lps manage deletions have been omitted. However, during deletions clean-up takes place and, consequently, in practice the lps technique makes this B-tree variant even more compact (i.e. smaller height and fewer nodes).

The analysis presented above examines the two lowest levels of 2–3 trees with lps. An effort towards performing

TABLE 3. Comparison between 2–3 trees with lps restricted to the lowest level and 2–3 trees.

	2–3 tree with lps to the lowest level	2–3 tree
$\bar{f}(N)$	0.83686 ($N + 1$)	0.81792 ($N + 1$)
Pr{0 splits}	0.44310	0.57143
$\bar{h}(N)$	$\log_2(N + 1) - 0.57795$	$\log_2(N + 1) - 0.45736$
$\bar{s}u$ of the fringe	0.70716	0.66971
$\bar{n}(N)$	$[0.67328N + 0.17328, 0.75485N - 0.24515]$	$[0.70169N + 0.20169, 0.79273N - 0.20727]$
$\frac{\bar{n}(N)}{N/2}$	$[1.34656 + 0.34656N^{-1}, 1.50970 - 0.49030N^{-1}]$	$[1.40338 + 0.40338N^{-1}, 1.58546 - 0.41454N^{-1}]$

TABLE 4. Summary of second-order fringe analysis results for random 2–3 tree with lps.

	2–3 tree with lps
$\bar{f}(N)$	$[0.83686(N + 1), 0.90145(N + 1)]$
$\bar{h}(N)$	$<\log_2(N + 1) - 0.57795$
$\bar{s}u$ of the fringe	$[0.70716, 0.81632]$
$\bar{n}(N)$	$[0.64099N + 0.14099, 0.75485N - 0.24515]$
$\frac{\bar{n}(N)}{N/2}$	$[1.28198 + 0.28198N^{-1}, 1.50970 - 0.49030N^{-1}]$

a third-order analysis for 2–3 trees with lps has been made. The lps technique has been restricted to the two lowest levels to establish a closed tree collection and make the analysis feasible. The major problem in performing such an analysis is the large amount of tree types and the size of the corresponding transition matrix. Open, also, is the problem to perform a fringe analysis for B-trees with lps of degree d greater than 1. Finally, fringe analysis for other B-tree variants (for example, see [14, 15, 16]) might help in deriving globally concrete comparative results on the performances of B-trees.

REFERENCES

- [1] Biliris, A. (1989) Operation specific locking in balanced structures. *Inf. Sci.*, **48**, 27–51.
- [2] Fu, A. and Kameda, T. (1989) Concurrency control of nested transactions accessing B-trees. In *Proc. 8th ACM PODS Conf.*, Philadelphia, PA, March 29–31, pp. 270–285. ACM Press.
- [3] Johnson, T. and Shasha, D. (1993) The performance of concurrent B-tree algorithms. *ACM Trans. Database Syst.*, **18**, 51–101.
- [4] Mond, Y. and Raz, Y. (1985) Concurrency control in B+ trees databases using preparatory operations. In *Proc. 11th VLDB Conf.*, Stockholm, Sweden, August 21–23, pp. 331–334. Morgan Kaufmann.
- [5] Keller, A. M. and Wiederhold, G. (1988) Concurrent use of B-trees with variable length entries. *ACM SIGMOD Record*, **17**, 89–90.
- [6] Manolopoulos, Y. (1994) B-trees with lazy parent split. *Inf. Sci.*, **79**, 73–88.
- [7] Nurmi, O., Soisalon-Soininen, E. and Wood, D. (1987) Concurrency control in database structures with relaxed balance. In *Proc. 6th PODS Conf.*, San Diego, CA, March 24–26, pp. 170–176. ACM Press.
- [8] Sagiv, Y. (1985) Concurrent operations on B-trees with overtaking. In *Proc. 4th ACM PODS Conf.*, Portland, OR, March 25–27, pp. 28–37. ACM Press.
- [9] Baeza-Yates, R. A. (1995) Fringe analysis revisited. *ACM Computing Surveys*, **27**, 109–119.
- [10] Baeza-Yates, R. A. and Poblete, P. V. (1995) Higher-order analysis of 2–3 trees. *Int. J. Foundations Comp. Sci.*, **6**, 1–10.
- [11] Eisenbarth, B., Ziviani, N., Gonnet, G., Mehlhorn, K. and Wood, D. (1982) The theory of fringe analysis and its applications to 2–3 trees and B-trees. *Inf. Control*, **55**, 125–174.
- [12] Yao, A. C. C. (1978) On random 2–3 trees. *Acta Informatica*, **9**, 159–170.
- [13] Johnson, T. and Shasha, D. (1989) Utilization of B-trees with inserts, deletes and modifies. In *Proc. 8th ACM PODS Conf.*, Philadelphia, PA, March 29–31, pp. 235–246. ACM Press.
- [14] Matsliach, G. (1991) Using multi-bucket data leaves with overflow chains—performance analysis. *Inf. Sys.*, **16**, 497–508.
- [15] Prabhakar, T. V. and Sahasrabudde, H. V. (1984) Towards an optimal data-structure: CB-trees. In *Proc. 10th VLDB Conf.*, Singapore, August 27–31, pp. 235–244. Morgan Kaufmann.
- [16] Srinivasan, B. (1991) An adaptive overflow technique to defer splitting in b-trees. *Comp. J.*, **34**, 416–425.