

B-tree 型インデックスにおけるページ内使用率†

牧野 武 則††

エントリの削除と挿入、すなわちキイ値の更新が繰り返される場合の B-tree 型インデックスにおける平均ページ内使用率を、再帰式とモンテカルロ法によるシミュレーションにより解析を行い、エントリ削除時の三つのアルゴリズムについて比較する。

初期ロード時あるいは削除のない環境では、平均ページ内使用率は約 70% であるが、その後、エントリの削除・挿入が繰り返されると、ページ併合 (catenation) と平滑化 (underflow) を行わない場合では、ページ内使用率は平衡状態に向って減少し、40% を下まわることもありうる。

しかし、ページ併合を導入すると、初期ロード時の使用率よりもわずかに低い程度の平均ページ内使用率が確保される。さらに、ページ併合に加えて平滑化を導入すると、初期ロード時にはほぼ等しい平均ページ内使用率が維持される。ページ内使用率でみると、平滑化の導入の効果は大きくなく、高々 1% 改善するにすぎない。

1. ま え が き

B-tree¹⁾ は、よく知られた多重ページインデックスの一種であり、ルートページからリーフページへのパスの長さが一定に保たれることから、検索時間の最大が保障される利点がある。また、その構造がページング方式のメモリスシステムに適合していることから、VSAM²⁾をはじめ、いくつかのデータ管理システムに導入されている。正確には、VSAM 等 B-tree を実現している実際のシステムでは、すべてのキイエントリをリーフページセットに有する等、オリジナルな B-tree に変更を加えている。ここでは、VSAM のようなインデックスを B-tree 型インデックスと呼ぶことにする。

B-tree 型インデックスの性能を評価する上で、ページ内のエントリ数の分布や平均ページ内使用率を知ることが必要である。これらが明らかにされると、エントリの検索・挿入・削除に対する処理時間の解析が可能となる。

ページ内使用率の解析は、エントリのインデックスへの初期ロードあるいはエントリの削除がない環境に対応する連続挿入の場合と、エントリの削除と挿入が繰り返される場合に分けられる。連続挿入に対する解析は、Nakamura, Mizoguchi³⁾によって再帰式による詳細な解析が行われ、平均ページ内使用率が約 70% になることが報告されている。

一方、削除を含む解析は、ほとんど行われていない

が、キイ値の更新の頻度の高い環境では重要である。あるデータベースシステム^{4), 5)}では、2次キイインデックスとしても使用されるため、更新頻度の高いキイをエントリとする場合が考えられ、挿入だけでなく削除が行われたときのページ内使用率の解析が要請される。

B-tree の提唱者である Bayer¹⁾ は、エントリ削除時、ページ内使用率の低下を防ぐため、隣接ページ間でのページ併合 (catenation) と平滑化 (underflow) の二つのアルゴリズムを導入している。ページ併合により平均ページ内使用率は 50% 以上が保障され、さらに平滑化を行うことによりどのページもページ内使用率は 50% 以上が保障される。しかし、これらの処理を行うためには、ページ間での物理的なエントリの移送が必要であり、削除処理にとって負担となる。

この論文では、エントリの削除と挿入が繰り返される場合、すなわち、インデックス内のエントリ数は変化せず、キイ値の更新が繰り返される場合について解析を行う。この解析では、リーフページだけを対象とし、また、仮想メモリスシステムを想定し、物理的なメモリ容量に対する使用率は問題にしない。

始めに、エントリの挿入と削除に対するインデックスの振る舞いをモデル化し定式化する。つぎに、エントリの削除と挿入が繰り返される場合の再帰式を、削除アルゴリズムにおいて、ページ併合と平滑化を行わない場合とページ併合を行う場合について得る。ページ併合と平滑化を行う場合については、モンテカルロ法によるシミュレーションにより解析する。それらの結果により、三つの削除アルゴリズムについて、エントリの削除・挿入の繰り返しの増加に伴う平均ページ

† Intra-page Utilization in B-tree Type Indexes by TAKENORI MAKINO (Central Research Laboratories, Nippon Electric Co. Ltd.).

†† 日本電気(株)中央研究所

内使用率の変化と、平衡状態における平均ページ内使用率について検討する。

2. モデル

議論を簡単にするため、解析の対象はインデックス内の大半のページを占めるリーフページのみとし、上位インデックスページは考慮しない。さらに、どのページについても入りうるエントリの数は一定とし、その数を $2k$ (偶数) とする。

つぎに、挿入あるいは削除されるエントリのキー値は、それ以前に挿入・削除されたエントリのキー値と独立であるとする。このことは、ランダムに挿入・削除が行われる場合に対応する。

いま、 $P_{m,i}$ をインデックスに m 個のエントリが登録されているとき、あるエントリが i 個のエントリを有しているページ (i エントリページと呼ぶ) に属している確率とする。すなわち、

$$P_{m,i} = i \cdot n_i / m. \quad (1)$$

ここで、 n_i は i エントリページの数である。 i エントリページの分布を $\{P_{m,i}^*\}$ とすると、

$$P_{m,i}^* = (P_{m,i}/i) / \sum_{j=1}^{2k} P_{m,i}/j. \quad (2)$$

新たにエントリが挿入される時、そのキー値は既に登録されているエントリのキー値と独立であることから、どのエントリに対しても、そのエントリの隣りに挿入される確率は等しい。したがって、 $P_{m,i}$ の確率で i エントリページに挿入される。削除についても同様に $P_{m,i}$ の確率で、 i エントリページから削除される。

2.1 エントリの挿入

現在、 $(m-1)$ 個のエントリが登録されているとする。新たに挿入されるエントリは $P_{m-1,i}$ の確率で i エントリページに挿入される。 $i \neq 2k$ ならば、 i エントリページ数は $n_i - 1$ となり、 $(i+1)$ エントリページ数は $n_{i+1} + 1$ となる。一方、 $i = 2k$ 、すなわちページが既に満杯ならば、ページスプリットが起り、 $2k$ エントリページ数は $n_{2k} - 1$ となり、 k エントリページおよび $(k+1)$ エントリページ数は、それぞれ $n_k + 1$ 、 $n_{k+1} + 1$ となる。したがって、結果としての $P_{m,i}$ は次のように表わせる。 $i \in [1, 2k]$ について、

$$\left. \begin{aligned} P_{m,i} &= i(n_i + P_{m-1,i-1} - P_{m-1,i})/m, \\ &\quad \text{for } i \neq k, k+1, \\ P_{m,i} &= i(n_i + P_{m-1,i-1} - P_{m-1,i} + P_{m-1,2k})/m, \\ &\quad \text{for } i = k, k+1. \end{aligned} \right\} \quad (3)$$

(3)に、 $n_i = (m-1)P_{m-1,i}/i$ を代入し簡単にすると、

$$\left. \begin{aligned} mP_{m,i} &= (m-i-1)P_{m-1,i} + iP_{m-1,i-1}, \\ &\quad \text{for } i \neq k, k+1, \\ mP_{m,i} &= (m-i-1)P_{m-1,i} + i(P_{m-1,i-1} + P_{m,2k}), \\ &\quad \text{for } i = k, k+1. \end{aligned} \right\} \quad (4)$$

2.2 エントリの削除

現在、 m 個のエントリが登録されているとすると、新たに削除されるエントリは $P_{m,i}$ の確率で i エントリページに属している。

削除アルゴリズムとして、ページ併合と平滑化を行わない場合と、ページ併合だけを行う場合、ページ併合と平滑化を行う場合の三つのケースについて考える。

(a) ページ併合と平滑化を行わない場合

i エントリページから削除されると、 i エントリページ数は $n_i - 1$ となり、 $(i-1)$ エントリページ数は $n_{i-1} + 1$ となる。ただし、 $i=1$ のとき、1 エントリページ数は $n_1 - 1$ となるが、空になったページは仮想メモリシステムでは削除され、その実体 (物理領域) を失うため、ページ内使用率に含めないとする。

エントリ削除の結果としての $P_{m-1,i}$ は次のように表現できる。 $i \in [1, 2k]$ について、

$$\left. \begin{aligned} P_{m-1,i} &= i(n_i + P_{m,i+1} - P_{m,i})/(m-1), \\ &\quad \text{for } i \neq 2k, \\ P_{m-1,2k} &= i(n_i - P_{m,2k})/(m-1). \end{aligned} \right\} \quad (5)$$

(5)に、 $n_i = mP_{m,i}/i$ を代入して簡単にすると、

$$\left. \begin{aligned} (m-1)P_{m-1,i} &= (m-i)P_{m,i} + iP_{m,i+1}, \\ &\quad \text{for } i \neq 2k, \\ (m-1)P_{m-1,2k} &= (m-2k)P_{m,2k}. \end{aligned} \right\} \quad (6)$$

(b) ページ併合を行う場合

i エントリページからエントリが削除され、そのページの隣接ページが j エントリページであったとすると、 $i+j-1 \leq 2k$ ならば、この二つのページ間でページ併合が行われ、 i エントリページと j エントリページ数が $n_i - 1$ 、 $n_j - 1$ となるとともに、 $(i+j-1)$ エントリページ数が $n_{i+j-1} + 1$ となる。一方、 $i+j-1 > 2k$ ならば、(a)と同様、 i エントリページと $(i-1)$ エントリページ数が、 $n_i - 1$ 、 $n_{i-1} + 1$ となる。

ここで、二つの仮定を設ける。まず、隣接ページは2ページ存在するが、どちらか一方とする。キー値順でいえば大きい方 (あるいは小さい方) のページを隣接ページとする。つぎに、異なった上位インデックスレベルに從属する二つのページ間でのページ併合も許すとする。このページ併合は、上位インデックスの変

更を必要とするため、インデックスのレベル数が大きい場合、現実的ではない。しかし、 k がある程度大きければ、平均ページ内使用率への影響は小さいと考えられる。

この二つの仮定から、あるページの隣接ページが j エントリページである確率は、ページ分布においてページ間従属性がなければ、 $P_{m,j}^*$ とすることができる。

i エントリページの数が増えるのは次の場合である； i) j エントリページからエントリが削除され、隣接ページが $(i-j+1)$ エントリページであるとき、 ii) $(i+1)$ エントリページからエントリが削除され、隣接ページが $(2k-i+1)$ 個以上のエントリを有しているとき。一方、 i エントリページの数が増えるのは次の場合である； i) i エントリページからエントリが削除される時、 ii) j エントリページ、 $1 \leq j \leq 2k-i+1$ 、からエントリが削除され、その隣接ページが i エントリページであるとき。

したがって、エントリの削除の結果、 $P_{m-1,i}$ は次のように表現できる。 $i \in [1, 2k]$ について、

$$P_{m-1,i} = \left(n_i + \sum_{j=1}^{2k-i+1} P_{m,j} P_{m,i-j+1}^* + \sum_{j=2k-i+1}^{2k} P_{m,i+1} P_{m,j}^* - P_{m,i} - \sum_{j=1}^{2k-i+1} P_{m,j} P_{m,i}^* \right) / (m-1). \quad (7)$$

(7) に $n_i = m P_{m,i} / i$ を代入して、

$$(m-1) P_{m-1,i} = (m-i) P_{m,i} + i \left[\sum_{j=1}^{2k-i+1} P_{m,j} P_{m,i-j+1}^* + P_{m,i+1} \sum_{j=2k-i+1}^{2k} P_{m,j}^* - P_{m,i}^* \sum_{j=1}^{2k-i+1} P_{m,j} \right]. \quad (8)$$

ただし、 $P_{m,2k+1} = 0$ である。

(c) ページ併合と平滑化を行う場合

i エントリページからエントリが削除され、そのページの隣接ページが j エントリページであるとする。 $i+j-1 \leq 2k$ ならばページ併合が行われ、 $i+j-1 > 2k$ ならば、この二つのページの間で平滑化が行われる。そして、 i エントリページ数は n_i-1 、 j エントリページ数は n_j-1 となるとともに、 $\lfloor (i+j-1)/2 \rfloor$ エントリページと $\lfloor (i+j)/2 \rfloor$ エントリページの数それぞれ $n_{\lfloor (i+j-1)/2 \rfloor} + 1$ 、 $n_{\lfloor (i+j)/2 \rfloor} + 1$ となる。

ここで、 $\lfloor x \rfloor$ は x 以下でかつ最大の整数である。

i エントリページの隣接ページが j エントリページである確率を $P_{m,j}^*$ とすれば、(b)と同様にして再帰式を得ることができるが、平滑化を行うと、このことは成り立たず、 i エントリページの隣接ページは、 i 個あるいは i 個に近いエントリを有する確率が高くなる。この様子を解析的にモデル化することは困難であり、モンテカルロ法によるシミュレーションにより解析される。

3. 平均ページ内使用率

i エントリページの分布が $\{P_{m,i}^*\}$ のとき、ページに入っている平均のエントリ数は、

$$\sum_{i=1}^{2k} i P_{m,i}^*$$

であり、平均ページ内使用率を $\rho_{m,2k}$ とすると、

$$\rho_{m,2k} = \sum_{i=1}^{2k} i P_{m,i}^* / 2k. \quad (9)$$

ここでは、エントリ削除の三つのアルゴリズムにおける平均ページ内使用率を比較するために、まず、 m 個のエントリが連続挿入された後、 n 回のエントリの削除と更新の繰り返し、すなわち、ランダムに n 個のエントリのキー値の更新が行われる状況を想定する。

3.1 エントリの連続挿入

(4)において、初期分布を $P_{2k,2k}=1$ 、 $P_{2k,i}=0$ 、 $i \in [1, 2k-1]$ とし、 $(m-2k)$ 回反復演算を行うことにより、 $\{P_{m,i}\}$ を得ることができる。この $\{P_{m,i}\}$ を引き続いて行われるエントリの削除・挿入に対する初期分布、 $\{P_{m,i}^{(0)}\}$ とする。

エントリの連続挿入に対する解析は、Nakamura³⁾によって詳細に解析されており、十分に大きな m について平均ページ内使用率が約 70% になること、さらに十分に大きな k について、 $ln 2$ となることが報告されている。その解析で導入されたページ数を確率変数とする再帰式と、(4)は本質的に同じであり、 $P_{m,i}$ を n_i で置き換えれば同じ形になる。

3.2 エントリの削除・挿入の繰り返し

あるエントリのキー値が更新されると、まず旧キー値をもつエントリを削除し、次に新キー値をもつエントリが挿入される。この削除・挿入が n 回行われたときの分布 $\{P_{m,i}\}$ を $\{P_{m,i}^{(n)}\}$ と表わし、 $\{P_{m,i}^*\}$ を $\{P_{m,i}^{*(n)}\}$ と表わす。 $\{P_{m,i}^{(0)}\}$ は初期分布とする。

削除・挿入の繰り返し、 n の増加とともに $\{P_{m,i}^{(n)}\}$ が収束するとすれば、 $P_{m,i}^{(n)}$ は確率的平衡状態をも

つ. 平衡状態では, 各 $P_{m,i}^{(n)}$ におけるページ数の遷移が釣り合っており, 十分に大きな n について,

$$P_{m,i}^{(n+1)} = P_{m,i}^{(n)}$$

が成り立つ.

(a) ページ併合と平滑化を行わない場合

$(n-1)$ 回目の削除・挿入の後, 新たに削除・挿入が行われたとすると, $\{P_{m,i}^{(n)}\}$ は (4) に (6) を代入して, 次の形で表わせる. $i \in [1, 2k]$ について,

$$\left. \begin{aligned} & m(m-1)P_{m,i}^{(n)} \\ &= \{(m-i-1)(m-i) + i(i-1)\} P_{m,i}^{(n-1)} \\ & \quad + i(m-i+1)P_{m,i-1}^{(n-1)} \\ & \quad + i(m-i-1)P_{m,i+1}^{(n-1)}, \\ & \quad \text{for } i \neq k, k+1, \\ & m(m-1)P_{m,i}^{(n)} \\ &= \{(m-i-1)(m-i) + i(i-1)\} P_{m,i}^{(n-1)} \\ & \quad + i(m-i+1)P_{m,i-1}^{(n-1)} \\ & \quad + i(m-i-1)P_{m,i+1}^{(n-1)} \\ & \quad + i(m-2k)P_{m,2i}^{(n-1)}, \text{ for } i=k, k+1. \end{aligned} \right\} \quad (10)$$

この $\{P_{m,i}^{(n)}\}$ が n の増加とともに収束することは, (10) の反復演算による数値解析を行って確かめることにして, 平衡状態における $\{P_{m,i}^{(n)}\}$ を求める. 平衡状態では, 次式が成り立つ.

$$P_{m,i}^{(n)} = P_{m,i}^{(n-1)} \equiv P_{m,i}$$

(10) に上式を代入すると次式が得られる. $i \in [1, 2k]$ について,

$$\left. \begin{aligned} & 2(m-i)P_{m,i} - (m-i-1)P_{m,i+1} \\ & \quad - (m-i+1)P_{m,i-1} = 0, \\ & \quad \text{for } i \neq k, k+1, \\ & 2(m-i)P_{m,i} - (m-i-1)P_{m,i+1} \\ & \quad - (m-i+1)P_{m,i-1} \\ & \quad - (m-2k)P_{m,2i} = 0, \\ & \quad \text{for } i=k, k+1. \end{aligned} \right\} \quad (11)$$

(11) は線型同次式であり, ある定数 c の導入により解くことができ, 次のように解が得られる.

$$\left. \begin{aligned} & P_{m,i} = c(2k-i+1)/(m-i), \\ & \quad \text{for } i \in [k+1, k], \\ & P_{m,i} = ci/(m-i), \text{ for } i \in [1, k]. \end{aligned} \right\} \quad (12)$$

定数 c は, $\sum_{i=1}^{2k} P_{m,i} = 1$ から求められる.

i エントリページの分布, $\{P_{m,i}^*\}$ は, (2) と (12) から次のように書ける.

$$\left. \begin{aligned} & P_{m,i}^* = c'(2k-i+1)/\{i(m-i)\}, \\ & \quad \text{for } i \in [k+1, 2k], \\ & P_{m,i}^* = c'/(m-i), \text{ for } i \in [1, k]. \end{aligned} \right\} \quad (13)$$

表 1 ページ併合と平滑化を行わない場合の平衡状態でのページ内使用率

Table 1 Intra-page utilization at equilibrium state for no catenation and no underflow.

#entries /page, $2k$	#entries (m)					
	256	512	1024	2048	...	$m \gg k$
4	0.5151	0.5147	0.5145	0.5144		0.5143
8	0.4395	0.4386	0.4382	0.4380		0.4378
16	0.4027	0.4010	0.4002	0.3997		0.3993
32	0.3869	0.3834	0.3817	0.3808		0.3800
64	0.3848	0.3771	0.3736	0.3720		0.3703
128	0.3989	0.3798	0.3722	0.3688		0.3655

表 2 繰り返し数, n , に対するページ内使用率
Table 2 Intra-page utilization to reiterations, n , for no catenation and no underflow; $2k=16$.

n/m	0.1	1.0	10.0
$m=10^2$	0.6631	0.5413	0.4056
$m=10^3$	0.6618	0.5350	0.4006
$m=10^4$	0.6618	0.5344	0.3998

ここで, 定数 c' は, $\sum_{i=1}^{2k} P_{m,i}^* = 1$ から求められる. したがって, 平衡状態における平均ページ内使用率, $\rho_{m,2k}$ は (13) と (9) から求められる.

表 1 は, 平衡状態における平均ページ内使用率, $\rho_{m,2k}$ を示す. しかし, まだ, $P_{m,i}^{(n)}$ が収束することを述べていない.

n の増加に伴うページ内使用率の変化を調べるため, $\{P_{m,i}^{(0)}\}$ を (4) の反復演算で求め, さらに (10) の反復演算を行うことにより $\{P_{m,i}^{(n)}\}$ を求めた.

表 2 は, $2k=16$ の場合について, 繰り返し数とエントリ数に対する平均ページ内使用率を示す. 表から, 繰り返し数とエントリ数の比に対するページ内使用率は, エントリ数にほとんど依存しない.

図 1 は, エントリ数が 1024 のときの, 繰り返し数とエントリ数の比に対するページ内使用率を示す. ペ

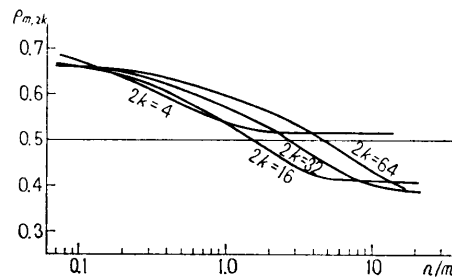


図 1 ページ併合, 平滑化をともに行わない場合の繰り返し数, n , に対する平均ページ内使用率
Fig. 1 Expected intra-page utilization to n/m for no catenation and no underflow; $m=1024$.

ージに入りうるエントリ数, $2k$, のいずれの場合にも, $2k$ に依存するある平衡状態に, 平均ページ内使用率が収束する様子が示されている.

充分に大きな n について反復演算で求められた $m=1024$ に対するページ内使用率は次のようになっている; $2k=4$ のとき $n=5000$ で 0.5146 , $2k=8$ のとき $n=5000$ で 0.4384 , $2k=16$ のとき $n=10000$ で 0.4003 . これらの値は, (13) を使用して求めた表 1 の第 3 カラム目の値とほぼ等くなる. したがって, (10) は n の増加とともに収束し, (12) で与えられた平衡状態に漸近するとすることができる.

削除時のアルゴリズムにページ併合と平滑化を行わない場合, ページに入りうるエントリ数が 16 以上のとき, 削除・挿入が繰り返されると平均ページ内使用率が 40% 以下になりうる. しかし平衡状態は存在し, 無限に平均ページ内使用率が低くなるわけではない. また, 図 1 に示すように, 平均ページ内使用率が 50% 以下になるのは, $2k=16$ で $n/m \cong 1.3$, $2k=32$ で $n/m \cong 2.5$, $2k=64$ で $n/m \cong 4.0$ となっており, 比較的更新頻度の少ないキをエントリとするインデックスでは削除処理の単純化・高速化のため, ページ併合と平滑化を行わない削除アルゴリズムも有効であろう.

(b) ページ併合を行う場合

エントリ挿入時, ページスプリットが行われることがあり, ページ分布においてページ間従属性が生じるため, (8) は厳密には成り立たないが, ページ間独立性が保障されるものとし, 近似解を求める.

$(n-1)$ 回の削除・挿入が行われた後, n 回目の削除が行われると, $\{P_{m-1,i}\}$ は, (8) から次のように書ける. $i \in [1, 2k]$ について,

$$\begin{aligned} & (m-1)P_{m-1,i}^{(n)} \\ &= (m-i)P_{m,i}^{(n-1)} \\ &+ i \left[\sum_{j=1}^{2k-j+1} P_{m,j}^{(n-1)} P_{m,i-j+1}^{*(n-1)} \right]. \end{aligned}$$

表 3 ページ併合を行う場合のページ内使用率
Table 3 Intra-page utilization for catenation and no underflow obtained by numerical analysis.

#entry /page	#entry	initial util.	#reiteration/#entry (n/m)						
			0.125	0.25	0.50	1.00	1.50	2.00	
4	64	0.7291	0.7312	0.7321	0.7325	0.7326	0.7326	0.7326	
8	128	0.7138	0.7026	0.7007	0.6999	0.6998	0.6998	0.6998	
16	256	0.7043	0.6919	0.6933	0.6954	0.6960	0.6961	0.6961	
32	512	0.6983	0.6877	0.6899	0.6915	0.6919	0.6922	0.6923	

$$\begin{aligned} & + P_{m,i+1}^{(n-1)} \sum_{j=2k-i+1}^{2k} P_{m,j}^{*(n-1)} \\ & - P_{m,i}^{*(n-1)} \sum_{j=1}^{2k-i+1} P_{m,j}^{(n-1)} \end{aligned} \quad (14)$$

次に, n 回目の削除に続く n 回目の挿入が行われると, (4) から次の形に書ける. $i \in [1, 2k]$ について,

$$\left. \begin{aligned} mP_{m,i}^{(n)} &= (m-i-1)P_{m-1,i}^{(n)} + iP_{m-1,i-1}^{(n)}, \\ & \text{for } i \neq k, k+1 \\ mP_{m,i}^{(n)} &= (m-i-1)P_{m-1,i}^{(n)} \\ &+ i(P_{m-1,i-1}^{(n)} + P_{m-1,2k}^{(n)}), \\ & \text{for } i = k, k+1 \end{aligned} \right\} \quad (15)$$

(14) を (15) に代入すれば削除・挿入の繰り返しに対する再帰式が得られ, 平衡状態を仮定すれば (11) 式に対応する同次式が得られるが, その同次式は連立 2 次同次式であり, 直接解を求めるのは困難である.

ここでは, まず (14) から $\{P_{m-1,i}^{(n)}\}$ を求め, つぎに, (15) から $\{P_{m,i}^{(n)}\}$ を求める反復演算で数値解析を行う. なお, $\{P_{m,i}^{*(n)}\}$ は (2) と同様にして $\{P_{m,i}^{(n)}\}$ から得る.

表 3 は, エントリ数が $32 \times k$ のときの, ページに入りうるエントリ数に対する平均ページ内使用率の変化を示す. 平均ページ内使用率は, 初期状態 ($n=0$) から n の増加とともに, 最初わずかに低下するが, やがて初期のページ内使用率よりも高々 1% 低い使用率に漸近する. 表 3 において, $2k=16$, $m=256$ のときのモンテカルロ法によるシミュレーションの結果は, $n/m=0.125$ で 0.6902 , $n/m=0.25$ で 0.6940 , $n/m=0.5$ で 0.6956 , $n/m=1.0$ で 0.6958 となっており, (14), (15) による解析はよい近似値を与える.

図 2 は, 削除アルゴリズムでページ併合と平滑化を行わない場合と, ページ併合を行う場合について, ページ内エントリ数の分布, $\{P_{m,i}^{*(n)}\}$ を示す. 図 2 で

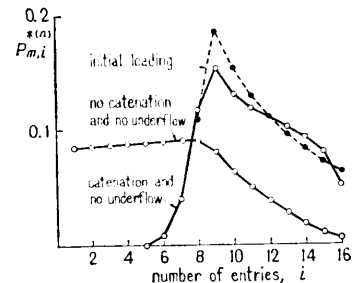


図 2 ページ内エントリ数の分布
Fig. 2 Distribution of entries in a page for reiterative entry deletion and insertion; $m=1024$, $n=1024$, $2k=16$.

破線で示している $\{P_{m,i}^{*(0)}\}$ は初期ロード，連続挿入に対する分布である。

ページ併合と平滑化を行わない場合に比べ，ページ併合を行う場合には，初期ロードにおけるページ内エントリ数の分布に近い分布を示している。平均ページ内使用率がページ併合を行う場合，初期ロード時の使用率にかなり近くなるのはこのためである。なお，ページ併合を行う場合，使用率 50% 以下のページは全体の 4.6% であり，平均ページ内使用率に対する寄与は小さい。

(c) ページ併合と平滑化を行う場合

平滑化を含む場合，再帰式を得るのは困難なため，モンテカルロ法によるシミュレーションにより解析する。この解析においても (b) と同様上位インデックスページの存在は無視する。

いま， m 個のエントリが h 個のページに登録されており， i 番目のページに入っているエントリ数を e_i とする。新たに挿入・削除されるエントリは，一様乱数 $r(0 < r < 1)$ を使用して， $\sum_{j=0}^{i-1} e_j < \lfloor m \cdot r \rfloor + 1 \leq \sum_{j=1}^i e_j$ を満足する i 番目のページへ挿入，あるいは，から削除される。これは (1) のランダム挿入・削除に対応する。

挿入の場合， $e_i \equiv 2k$ ならば， $e_i \leftarrow e_i + 1$ 。一方， $e_i = 2k$ ならば， j が $i+1$ から h について $e_{j+1} \leftarrow e_j$ とし， $e_i \leftarrow k$ ， $e_{i+1} \leftarrow k + 1$ ，そして $h \leftarrow h + 1$ とする。

削除の場合， $e_i + e_{i+1} - 1 \leq 2k$ ならば， $e_i \leftarrow e_i + e_{i+1} - 1$ ， j が $i+1$ から $h-1$ について $e_j \leftarrow e_{j+1}$ ，そして $h \leftarrow h - 1$ とする。一方， $e_i + e_{i+1} - 1 > 2k$ ならば， $e_i \leftarrow \lfloor (e_i + e_{i+1} - 1) / 2 \rfloor$ ， $e_{i+1} \leftarrow \lfloor (e_i + e_{i+1}) / 2 \rfloor$ とする。

シミュレーションは，まず， $e_1 = 2k$ ， $h = 1$ を与え， $(m - 2k)$ 回の挿入を行い $\{e_i\}$ の初期値を得て，その後，削除と挿入を繰り返すことで行われる。平均ページ内使用率は， $m / (h \cdot 2k)$ から計算される。

表 4 はシミュレーションにより得られた平均ページ内使用率を示す。シミュレーションの試行回数は計算

表 4 ページ併合と平滑化を行う場合のページ内使用率
Table 4 Intra-page utilization for catenation and underflow obtained by Monte Carlo simulation.

#entry/page	#entry	#reiteration/#entry (n/m)					
		0.125	0.25	0.50	1.00	1.50	2.00
4	64	0.7282	0.7346	0.7379	0.7384	0.7388	0.7386
8	128	0.7030	0.7085	0.7096	0.7097	0.7109	0.7091
16	256	0.6911	0.6944	0.6961	0.6986	0.6997	0.6996
32	512	0.6900	0.6913	0.6925	0.6928	0.6932	0.6930

に要する時間の都合と，(b)のシミュレーション結果から 10^4 とした。

平均ページ内使用率は， n の増加とともに，一旦減少の傾向がみられるが，ほぼ初期ロード時の平均ページ内使用率に漸近する。したがって，削除アルゴリズムにページ併合と平滑化を導入することで，初期ロード時のページ内使用率を維持することができる。しかし，ページ併合のみを行う場合に比べ，さらに大きな改善はみられず，高々 1% 程度使用率が向上するにすぎない。このことは，図 2 でみられるように，ページ併合により使用率 50% 以下のページがすでに十分に少なくなっていることによると考えられる。

4. ま と め

B-tree 型インデックスにおける，エントリの削除・挿入が繰り返される場合のページ内使用率を解析した。そして，エントリの削除時に，ページ併合と平滑化を行わない場合とページ併合だけを行う場合，ページ併合と平滑化を行う場合の三つのアルゴリズムについて比較した。

ページ併合と平滑化を行わない場合，エントリの削除・挿入が繰り返されると，平均ページ内使用率は 40% 以下になることがあるが，ページ併合を行うと平均ページ内使用率は，初期ロード時の使用率よりもわずかに低い程度となり，さらに平滑化を行うと約 70% となる。削除アルゴリズムにページ併合と平滑化を入れることで，連続挿入時の平均ページ内使用率を維持することができる。

ページ併合を行う効果は充分に大きいですが，さらに平滑化を行う効果は，平均ページ内使用率でみると，かなり小さく，高々 1% 向上するにすぎない。

一方，ページ併合も平滑化も行わない場合，平均ページ内使用率は無限に低くなるわけではなく，ある平衡状態に収束する。また，インデックス内のエントリが平均 1 回ずつキー値が更新されても平均ページ内使用率は 50% 以下にはならない。

これらの結果は，B-tree 型インデックスを設計する上で，また，その性能予測を行う上で有用であろう。

最後に，この論文をまとめる上で協力・助言を頂いた当研究所，コンピュータシステム研究部，箱崎勝也主任，周辺機器研究部，津雲淳氏に謝意を表わします。

参 考 文 献

1) Bayer, R., McCreight, E.: Organization and

- maintenance of large ordered indexes, Acta Informat., 1, 3, pp. 173-189 (1972).
- 2) Keehn, D., et al.: VSAM data set design parameters, IBM Syst. J., Vol. 13, No. 3, pp. 184-213 (1974).
 - 3) Nakamura, T., and Mizoguchi, T.: Analysis of storage utilization factor in block split data structuring scheme, Proc. 4th Int. Conference on Very Large Data Bases, pp. 489-495 (1978).
 - 4) Astrahan, M. M., et al.: System-R: Relational approach to data base management, ACM trans. Database Systems, Vol. 1, No. 2, pp. 97-137 (1976).
 - 5) Hakoziaki, K., et al.: A conceptual design of a generalized database subsystem, Proc. 3rd Int. Conference on Very Large Data Bases, pp. 246-253 (1977).

(昭和53年9月19日受付)

(昭和54年3月19日採録)
