Short Communications
Management Applications

Approximating Block Accesses in Database Organizations

S.B. Yao
Purdue University

Key Words and Phrases: database, inverted file organization, database performance and measurement, information retrieval, query answering

CR Categories: 3.70, 3.72, 4.33

When data records are grouped into blocks in secondary storage, it is frequently necessary to estimate the number of blocks X_D accessed for a given query. In a recent paper [1], Cardenas gave the expression

$$X_D = m(1 - (1 - 1/m)^k), \tag{1}$$

assuming that there are n records divided into m blocks and that the k records satisfying the query are distributed uniformly among the m blocks. The derivation of the expression was left to the reader as an exercise.

Let us take a closer look at the expression. $(1 - 1/m)$ gives the probability that a particular block does not contain a particular record. If k records are selected independently, then the probability that a particular block not being "hit" is given by $(1 - 1/m)^k$. Hence $1 - (1 - 1/m)^k$ gives the probability that a particular block is "hit," and the expression follows.

The assumption that the k records are selected independently implies selection with replacement. Since a record may be selected more than once, the k records may not be distinct. This is not valid in the case of a query access which retrieves all k distinct records at one time. In fact, Rothnie and Lozano showed that the result of eq. (1) gives the lower bound of the expected number of blocks accessed [2]. A more accurate analysis based on selection without replacement was given by Severance, but the precision problem makes the expression obtained computationally intractable (Appendix D in [3]). A similar approach by Siler results in a rather complicated recursive formula which can be computed (Appendix B in [4]). Another recursive formula was given by [3]. Using a different approach, a simple closed form was obtained by Yao in a different context [5]. The resulting expression was used in several applications [5, 6, 7] to estimate the expected number of data blocks accessed. Comparing this to the Cardenas approximation, it is shown that this refinement is significant when the blocking factor n/m is small. For large blocking factors (e.g. $n/m \geq 10$), the error involved in Cardenas' approximation is practically negligible.

THEOREM (Yao). *Given n records grouped into m blocks $(1 < m \leq n)$, each contains n/m records. If k records $(k \leq n - n/m)$ are randomly selected from the n records, the expected number of blocks hit (blocks with at least one record selected) is given by*

$$m \cdot \left[1 - \prod_{i=1}^{k} \frac{nd - i + 1}{n - i + 1}\right] \text{ where } d = 1 - 1/m. \tag{2}$$

PROOF. Let X be a random variable representing the number of blocks hit and let I_j be a random variable where $I_j = 1$ when at least one record in the jth block is selected, and $I_j = 0$ otherwise.

Fig. 1. Comparison of expected block accesses.

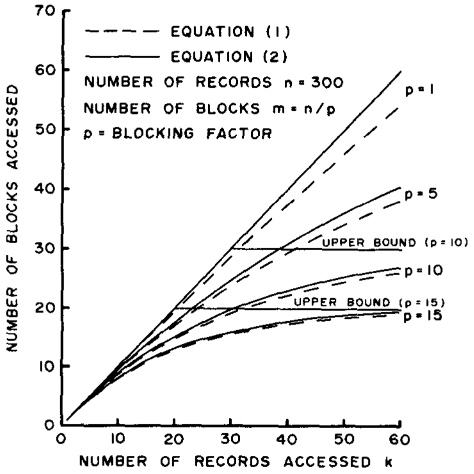
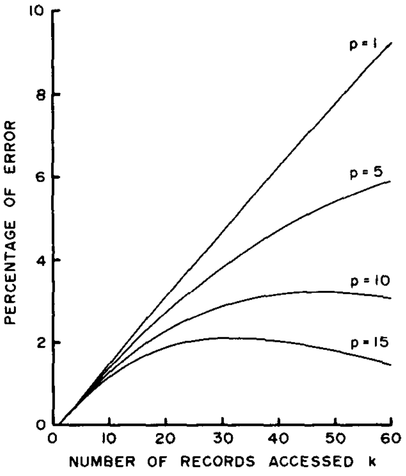


Fig. 2. Errors of eq. (1) approximation.



Author's address: Department of Computer Sciences, Purdue University, West Lafayette, IN 47907.

The *j*th block has $p = n/m$ records and there are $n - p$ records not in the *j*th block. The probability that no records are selected from the *j*th block is

$$C_k^{n-p}/C_k^n \text{ or } C_k^{nd}/C_k^n, \text{ where } d = 1 - \frac{1}{m}.$$

It follows that the expectation of I_j is

$$E(I_j) = 1 - C_k^{nd}/C_k^n.$$

Hence the expected number of blocks hit is

$$\begin{aligned} E(X) &= \sum_{j=1}^{m} E(I_j) \\ &= m \cdot (1 - C_k^{nd}/C_k^n). \end{aligned}$$

Using the identity $C_y^x = \frac{x!}{y!(x-y)!}$, we have

$$\begin{aligned} E(X) &= m \cdot \left[1 - \frac{(nd)!\,(n-k)!}{n!\,(nd-k)!}\right] \\ &= m \cdot \left[1 - \prod_{i=1}^{k} \frac{nd - i + 1}{n - i + 1}\right]. \end{aligned} \qquad \square$$

The following corollary of the theorem is obvious.

COROLLARY. *If $k > n - n/m$ or $m = 1$, then all m blocks are hit.*

It is interesting to note that eq. (1) is independent of the number of records n. We observe that the approximation of eq. (1) is good if $k \ll n$ and $m \ll n$. Intuitively, if n is large and each block contains many records, the selection with and without replacement makes little difference, especially when a small number of records are selected. In many actual applications it is true that $k \ll n$, otherwise sequential processing of the records can be considered. Using Siler's example, the results from Cardenas, Siler, and Yao are compared in Figure 1. The Siler result for 300 records grouped into 20 blocks with a blocking factor of 15 matches Yao's result. The errors of Cardenas' expression are plotted in Figure 2.

Received January 1976; revised April 1976

References

1. Cardenas, A.F. Analysis and performance of inverted data base structures. *Comm. ACM 18*, 5 (May 1975), 253–263.
2. Rothnie, J.B., and Lozano, T. Attribute based file organization in a paged memory environment. *Comm. ACM 17*, 2 (Feb. 1974), 63–69.
3. Severance, D.G. Some generalized modeling structures for use in design of file organizations. Ph.D. Diss., U. of Michigan, Ann Arbor, Mich. 1972.
4. Siler, K.F. A stochastic evaluation model for data base organizations in data retrieval systems. *Comm. ACM 19*, 2 (Feb. 1976), 84–95.
5. Yao, S.B. Tree structures construction using key densities. Proc. ACM 1975 Nat. Conf., pp. 337–340.
6. Yao, S.B. Evaluation and optimization of file organizations through analytic modeling. Ph.D. Diss., U. of Michigan, Ann Arbor, Mich. 1974.
7. Yao, S.B. An attribute based model for database access cost analysis. *ACM Trans. Database Syst. 2*, 1 (March 1977) 45–67.
8. Yue, P.C., and Wong, C.K. Storage Cost Considerations in Secondary Index Selection. *Int. J. Cptr. Inform. Sci. 4*, 4 (1975), 307–327.

Short Communications
Management Applications

Remark on Uniform Insertion in Structured Data Structures

Clifford R. Hollander
IBM Scientific Center, Palo Alto

Key Words and Phrases: data structures, directed graphs, uniform insertion
CR Categories: 4.34

In a recent paper, Shneiderman and Scheuermann [1] have defined an interesting operation, called *uniform insertion*, whereby a (linked) data structure G_2 can be embedded within another structure G_1 in such a way that each data node of G_1 serves as an entry node for G_2. This communication points out and corrects an error in their definition, while preserving the spirit of that operation.

For convenience, let us summarize here the terminology from [1] that is relevant to the notion of uniform insertion:

Definition. A *structured data structure* is viewed formally as a directed graph $G = (e, D, L, \Gamma)$ where

(1) e is an *entry node* of indegree zero and is the only node of G which is immediately accessible from outside of G.[1]
(2) D is a set of *data nodes.*
(3) L is a set of *branch* (link) *labels.*
(4) Γ is a mapping which defines the internode branches, i.e. $\Gamma\colon (\{e\} \cup D) \times L \rightarrow D$.
(5) $\{e\} \cup D$ is *connected.*
(6) Each $d \in D$ is *reachable* from e.

Figure 1 depicts the graph for a one-way linked list: $G_1 = (e_1, \{a_i \mid 1 \leq i \leq n\}, \{FIRST, LINK\}, \Gamma_1)$ where

This work was done while the author was on leave at the Digital Systems Laboratory, Stanford University, Stanford, CA. It was supported in part by the Joint Services Electronic Program under Contract N00014-75-C-0601. Author's address: IBM Scientific Center 63G, 1530 Page Mill Road, Palo Alto, CA 94304.

[1] In [1], a set E, with $|E| = 1$, is used in place of e.