1975

Reorganization Points for File Designs with Nonlinear Processing Costs

K. Sundar Das

T. J. Teorey

S. B. Yao

Report Number:
75-160

REORGANIZATION POINTS FOR FILE DESIGNS WITH NONLINEAR PROCESSING COSTS

by

K. Sundar Das, T. J. Teorey, S. B. Yao*

CSD TR 160

Proceedings of the International Conference

on

Very Large Data Bases

September 22-24, 1975

* Current address: Department of Computer Science, Purdue University

REORGANIZATION POINTS FOR FILE DESIGNS WITH NONLINEAR PROCESSING COSTS

BY

K. SUNDAR DAS, T. J. TEOREY, S. B. YAO*

THE UNIVERSITY OF MICHIGAN

ANN ARBOR, MICHIGAN 48104

I. Introduction

An important problem in accessing and updating large scale data bases is the determination of when to reorganize the data and eliminate records that have been flagged for deletion but have not as yet been physically deleted. Such records take up valuable storage space and cause a degradation of access times to the active records, mainly because of the necessity to maintain overflow areas. After a certain length of time it is usually desirable to merge the old data and its overflow areas into a new data base of the original format, thus reallocating the physical storage for maximum efficiency within the constraints imposed by the data base organization. The purpose of this study is to determine the effect of various system and user parameters on the choice of optimal reorganization points over a wide range of file organizations and user applications. The analysis of file reorganization is seen as a first step toward understanding large scale data base reorganization.

The models for file organizations are based on the hierarchical access tree descriptions developed by Yao [3]. This generalized model is applicable to the access mechanisms for all file organizations (i.e., file designs) included in study. The model has been implemented and tested with actual files, and is called the File Design Analyzer program [1]. The program accepts as input, various parameters describing the file characteristics, user workload and hardware for storing the file. The output is a summary of secondary storage space and I/O processing time required to process the user workload.

A near-optimal data reorganization point is determined on the basis of minimum cost of user access to the file. Cost is defined in this context as the product of secondary storage space and I/O processing time for a given set of activities over a specified time interval t. Storage space is computed as the space to house the entire file including intermediate data such as pointers, directories, etc. I/O processing time for the user application includes the retrieval, modification, insertion and deletion of records; modification and insertion of keyword values at an intermediate access tree level; and query type retrieval. I/O processing time is defined as the elapsed I/O time to perform the user activities in a serial manner.

A definition of file growth in terms of keyword values in addition to the number of actual records in the file results in a nonlinear processing cost function for a linear file growth rate. An earlier study by Shneiderman [2] assumed a linear cost function and finite time horizon T, and a closed form solution for a fixed reorganization point n was determined. It is shown here that with nonlinear costs, an infinite horizon, and a fixed size workload, the time between reorganization points increases as the file grows. On the other hand, a workload level proportional to file size results in a series of rapidly decreasing reorganization points [1].

*Current address: Department of Computer Science, Purdue University.

II. Reorganization Point Determination

The decision rule for determining reorganization points under general conditions regarding growth rate, user activity level and time horizon should satisfy two conditions: optimality and efficiency of computation. The optimality criterion requires that total cost be minimized, but if the time horizon is unknown, the total cost is also unknown. Consequently for a time horizon that is only known at the time of occurrence (or shortly before), the optimal solution can only be approximated. One approach to this dilemma is to assume that the difference in total file accessing cost between a file that is reorganized and one that is not looks the same in the near future as it was in the recent past.

Assume that the cost of reorganization R_n is a parameter that changes very little with respect to total file accessing cost. Thus, one may reasonably conjecture that Eq. (1) below will provide a method that will reduce total cost within n time units after a reorganization point is established.

The decision rule to reorganize is defined as the smallest value of n for which

$$C_n \geq C'_n + \frac{R_n}{n} \tag{1}$$

where C_n = I/O processing cost during the n+1st time period when the latest reorganization occurred at the beginning of the first time period.

C'_n = I/O processing cost during the n+1st time period when the latest reorganization occurred at the end of the nth time period.

R_n = Reorganization cost at the end of the nth time period since the previous reorganization. In Eq. (1) the value R_n/n represents the average reorganization cost per time period over the past n time periods.

If we assume a linear file growth rate, both C_n and C'_n increase as n increases because the search time for a record increases as the file grows over time. R_n also grows because reorganization time is defined as the time to process all records in the file, including those flagged for deletion, plus the time to rewrite nondeleted records.

III. Experimental Results

In Table I file organizations are ranked in order of lowest time to service an application and in order of lowest time-space product of secondary storage. The latter measure is used as the cost criterion for reorganization point computations. In this case, a heavily query-type application, the inverted list structure performs the best and has the longest period between reorganizations.

Table I. Reorganization Points for File Organizations (Single Access)

FILE ORGANIZATION	TOTAL TIME (SEC.)	RANK	TIME-SPACE PRODUCT (SCALED)	RANK	% STORAGE OVERHEAD	1st REORG. POINT (DAYS)	20th REORG. POINT (DAYS)
SEQUENTIAL	$54*10^4$	5	29	5	6.2%	1	1
ISP	$1.42*10^4$	2	1.1	3	37.9%	6	7
DIRECT	$10.8*10^4$	4	6.9	4	21.8%	2	2
MULTI-LIST	$1.6*10^4$	3	.9	2	10.1%	11	15
INVERTED	$.98*10^4$	1	.57	1	14.3%	15	24

FILE ORGANIZATION	TOTAL TIME (SEC.)	RANK	TIME-SPACE PRODUCT (SCALED)	RANK	1st REORG. POINT (DAYS)	20th REORG. POINT (DAYS)	TIME DEGRADATION FROM SINGLE ACCESS (ratio of total time)
SEQUENTIAL	$452*10^4$	5	241	5	1	1	8.3
ISP	$13*10^4$	3	10	3	3	3	9.1
DIRECT	$18*10^4$	4	11	4	2	2	1.7
MULTI-LIST	$9.4*10^4$	2	5.2	2	4	5	5.9
INVERTED	$8.7*10^4$	1	5.1	1	5	6	8.9

Table II. Reorganization under Multi-Access Interference

Table II illustrates the influence of multi-access interference. The total cost is increased with varying amounts depending on the sequential or linked-list nature of the file organizations. The time between consecutive reorganizations is highly sensitive to the way in which a file organization is implemented. As access time and storage overhead parameters increase, reorganization points tend to decrease unless the cost of reorganization increases at a higher rate. In the test cases surveyed, this only occurred for the direct organization for the multi-access interference (shared disk) vs. single-access case (dedicated disk) because of the sequential orientation of its reorganization and therefore a greater degradation due to multi-access interference.

ACKNOWLEDGMENTS

The authors wish to thank Arnon S. Rosenthal and John DeHeus for their assistance and helpful insight. This research was supported by the Air Force Data Service Center, Defense Supply Service Contract No. MDA-903-74-C-0300.

REFERENCES

1. Das, K. S. and Teorey, T. J., *Detailed Specifications for the File Design Analyzer*, Systems Engineering Laboratory Technical Report No. 87, The University of Michigan, Ann Arbor, July, 1975.

2. Sheiderman, B., Optimum Data Base Reorganization Points, *Comm ACM* 16, 6 (June 1973), pp. 362-365.

3. Yao, S. B., *Evaluation and Optimization of File Organizations Through Analytic Modeling*, Ph.D. Thesis, The University of Michigan, 1974.