DYNAMIC ADDITION AND REMOVAL OF ATTRIBUTES IN BANG FILES

Nalini Hosur

Oklahoma Department of Vocational and Technical Education
Systems Design and Computer Services

Huizhu Lu and George Hedrick

Oklahoma State University
Computer Science Department
Phone: (405) 744-5673

ABSTRACT

BANG (Balanced and Nested Grid) files provide fast access and updates of large multidimensional databases maintained on secondary storage. This research extends the applicability of the BANG file structure by allowing both dynamic addition and dynamic removal of attributes.

Two new techniques, "Directory Modification Technique (DMT)" and "Multiple BANG File Technique (MBFT)" have been developed and tested. DMT uses a geometrical approach to accommodate the changes in the number of attributes while MBFT uses separate BANG files. Both of these methods are built upon the existing BANG file data structure.

The number of disk accesses for the DMT when appropriately normalized shows a linear relationship with the fraction of total records whose attributes were modified. This method is superior to the traditional regeneration of the BANG file for attribute modifications on less than 50% of the records in the data file.

MBFT shows significant improvement over a single large BANG file in the number of disk accesses for range and partial queries. As the fraction of the total data searched becomes smaller, the improvements become more dramatic. The absolute reduction in disk accesses is enhanced as the range increases, or the number of attributes in the partial query decrease. This method is particularly useful for those files with large numbers of attributes or, where the data is distributed over a combination of attributes.

1. INTRODUCTION

Large database applications are often designed so that major portions of the data reside on secondary storage. Access to secondary storage is relatively slow compared to operations on data in main memory. As a result, the time used for retrieval of data from secondary storage can become the dominant factor in determining the performance of the database system.

The BANG (Balanced And Nested Grid) file structure has proved to be an efficient approach to minimizing the number of accesses to secondary storage in response to queries, or for addition and deletion of records with multiple attributes.

Many practical applications however require an added capability: the addition and removal of attributes from an existing database either for some of the records, or all the records. For this reason, all recent relational databases allow the addition and removal of attributes. However implementing this capability in main memory may not be optimal for databases resident on secondary storage devices.

In view of the established superiority of the BANG file structure [FREE 87, FREE 89A, FREE 89B and LIAN 89] in minimizing accesses to secondary storage, it is highly desirable to investigate the efficiency of this structure with respect to the addition and removal of attributes. A naive and obvious approach is to change the contents of the entire database whenever we add or remove attributes from any record. However this can be highly inefficient and can seriously limit the applicability of BANG files in many practical applications. To date there is no published work dealing with efficient approaches for the addition and removal of attributes from BANG files.

The goal of this research is to explore two new techniques for the removal and addition of attributes within the structural framework of BANG files. The first technique is termed the "Directory Modification Technique (DMT)". A change in the number of attributes (dimensions) is handled by modifying the existing directory entries (region, level); the data stored on secondary storage is unaltered. DMT is based on a geometrically derived transformation function. The second technique is termed the "Multiple BANG File Technique (MBFT)". Records with unique combinations of attributes are grouped into distinct BANG files. MBFT allows dynamic modification of the number of attributes by moving records from one BANG file to another.

In section 2 we describe these two new techniques for adding and removing attributes from a BANG file. In section 3, we present results for each of these methods when we add and remove attributes; comparisons are made with the naive (rewriting) approach. In section

4, we analyze the advantages of each of the methods, and conclude with recommendations regarding the use of these methods.

2. TECHNIQUES FOR DYNAMIC ADDITION AND REMOVAL OF ATTRIBUTES

Before describing these new techniques, we will describe an intuitively obvious approach for addition and removal of attributes which we term the "Naive Approach (NA)".

Naive Approach (NA)

An attribute (or a key) in a multidimensional database can be associated with a dimension. An N attribute database can be considered as a hyper-rectangle in an N-dimensional space.

The BANG file structure provides a mapping function between the data buckets, and the regions in N-dimensional space that contain the data values. Algorithms for existing BANG files consider the number attributes fixed. Changing attributes for some or all records requires the creation of a new BANG file, with new (modified) number of attributes requiring modification of the data in every data bucket. Since these data buckets are usually resident on secondary storage, this approach is heavily penalized in data access time. The shortcomings of this method are particularly evident when the number of attributes is altered for only a small number of records.

In view of these shortcomings, it is necessary to explore alternate approaches for the addition and removal of attributes that are more efficient in terms of access to secondary storage are required.

DIRECTORY MODIFICATION TECHNIQUE (DMT)

When an attribute is added or removed there is a change in the number of dimensions; it is possible to modify the directory entries without redistributing the records. This leads to significant savings in disk access time over the NA.

In order to illustrate the basic idea, consider a special, simple problem where the addition and removal of an attribute can be done without undue complications. A data base with two attributes can be represented as a rectangle in 2-D space, say X-Y plane. The addition of a third attribute changes the representation to three dimensions, X,Y, and Z. The 2-D distribution forms the X-Y plane in the 3-D case which is shown in Figure 1. All the points in X-Y plane in 2-D still lie in X-Y plane in 3-D, except that the values of the z coordinate are zero.

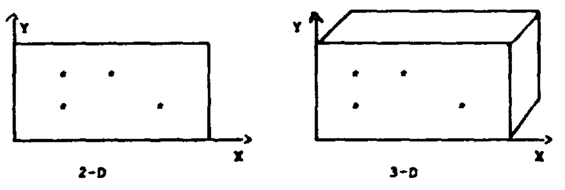


Figure 1. Distribution of Set of Points in 2-D and 3-D

The removal of attributes is also based on the geometrical approach. In this case it is necessary to assume that the values of the attributes that are being removed are zero. When any dimension is removed from a data base with dimensions, the data can be represented in H-1 dimensions by suitable modifications to the directory without redistributing the records.

For the above simple example it is not difficult to implement the DMT to add and remove attributes. However for more general problems, notably those that involve buddies, the structure of the BANG file introduces serious complications. A number of problems occur at almost every stage, and it is necessary to develop individual approaches to handle these problems which make this a highly involved and complex approach.

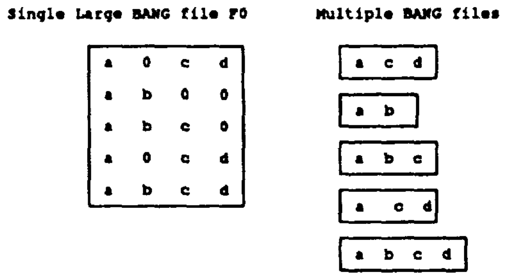
The details of the implementation of the DMT in the general case are too involved to be satisfactorily described here, and the reader is referred to [HOSU 91]. We mention here that in the development of the DMT for the general case some unexpected properties of the BANG file structure have been uncovered; these have not been published in the literature but have the effect of drastically increasing the complexity of developing and coding for the DMT.

Multiple BANG Files Technique (MBFT)

This approach for the addition and removal of attributes differs from the previous technique by maintaining separate BANG file for each unique combination of attributes. Multiple files are generated to accommodate changes in the dimension or the combination of attributes in the database. We describe the organization of the BANG files followed by the discussion of addition, and deletion of attributes.

Organization of Multiple BANG Files

Consider the single BANG file, where all the records with different combination are stored in one big file F0, it would look as shown in Figure 2a. The Multiple BANG files, corresponding to the same data is shown in Figure 2b.



(a) Traditional (b) Multiple BANG

Figure 2. Record Distribution in 'Traditional', and 'Multiple BANG' Files

In the above data all the records do not have all 4 attributes. Instead of maintaining one large BANG file with four attributes, the records can be distributed into different BANG files according to the number and combinations of the attributes. Although 'ab' and 'ac' have the same number of attributes, the combination is different and thus are stored in separate files.

Each BANG file contains a unique set of attributes. As the modifications (insertion/deletion) of attributes is increased, the combination of attributes also increases, resulting in a proliferation of files, (2**n - 1, in the worst case, where n is the number of attributes). In the above example, there are 4 attributes, taking all combinations into consideration, there can be 15 files. However, in practical situations, the number may be considerably lower, because all the combinations may not be present.

For any operation, including addition/deletion of records, or query etc., it is required to search the BANG file associated with that set of attributes. The operation is performed after locating the required file(s). In case of partial queries, it may be necessary to search more than one file; thus it is essential that we organize the collection of BANG files in such a way that the search is reasonably efficient.

There are many possibilities; one is to store the BANG filename as the nodes of a binary search tree. This structure is quite efficient for an exact query; for a partial query, more than one BANG file may need to be searched. In the worst case, when all the files are to be searched, this will cause a search of the entire tree.

Linked lists (with the BANG filename as its elements), are not as efficient as trees for exact queries. But it is easy to update the lists as required by the addition/removal of attributes. To date there is no best structure available to process range/partial query efficiently. The author proposes the following structure to organize the BANG files.

Let there be 4 attributes ('a', 'b', 'c', 'd') in a database. Figure 3., illustrates all possible BANG files and organization of these files using linked list data structure. An array with the size of the number of attributes is maintained. Each element of the array is associated with a list. Each element in a list is a BANG filename, and the filenames are arranged in alphabetical order.

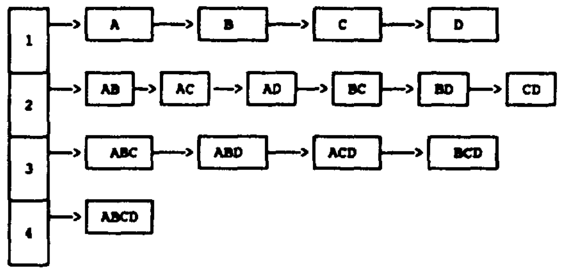


Figure 3. Organization of Multiple BANG Files

Note : symbol $\boxed{\text{AB}}$ represents a BANG file with attributes 'a', and 'b'.

For an exact query of attributes 'abcd', only the list associated with four attributes is searched. In case of a query involving attributes 'a' and 'd', lists of 2, 3, and 4 attributes need to be searched. In this particular case, it becomes a range query for file 'AD' and a partial query for files 'ABD', 'ACD', 'BCD', and 'ABCD'. Once the files are found, operations on each file are handled as in the case of a single BANG file.

During addition and removal of attributes, the major disk accesses are due to the updates of records, with changes in number of attributes. There is no need to rewrite the entire database. Since there are separate files for each unique set of attributes, there is no need to deal with maximum attributes for every operation, as in the NA, and DMT.

3. RESULTS AND DISCUSSIONS

The DMT and MBFT were developed and tested for dynamic modification (addition, and removal) of attributes. Comparison were made with the 'Naive Approach'. Results, discussion, and simulation specifications for each method are provided separately. Uniform random distributions are used to generate test files. Performance evaluation is based on the number of disk accesses for each of the two proposed techniques DMT, and MBFT, compared with the NA.

Results and Discussion for DMT

Evaluation of this technique is based on the number of disk accesses for addition and removal of attributes as compared to the number of disk accesses with the 'Traditional' BANG file method.

Addition of attributes (2-D ---> 3-D)

Start with a 2-dimensional data and modify into 3-dimensional data.

Removal of attributes (3-D ---> 2-D)

Start with a 3-dimensional data, modify it into 2-dimensional data (using the DMT).

The DMT was used on 2 sets of data, with data sizes of 1,000, 2,000, and 5,000 records in each set. The two sets of data represent different realizations of random numbers and are used to estimate the variability of the results. For each data size, disk accesses, data occupancy, and directory occupancy were computed for addition/deletion of attributes for 10, 25, 40, and 50% of the total number of records.

The results of the disk accesses for addition and deletion of attributes corresponding to the two sets are consistent. Consequently the results can be discussed for one data size, namely 5000.

The results of the disk accesses for addition of attributes corresponding to the above data is shown on Figure 4a. They have been normalized by the data accesses for recreation of the BANG file from scratch. The results for deletion are shown in Figure 4b.

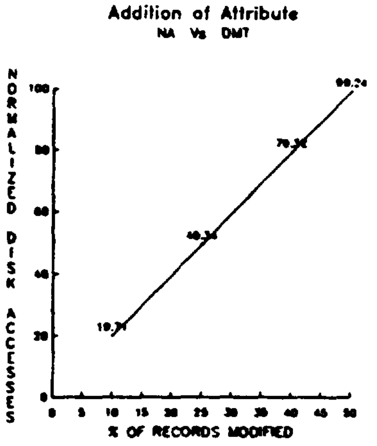


Fig 4(a) Test File I (n = 5,000)

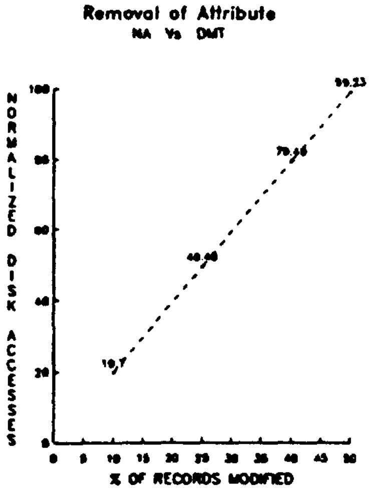


Fig 4(b) Test File I (n = 5,000)

The normalized disk accesses show consistent behavior during addition and deletion of attributes, for both data sets and also between the three data sizes. The relationship can be well approximated by a line through the origin with a slope of 2, especially when the number of records modified is low; consequently, it is seen that the DMT is superior to regeneration of the BANG file for addition/deletion of attributes for half the records of the data set.

Results and Discussions for MBFT

The second technique, the MBFT, stores the data in multiple files. A separate BANG file is created for each unique combination of attributes.

There are two important considerations in evaluating the Multiple BANG file approach; one deals with the disk accesses connected with the creation of the file, and the second with the accesses connected with the queries. The disk accesses associated with the addition, and deletion of records are well understood: namely four disk accesses for each operation. Consequently, testing of the method was restricted exclusively to disk accesses resulting from queries.

Queries may be divided into three types: exact queries, range queries, and partial queries. In exact queries, the values of all the attributes are specified, and finding the record consists of searching the bucket in which the record is stored. Exact queries require two disk accesses for both 'Traditional' and 'Multiple BANG' files; thus there is no need to perform any tests, since the results are known in advance.

Range queries specify the range of all attributes. In partial queries, the range is specified for one or more attributes but not for all the attributes. All allowable values are accepted for attributes whose range are not specified. This section deals with the comparison of disk accesses for range and partial queries.

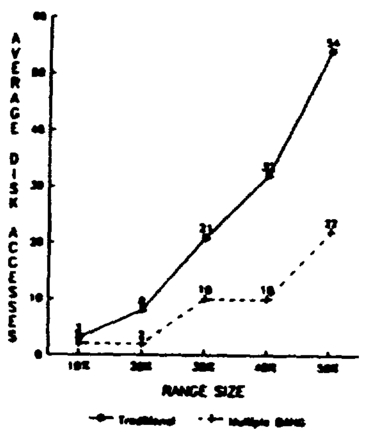


Fig 5(a) n = 5000 (25% with 3rd attribute)

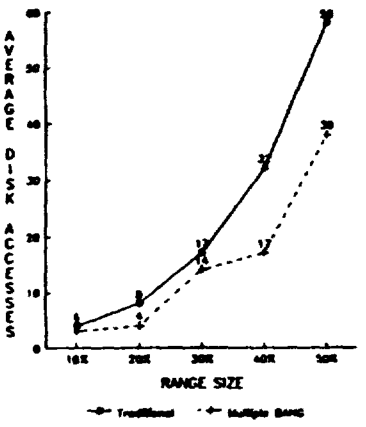


Fig 5(b) n = 5000 (50% with 3rd attribute)

Results and Discussions for Range Queries

Comparison of disk accesses for the NA, and the MBFT were performed using 3 data sizes (1000, 2000, and 5000) with three attributes. For each data size, the third attribute was specified for 25%, and 50% of the total records. Consider an example of 2,000 records with three attributes 'a', 'b', and 'c'. A value for 'c' may be specified for either 25% (500 records), or 50% (1000 records) for the data; the rest of the records are assigned a zero value for 'c' attribute.

The range query was performed on three attributes, and the range size was varied from 10 to 50% in increments of 10%. The results of the disk accesses for each of three data sizes are consistent. Consequently, the results can be discussed for one data size, namely, 5,000 records.

Figures 5, show the consistent reduction in disk accesses for the MBFT over the NA for all range values. As the range increases, the difference gets larger, and the superiority of the MBFT is clearly demonstrated.

Results may also be analyzed in terms of the percent of data to be searched. As the percentage of records with the third attribute gets lower, the difference gets larger,i.e the results are better for the case where the third attribute is specified for 25% rather than for 50%. The reduced disk accesses come from the searching of fewer records.

Disk accesses show a non-linear dependence on the range. In general disk accesses appear to increase as a power of the range. This pattern is seen for the two cases with 25%, and 50% of the third attribute defined. Additionally a simple doubling from 25% to 50% is not observed. A comparison of disk accesses for the 'Traditional' BANG file method, and 'Multiple BANG' files for the larger ranges, shows a reduction of 60% for the case where 50% of the records have third attribute. The reduction is even more significant, around 45%, for the case where only 25% of the data has the third attribute.

Results and Discussions for Partial Queries

The comparison of disk accesses for partial queries for the 'Traditional', and 'Multiple BANG' files were tested on the following examples.

Four attributes (4-D) were chosen to simulate some of the complexity that an arise in partial queries as the number of attributes is increased. It was also felt that a database with less than 5,000 records would not be statistically acceptable in terms of records encountered.

The database with 4 attributes, 'a', 'b', 'c', and 'd'. Figure 6, shows the actual combination of attributes and number of the records distributed in different files, used for partial query in the simulation.

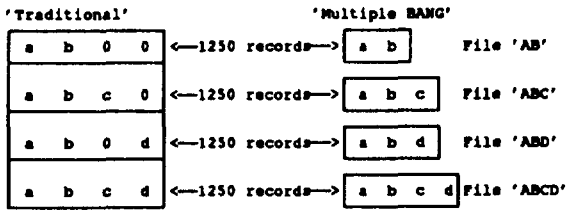


Figure 6. Single large and Multiple BANG files used for partial query

In a general four attribute data set, there can be fourteen partial queries, and a range query. The partial queries are 'a', 'b', 'c', 'd', 'ab', 'ac', 'ad', 'bc', 'bd', 'cd', 'abc', 'abd', 'acd', and 'bcd', and the range query is 'abcd'. The data distribution selected here allows attributes 'a', and 'b', and 'c', and 'd' to be interchanged. Consequently only seven types of partial queries namely, 'a', 'c, 'ab', 'ac', 'cd', 'abc', 'acd' and one range query 'abcd' exist.

Partial and range queries were selected for the following combinations 'a', 'ab', 'ac', 'abc', 'bd', 'cd', 'bcd', and 'abcd'. A range of 10%, 30%, and 50% is taken for each combination of attributes. The ranges start at different positions i.e. 0, 25, and 50% for each attribute.

The observed results appear to depend on two key quantities: the percent of data searched, and the number of attributes used in the query.

A search is said to be 100%, if the search covers the entire file. Similarly 50%, and 25% search indicate the percent of the file to be searched.
Let us consider the results for a two-attribute partial query. Figures 7.,show the disk accesses for partial queries 'ab','bd', and 'cd'.

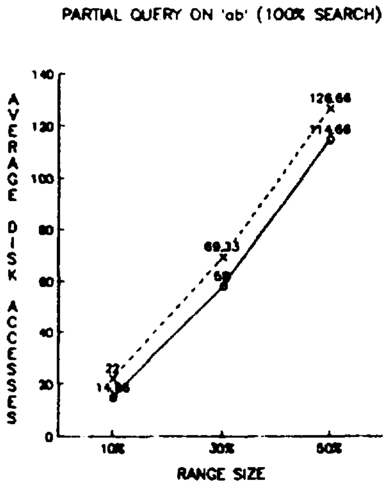


Fig. 7(a)

Partial query 'ab' requires a 100% search. It becomes a range query for file 'AB', a partial for 'ABC', 'ABD, and 'ABCD'. The MBFT uses four files, and needs between 8 to 12 additional disk accesses over the single BANG file. The search of three additional BANG files introduces 6 disk accesses (2 per BANG file). There also appears to be a small increase in accesses resulting from multiple accesses of a given 'ab' range in the separate BANG files.

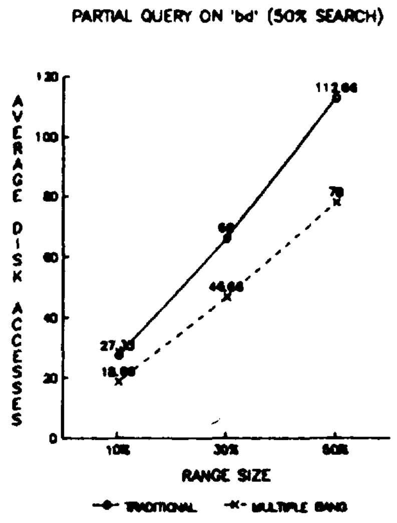


Fig. 7(b)

The partial query for 'bd' (and also 'ac') requires a partial search for file 'ABD'('ACD') and 'ABCD', equivalent to a 50% search. The MBFT requires less than 7/10th the number accesses required by the NA. The difference in disk accesses get larger as the range is increased.

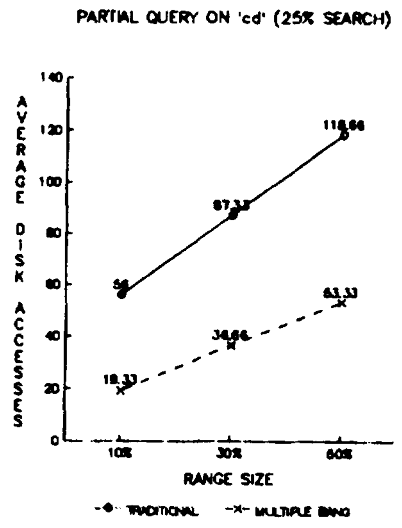


Fig. 7(c)

The partial queries for 3 attributes, 'abc', and 'bcd' Figures 8., require a partial search of the file 'ABCD' (50% and 25% respectively). The improvement in the disk accesses of the MBFT are even more dramatic, requiring less than 45% of the accesses required by the NA. Again the absolute difference gets larger as the range is increased.

The effect of the number of attributes on partial/range queries Figures 9., is seen by comparing the results of the 25% search, queries for 'cd', and 'abcd'. It is seen that the largest absolute difference in disk accesses between the NA, and MBFT occur for 'cd' and the smallest for 'abcd'. The volume of the data space that falls within the specified range for a given partial query is related to the range and the number of attributes.

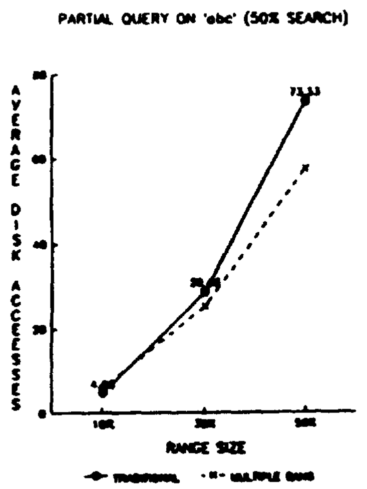


Fig 8(a)

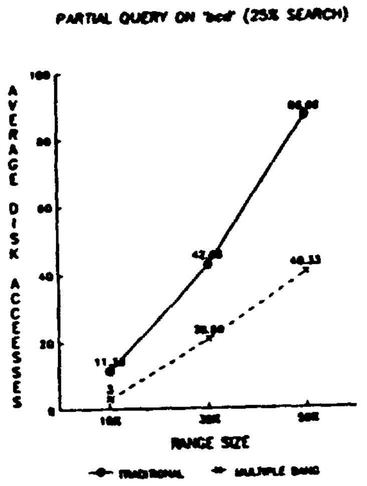


Fig 8(b)

For a given range, this volume decreases as the number of attributes are increased. The ratio of the disk accesses of the MBFT to the single BANG file approach for the 50% range increases slightly from 0.45 to 0.46 to 0.48, for 'cd', bcd', and 'abcd' respectively.

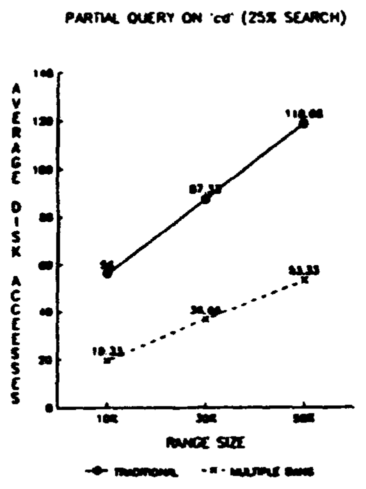


Fig 9(a)

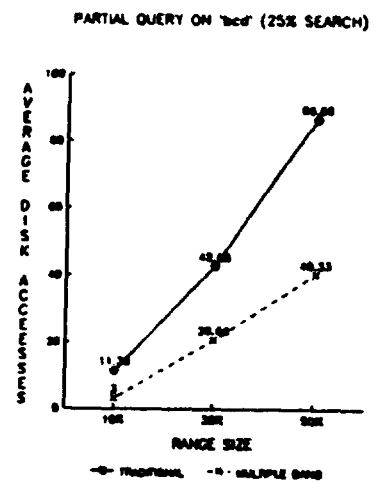


Fig 9(b)

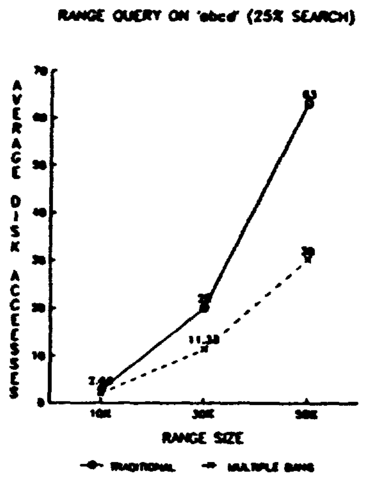


Fig 9(c)

The results clearly demonstrate the superiority of the MBFT over the NA method for range and partial queries. As the fraction of the total data searched becomes smaller, dramatic improvements in disk accesses are observed.

4. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

Two new techniques, "Directory Modification Technique (DMT)" and "Multiple BANG File Technique (MBFT)", have been developed and tested to extend the applicability of BANG files, by allowing insertion/deletion of attributes. DMT preserves the existing data structure of the data and modifies the dimension of the directory to reflect the addition or deletion of attributes. The MBFT generates separate BANG files for each unique combination of attributes. Both these new methods have been developed to deal with changing attributes for a portion of the data, rather than for the whole data set. The Naive Approach (NA) has to recreate a BANG file from scratch each time the number of attributes are modified for any record(s) in the data set.

Comparison of both the DMT the MBFT with the NA were made on the basis of disk accesses required for addition and deletion of attributes as well for queries.

The first technique, DMT, has been developed to allow for arbitrary addition and/or deletion of attributes. The normalized disk accesses for the DMT show a linear relationship with the fraction of records whose attributes are modified; the slope of this line is 2. Consequently, DMT is superior to generation of the BANG file from scratch for attribute modifications of up to 50% of the records in the data file.

In practical examples where the number of attributes modified at a time is only a small fraction of data say less than 5%, DMT takes less than 10% of the disk accesses required for regeneration of the BANG file, a significant reduction in the disk accesses.

One of the limitations of the DMT is that, for every operation we have to deal with the maximum number of attributes/dimensions, even though all the attributes may not be present for all records. This is also true in the case of NA. The DMT can only match the disk access efficiency of the NA which is inefficient for range/partial queries when a large number of records have zero values for the attributes. The second technique MBFT, obviates the problems associated in dealing with databases where a large number of records have zero values for some of the attributes.

The generation of multiple BANG files requires disk accesses comparable to DMT. Consequently, this technique like the DMT is vastly superior to the NA when the number of records whose attributes are modified is small.

MBFT shows significant improvement over NA in disk accesses for range and partial queries. The improvement is dependent on the fraction of the data searched for range/partial queries between the single file and the multiple BANG files accessed by the query. As the fraction of the data searched becomes smaller the improvements become more dramatic. The absolute reduction in disk accesses is enhanced as the range is increased, or the number of attributes in the partial query are decreased. A reduction of greater than 50% in disk accesses is observed when the data searched is 25% of the single BANG file.

The MBFT files approach is the superior method. It will be particularly useful for large attribute files, where the data is distributed over a combination of attributes. DMT is useful if the data ends up with a full complement of attributes for all records after repeated modifications.

Adequate testing of statistical variability resulting from the data distribution needs to be studied in future work. This can be accomplished by averaging results over a large number of random realizations of attribute values for both the techniques. Non-uniform data distribution also needs to be tested.

In the case of DMT, range/partial queries should be performed to evaluate the effect of data occupancy on the disk accesses. This can then suggest an optimum percentage of attribute addition by DMT after which the BANG file needs to be regenerated from scratch.

This research presents the linked list structure in the case of MBFT. Evaluation of data structures adapted to partial queries for a large number of files can enhance the applicability of this powerful approach.

5. REFERENCES

[FREE 87] Freeston, M. " The BANG file : a new kind of grid file" Proc ACM SIGMOD (Dec 1987), 260-269.

[FREE 89a] Freeston, M. "Advances in the design of BANG file "Third International Conference on Foundations of Data Organization and Algorithms, Paris, June 1989

[FREE 89b] Freeston, M. "A well behaved file structure for the storage of spatial objects" Symposium on the Design and Implementation of Large Spatial Databases Santa Barbara, California, July 1989.

[HOSU 91] Hosur, N.T., "Dynamic Addition and Removal of Attributes in BANG Files" O.S.U., M.S. Thesis 1991.

[LIAN 89] Lian, T., Fisher, D., and Lu, H. "Implementation and Evaluation of Grid and BANG (Balanced And Nested Grid) File Structure" Proc. of Workshop on Applied Computing 1989, 80-85.

RELATED BIBLIOGRAPHY

[BENT 80] Bentley, J.L., and Maurer, H.A. "Efficient Worst-Case Data Structures for Range Searching" Acta Formatica 13, 1980, 155-168

[BURK 83] Burkhard,W.A. "Interpolation based index maintenance" Proc ACM symp Principles of Database Systems (1983), 76-89.

[CHUN 89] Chun,S.H., Hedrick G.E., Lu H., and Fisher D.D. "A Partitioning Method for Grid File Directories" IEEE Proc of the 13th Annual International Computer Software and Applications Conference 1989, 271-277.

[GUTT 84] Guttman, A. "R Trees: A Dynamic Index Structure for Spatial Searching" ACM 1984 47-57.

[HINR 85] Hinrichs, K. " Implementation of grid file : Design concept and Exp" BIT 25, 3 (1985) 569-582.

[HUTF 88] Hutflesz, A., Six Hans-Werner., and Widmayer, p. "Twin Grid Files" ACM March 1988, 183-198.

[IYEN 88] Iyengar,S.S., Rao,N.S.V., Kashyp,R.L., and Vaishnavi, V.K. "Multidimensional data stuctures: Review and Outlook" Adv. in computers Vol 27(1988), 69 - 119.

[NIEV 84] Nievergelt,H.J., and Sevcik,K.C. " The Grid File: An adaptable, symmetric multikey structure" ACM Trans on Database systems vol 9, No.1, (March 1984).

[SARI 87] Saritepe, H.N.A. "An Analytical Study of Grid File and k-d-B Tree Structures" O.S.U., M.S. Thesis 1987