

The Grid File: An Adaptable, Symmetric, Multikey File Structure.

J. Nievergelt, Hans Hinterberger, Kenneth C. Sevcik

Abhishek Srivastava
Student ID: 861307778

February 2, 2017
CS 236, Winter 2017

The problem:

This paper presents problems with single-key file system to provide good performance for efficient access to records based on the any one or combinations of its attributes. Traditional file system does not do well against highly dynamic files. And also did not perform well for partially specified queries and had high disc access for range queries.

The contribution:

The authors propose a 'Grid' file system which is symmetric, adaptable, efficient for multi-key queries and can support very large high-dimensional data. The file structure was based on two principles: *Two-disk-access principle* and *Efficient range queries with respect to all attributes*. These principles were important to provide a system which has a fast response and can support complex range queries along all dimension of space.

The method:

The design of Grid multi-key file structure is done by using *bitmap representation*, which reserves bits for each possible records and it resulted in very large sparse matrix and is stored using compression with help of a dynamic directory which manages all the dynamic partitions in key-value form.

To support multi-key file structure it utilizes Grid partitions or blocks and all dimensions are treated symmetrically. For grid partition it was assumed that all attributes are independent and each partition can be further divided or merged with adjacent partitions. Grid partition help support fast range queries and partially specified queries.

In grid file structure buckets were used to store records and are mapped with grid blocks, buckets are nothing but disk allocated storage unit of fixed size. Grid directory was used to keep correspondence between grid blocks and buckets, it consists of grid array and linear scales. Grid directory supported the situations where bucket overflow or underflow by splitting or merging records and storing it in the buckets. buckets can be shared between different Grid blocks and are pairwise disjoint.

Grid file structure focus on these further decisions to provide general purpose file structure suited to multikey access:

- choice of splitting and merging policy
- implementation of grid directory
- concurrent access.

Comments:

The paper presents a novel file structure that supported efficient multi-key queries and also conducted comparisons with single key file structures. However, it also gives rise to a few problems:

- The Grid file structure utilizes the sparse nature of the data but if the data is very large this will result in less grid partitions and very large number of buckets which will result in can affect directory and bucket size.
- Grid file does not support dependent attributes because it will again effect their sparse nature.