

Generalized Search Trees for Database Systems

Author: **Joseph M. Hellerstein, Jeffrey F. Naughton, and Avi Pfeffer**

Reviewer: **Mu-Woong Lee**

Summary:

This paper proposes a general framework of tree-like index structures, Generalized Search Tree (GiST). The authors indicate that most index structures (i.e. B-tree, R-tree, etc) have similar structures, only some details are different from each other: the type of the predicates, comparison rules of the predicates, the node splitting method, and so forth. For this, GiST provides a general height balanced tree structure, then users can define their own predicates matching rules and split algorithms. Therefore, GiST is easily extensible both in data types it can index and queries it can support. To illustrate the flexibility of the Gist, GiST-based implementations of B-tree, R-tree, and RD-tree are presented, and their performance issues are also described.

Comments:

The basic idea of GiST, a general purpose tree structure is pretty impressive idea for me. However, I think that more empirical studies for performance issues are needed to justify the effectiveness of GiST. More precisely, performance comparisons between GiST-based index structures and their dedicated implementation structures should be presented. Actually, I'm not sure that GiST is a good solution, because the performance of the index is a critical factor of the entire system performance in many cases.

Furthermore, most modern DBMSs provide ways to implement user's own index structures easily and effectively, and they can possibly show better performance than dedicated index structures, because of the highly efficient storage management of the modern DBMSs. Consequently, the effectiveness of GiST is diminished for real applications.