Research on B-tree in Embedded Database SQLite

Di Nan

Computer Science and Technology, Harbin Engineering University
Harbin, China
lijingmei@hrbeu.edu.cn

Abstract. The methods that format the data in database and organize the structure of B-tree-page in open-source embedded database SQLite is described, and the database which is established by SQLite is analyzed. Further, the code of achiving this part of SQLite is analyzed in depth. Finally, the code to operate the red-black tree is designed, which uses the interface function provided by SQLite, then realizes the application of B-tress in the database.

Keywords: Database, B-tree-page, SQLite.

1 Introduction

Embedded database SQLite runs in the process of application program directly, so SQLite can work in the zero-configuration mode and occupy little resources. SQLite is realized by C language, which provides full independence and openness even does not depend on other resource out of the application program. SQLite is introduced as one of PHP V4.3 options and built on PHP V5 [1]. SQLite supports SQL92 standard and can be ported to all major embedded operating systems, even supports a lot of major high-level languages (such as C and Java, etc.). In addition, SQLite is also good robust, and can handle 10,000 CTR on Web site everyday[2]. The size of database file that is created by SQLite is up to 2 TB, and each database can store in the single file system entirely. These files can be transported between the computers even in the different byte order. These data are stored on disk in the form of B-tree, and SQLite gets the authority to access its database from file system. If SQLite is used to manage the database, the first step is to analyze the method how SQLite realizes B-tree.

2 The Structure of B-tree

B-tree was invented by R. Bayer and E. McCreight in 1970, and its structure is different from the multi-index formed by order index[4]. All the nodes in B-tree have the same structure as shown in Figure 1, which searches the node with the number of n-1 (such as K1, K2,…, Kn-1) and the pointer with the number of n (such as P1, P2,…, Pn), every node in the search codes is stored in order. Pointer P_i (i = 1,2,…, n-1) points to the pointer barrels that have search code K^i, and any pointer in the pointer barrels points to file with the search code K_i.

Y. Wang and X. Zhang (Eds.): IOT Workshop 2012, CCIS 312, pp. 509–515, 2012.

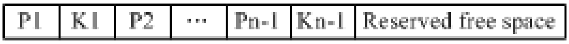

Fig. 1. The structure of node in B-tree

2.1 Leaf Node in B-tree

The search code of each leaf node is in the number of 1 to n-1, and it is different from each other. If index of B-tree is fully indexed, any search code in database files must be in a leaf node only once. In addition, the leaf nodes are sorted to linear sequence by the value of search code, so SQLite can use the pointer Pn of each leaf node to connect the leaf nodes orderly, as is shown in Figure 2, which makes B-tree efficient to operate the database file in linear.

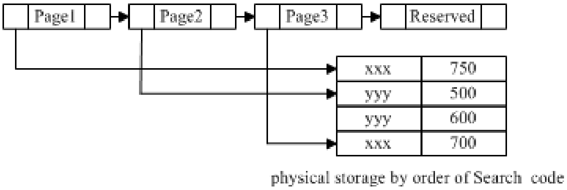

Fig. 2. Structure of leaf nodes in B-tree with n = 3

2.2 Non Leaf Node in B-tree

B-tree uses the non leaf nodes to form the multi-index of the leaf nodes, and the structure of the non leaf node is the same as the leaf node. No leaf node also contains search code to form the structure of storage unit, However, all the pointers of the non leaf nodes point to the nodes in B-tree.A non leaf node contains pointers with the number of m ($1 \le m \le n$), if m <n, from P_m all the free space is reserved. For example, a non leaf node have pointers with the number of m, the pointer P_i (1 <i <m) points to a sub-tree, In the nodes of this sub-tree all the search code that are less than K_i are equal to or greater than K_i-1. Pointer P_m points to the part of sub-tree that contains the search code greater than or equal K_m-1, and the pointer P_1 points to the part of sub-tree that contains the search code less than K_1.

3 The Structure of Database File

If there is a lot of data in the database that is organized by B-tree, it is much faster to achieve the operation such as searching, removing and adding. But the structure of large database is very complex, so in this paper a simple database that can be analyzed easily is established, besides this database file only has only one table.