Information Sciences 179 (2009) 3136–3161

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A B-Tree index extension to enhance response time and the life cycle of flash memory

Hongchan Roh, Woo-Cheol Kim, Seungwoo Kim, Sanghyun Park *

Department of Computer Science, Yonsei University, South Korea

ARTICLE INFO

Article history:
Received 8 November 2007
Received in revised form 15 January 2009
Accepted 4 May 2009

Keywords:
B-Tree index extension
Flash memory life cycle
Flash memory response time
Write count reducing
BFTL
MB-Tree

ABSTRACT

Flash memory has critical drawbacks such as long latency of its write operation and a short life cycle. In order to overcome these limitations, the number of write operations to flash memory devices needs to be minimized. The B-Tree index structure, which is a popular hard disk based index structure, requires an excessive number of write operations when updating it to flash memory. To address this, it was proposed that another layer that emulates a B-Tree be placed between the flash memory and B-Tree indexes. This approach succeeded in reducing the write operation count, but it greatly increased search time and main memory usage. This paper proposes a B-Tree index extension that reduces both the write count and search time with limited main memory usage. First, we designed a buffer that accumulates update requests per leaf node and then simultaneously processes the update requests of the leaf node carrying the largest number of requests. Second, a type of header information was written on each leaf node. Finally, we made the index automatically control each leaf node size. Through experiments, the proposed index structure resulted in a significantly lower write count and a greatly decreased search time with less main memory usage, than placing a layer that emulates a B-Tree.

1. Introduction

Recently, flash memory, which has advantages in the mobile environment, has become more popular as the usage of mobile devices has increased. Flash memory is suitable for the mobile environment because it is non-volatile like hard disk drives, and more shock-resistant and efficient in energy consumption than hard disk drives.

However, flash memory has several disadvantages such as long write latency, a short life cycle, and impossible in-place updates (which means that overwriting as a unit of flash memory page is infeasible). A page is the smallest unit of input and output, and several contiguous pages compose a block (in this paper, a page means a flash-memory page). Table 1 indicates that writing a page is approximately 5.5 times slower than reading a page in the case of the MLC NAND type flash memory. Another disadvantage is that flash memory devices have to erase a block to overwrite even a single page. Thus, in order to rewrite a page, most flash memory storage systems do not erase the entire block but rewrite the updated data to a free page of another block. Finally, the maximum erase count of each block is limited to about 10,000–1,000,000 operations [10]. If the erase count of a block exceeds this threshold, flash memory can no longer store data on the block.

The B-Tree [2], one of the most popular indexes based on hard disk drives, focuses on minimizing the access count to a hard disk. However, a B-Tree on flash memory without any conversion causes a greater number of write operations than a B-Tree on a hard disk. Consequently, the response time to user requests on the index increases because of the long write latency of flash memory, and its life cycle is shortened by frequent erase operations caused by numerous write operations.

* Corresponding author. Tel.: +82 2 2123 7757; fax: +82 2 365 2579.
E-mail address: sanghyun@cs.yonsei.ac.kr (S. Park).

0020-0255/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2009.05.007