Advanced Science and Technology Letters
Vol.107 (Ubiquitous Science and Engineering 2015), pp.66-69
http://dx.doi.org/10.14257/astl.2015.107.16

Enhancement study on NAND flash memory-based failure recovery using F-tree

Seong-Soo Han[1], Chang-Ho Suk[2]

[1] Bucheon University of Information and Communication, Adjunct Professor
[2] Seoil University of Computer and Software, Adjunct Professor

Abstract. Recently, NAND flash memoryis being broadly used for portable information memory devices for its low electricity consumption, favorable portability and strong durability. Its usage rate on bulk storage devices, such as SSD, has also been rapidly increasing. Accordingly, there are many studies being actively conducted regarding the improvement of memory capacities and effective storage techniques. B-Tree is one of the most commonly used tree structure in NAND flash memory. Yet, it has a shortcoming of deterioration of performance due to the frequent node insertion and repeated requests for deleting and writing processes. The purpose of this study is to propose an effective failure recovery technique for NAND flash memories using F-Tree as a solution to improve such shortcoming.

1 Introduction

With its characteristics of having an affordable price, low power consumption, portability, permanent storage, non-volatility, mass storage, its component size, and strong durability, NAND flash memory has recently been receiving attention as the storage medium for mobile devices such as smartphones as well as for SSD-equipped portable computers. Furthermore, due to the outstanding interface compatibility of SSDs created with mass storage flash memory, it has established itself as a HDD-substitute storage medium[1,2].

The B-Tree series of indexes is widely used as the index storage technique for existing flash memory. However, B-Trees have an issue when carrying out operations of writing in the unit of pages and erasing in the unit of blocks. The writing operation occurs frequently due to the insertion and deletion of the tree node, and this kind of writing operation can result in memory performance degradation as well as shortening the lifetime of the NAND flash memory. In order to improve these kinds of disadvantages, this study proposes the method that uses an F-tree that has the ability to build an index of exceptional performance and has a quick failure recovery in order to reduce the burden of writing operations and erasing operations and improve performance.

By using an F-Tree index, this study constitutes a system that has the ability to quickly recover in the occurrence of disorder by using the log directory-building F-

ISSN: 2287-1233 ASTL

ISLD technique (F-Tree Index Segment Log Directory technique: hereinafter referred to as the F-ISLD technique).

2 Related Research

2.1 B+-Tree Technique

This technique uses the B+-Tree in the system structure that uses an SSD as the NAND flash memory base to comprise an SSD. Also, the NAND flash memory searches data through the data structure and algorithm using the B+-Tree. However, the B-Tree node is composed of multiple entries and is inserted in the unit of entries when the B-Tree is built and so repetitive writing operations occur in specific areas. Therefore, the concentrated writing operations in specific areas can cause severe performance degradation due to the flash memory's inability to renew in its place, and has the disadvantage of degradation in performance needed for building within the NAND flash memory due to the occurrence of concentrated writing operations in restricted areas when building a B-Tree[3-4].

2.2 MR-Tree Technique

The MR-tree[4] is a space index structure modified from the R-Tree series index that allows space data access without using the generally-used main memory as the disk-based index. The MR-Tree allows the efficient access and management of mass data. However there is a burden in which many sectors are decoded or re-written in the same location when performing rebalancing operations such as insertion and deletion operations as well as division or merging[5].

3 Failure Recovery Technique of a NAND Flash Memory Base that uses an F-Tree

3.1 F-Tree Index System Structure

The F-ISLD technique recovery structure is composed of an F-Tree index and log directory, and the diagram is shown in [Image 1]

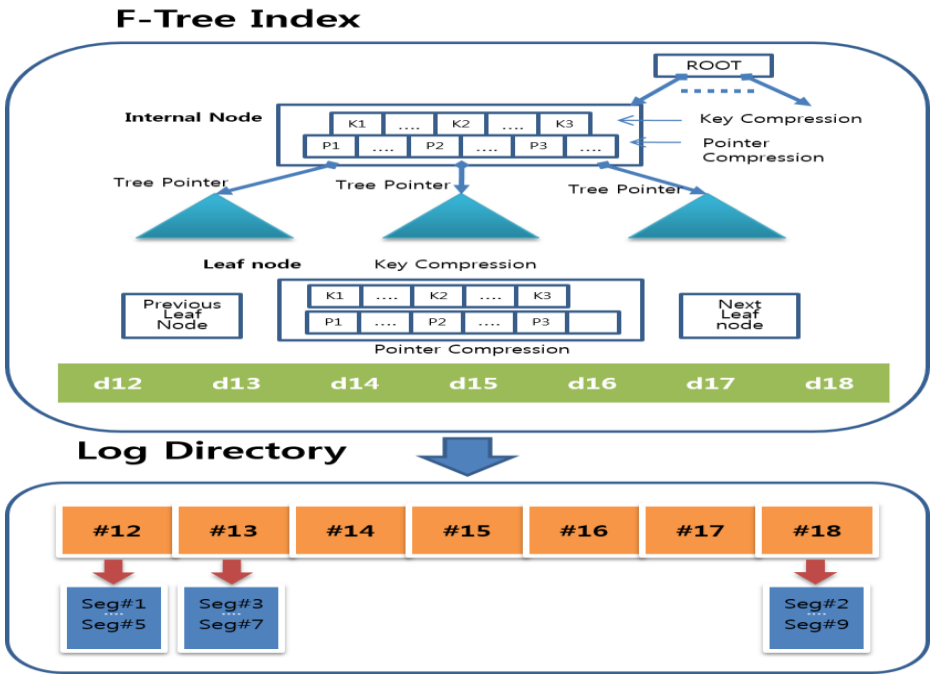

Fig. 1. F-ISLD Technique Recovery Structure

3.2 F-Tree Index Algorithm [6]

The F-Tree is a structure that reflects the composite operation of compression and subsequent writing of the existing B+-Tree structure. The node is compressed by separating the key section and the pointer section in order to raise compression efficiency. Because the F-Tree index is primarily based on the B+-Tree, it embodies the primary structure of the B+-Tree. The index operation adds compression and restoration operations to the primary operation when performing insertion, deletion, and search on the tree node, and the function to perform subsequent writing operations from the NAND flash memory is added to the to-be-saved node. For the algorithm, the F-Tree algorithm is applied[6].

Struct Internal_Node {
Short isLeaf: // 0: intermediate node
Short num_of_children; //fan-out Node
*pointer_to_children[MAX1]; // array of pointer to subnode
Key key/[MAX1-1];
};
	Struct Leaf_Node {
	Short isLeaf; // 1 : leaf node
	Short num_of_object ; //fan-out
	Node *pointer_to_object[MAX2];
	// array of pointer to sub-object
	Key key[MAX2];
	Struct Leaf-node *next;
	// pointer to next leaf node
	};

By implementing this algorithm it will reduce the storage space through compression and perform the primary writing at the beginning, then when modifying the tree it will perform the secondary writing at the end and in doing so can reduce the number of erasure operations by half and heighten the tree's overall processing performance, constituting an outstanding tree index.

4 Conclusion

The B-Tree structure that has recently been widely used as an NAND flash memory base index storage technique has brought about memory performance degradation and shortening of lifetime due to frequent writing operations when inserting and deleting nodes. As a result, it has a disadvantage of slow recovery in the case of disorder. In this study the use of a method that has the ability to build an index using an F-Tree and can simultaneously perform failure recovery has been proposed as a method to improve this kind of issue with recovery. The proposed technique builds an F-ISLD using an F-Tree from the NAND flash memory to improve performance through speedy failure recovery.

In the future we hope to prove that the F-ISLD recovery technique shows superior performance to that of existing recovery techniques through the performance evaluation of the F-ISLD(F-Tree Index Segment Log Directory) technique and the existing BISLD(B+-Tree Index Segment Log Directory) technique[6].

References

1. S. E. Corporation, "PM810," Data Sheet, 2011.
2. T.-S. Chung, et al., "A survey of Flash Translation Layer," Journal of Systems Architecture, vol. 55, pp. 332-343, 2009.
3. Chin-Hsien Wu, Li-Pin Chang, and Tei-Wei Kuo, "An Dfficient B-Tree Layer for Flash-Memory Storage Systems", Proc. RTCSA, Tainan, Taiwan, 2003b, pp. 409-430.
4. Chin-Hsien Wu, Li-Pin Chang, and Tei-Wei Kuo, "An Dfficient R-Tree Implementation over Flash-Memory Stirage Systems", Proc. Lf ACM CIS'03, New Orleans, Louisiana, USA, November 7-8, 2003a, pp. 17-24.
5. Kyung-Chang Kim and Suk-Woo Yun, MR-Tree: A cashe-conscious main memory spatial index structure for mobile GIS, Web and wireless geo-graphic information systems, The 4th international workshop (W2GIS 2004), pp. 167-180, 2004.
6. SiWoo-Byun, "F-Tree : Flash Memory based Indexing Scheme for Portable Information Devices", Journal of Information Technology Applications & Management, Vol, 13, Issue 4, pp.257-271, 2009

