

Available online at <http://www.mecs-press.net/ijwmt>

# Design of NAND FLASH File System Based on Loss of Balance Algorithm<sup>①</sup>

Jinwu Ju<sup>a,\*1</sup>, Lecai Cai<sup>a,\*2</sup>

<sup>a</sup> School of computer science, Sichuan University of Science & Engineering, Zigong, Sichuan, China

---

## Abstract

NAND FLASH is a commonly large capacity memory, which used in embedded systems. It often used to store the operating system kernel and file system. NAND FLASH memory has a limited number of block erase feature. Built file system in NAND FLASH, loss of balance method should be adopted. A balanced system of block erase operation, can extend the life of NAND FLASH and improve overall system reliability. The paper analyzes the characteristics of NAND FLASH work, presents a loss of balance algorithm which used in NAND FLASH memory, and given out the implementation of the algorithm design method.

**Index Terms:** NAND FLASH Memory; Embedded System; File System; Loss of Balance

© 2011 Published by MECS Publisher. Selection and/or peer review under responsibility of the Research Association of Modern Education and Computer Science

---

## 1. Introduction

In embedded systems, lead program, operating system kernel, file system is often stored in NAND FLASH memory. NAND FLASH memory is the mainstream technology of high-capacity nonvolatile memory, which with high capacity, low cost, fast erase, pin less and so on. NAND FLASH based on page realize data access operations. The block should be erased before the write operation, which contains the page, but the frequency of erase block are limited, it possibly still appear bad blocks in use process. Accordingly, FAT and NTFS can't be used in NAND FLASH which is based on RAM. In order to improve the whole NAND FLASH memory using life and reducing appearing of bad blocks in using, the file system needs to use loss equilibrium algorithm.[1] In order to prolong the life of NAND FLASH memory, block erase operation should be scattered throughout the storage, to avoid operate part blocks frequently.

---

<sup>①</sup> Fund projects: Science and Technology Bureau of Zigong, Sichuan (10N19), Sichuan province teaches hall youth fund projects(09ZB128).

\* Corresponding author:

E-mail address:<sup>\*1</sup> [jjwmail@163.com](mailto:jjwmail@163.com); <sup>\*2</sup> [clc@163.com](mailto:clc@163.com)

## 2. The Characteristics of NAND FLASH

The way to visit NAND FLASH is different from NOR FLASH. It is not based on bytes, but with the page as the basic unit. Small page NAND FLASH per page is 528B, including 512B data section and 16B spare area (OOB area). Large page NAND FLASH per page 2112B, including 2048B data section and 64B spare area. NAND FLASH write data should be preceded by the erase, erasure with block as a single unit, erasing time for 2ms, blocks constituted by multiple pages. Every block of small page NAND FLASH including 32 pages, totally 16KB, and every block of large page [2] NAND FLASH including 64 pages, totally 128KB. Different from NOR FLASH, NAND FLASH's erasure frequency are limited. Generally, the frequency are 10 million, far less than 100 million frequency of NOR FLASH. It will shorten the storage life if we erase operation on the block frequently. Meanwhile, it only guarantee the reliability of the first piece in memory before factory, the rest of storage may be still has bad blocks, in use process bad blocks may random increase. In addition, NAND FLASH still exist a flip phenomenon, namely data from 1 variable 0, or from 0 change 1, should to use ECC algorithms for a flip detection and recovery.

## 3. Embedded File System

File system commonly used by embedded system include CRAMFS、JFFS、YAFFS. The CRAMFS file system is a kind of read-only systems.[3] In the run-time, the part of CRAMFS required by the system will be unzipped from FLASH into RAM and the file system itself cannot be changed. If FLASH memory has some error block, the system will can't work. JFFS file system is a kind of log file system, which storage the metadata and data itself in the form of nodes in FLASH. The system adopts the CRC error parity when some error block appeared in the FLASH memory. JFFS file system will spend a lot of time to calculation the node's CRC checking information, then choosing normal blocks. JFFS file system is better suited for the NOR FLASH as storage devices.

YAFFS is special designed for NAND FLASH, using OOB area to store metadata. The system has the garbage collection to balance block erasing operation based on loss balance algorithm, but the loss balance algorithm only considers "hot" data. So the erasure operation cannot completely scattered throughout the FLASH. The file system can be written and modified, widely be used in the embedded system. YAFFS file system is applicable to small page NAND FLASH, while YAFFS2 is upgrade of YAFFS, can be used for small page NAND FLASH, also suitable for large pages NAND FLASH. Along with the embedded system equipment storage capacity of rapid growth, and a data transfer rate higher YAFFS2 will get more extensive application.

Table 1. The structure of YAFFS2 spare

NAME	BYTES	EXPLANATION
Block State	1	Block state , normal for 0xFF, otherwise for bad blocks
ChunkID	4	File index description, used to describe the current page's index in its files. If 0, explain this page for the file header, not, that is the document's data page.
ObjectID	4	Object ID number, used in a file system identify files.
nBytes	2	the page valid data bytes. On the last page of the documents, nBytes ≤ 2084, others for 2048.
Block Sequence	4	recording Block allocate sequence. his value in garbage collection was used to judge whether the block should be recycled.
Tags ECC	3	the spare's ECC calibrate data.
Data ECC	24	the data's ECC calibrate data.

YAFFS2 file system is designed for large page NAND FLASH. which page could capacity 2112B (64B spare 2048B data +64B spare area).The file system organized files into the corresponding page, page data section storage file data, spare area storage describe files related information, based on the data backup district to realize bad block management and error detection.[4] When system loading and scanning entire file system, it only needs read 64B spare area to construct the integral structure of file system, while shorting the system loading time. The spare structure is shown in table 1.

#### 4. Transparent Garbage Collection

Due to the differences between write operation mode of NAND FLASH and write operation mode of RAM, the block should be erased first where you want to write new data. Consequently file system based on NAND FLASH update file or save file not by write-covered. It means the data isn't written to the page with data existing but to the new blank data pages ,and then mark the old data pages deleted [5].However the block which the old data pages are located in won't erased at once .System will start garbage collection when the blank data pages are very few .It will calculate the number of the pages which have been deleted and should be retrieved based on loss equilibrium algorithm ,and delete the block to increase the blank data pages .The number of the invalid pages in the block how dirty is it, referred as algorithm selection guidance .These pages will move into the new block upgrading file information and erase the block which has been chosen if the data pages in the block which is chosen are still being used.

YAFFS file system would trigger garbage collection algorithms in each written operation, and produce large impact on system performance. In order to improve system performance, this design put garbage collection algorithm into kernel thread. When the system is idle, it run recovery algorithm and loss equalization algorithm, computing how dirty in the system, and complete the task of erasing old data block. Based on this way, overcoming the disadvantages of YAFFS file system, task of wiping NAND FLASH block was assigned to several blocks, and thus extend service life of NAND FLASH.

#### 5. Loss of Balance Algorithm Design

YAFFS file system uses Loss Balance algorithm which is based on greedy algorithm , retrieve the block which has the least effective pages. Recovery algorithm as shown in Fig.1.

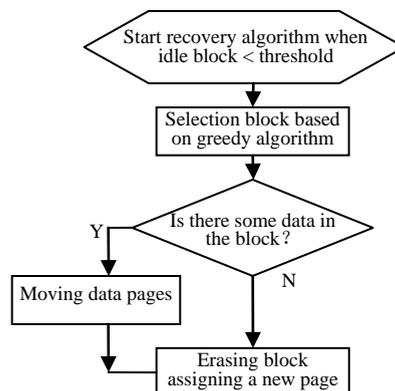


Fig. 1. Loss balance algorithm of YAFFS

When the system is writing operation, it will run recovery algorithm .Algorithm will first count the idle blocks in the system. When it is below the threshold and recovery system startup, use the greedy algorithm to choose the "dirtiest" block. Effective pages in the block will move to a new block, and then erase the block selected, and write data from the block allocation page .Because only the pages updated will identified as invalid pages ,this algorithm is essentially a dynamic Loss Balance algorithm . The scope of recovery block is limited to the blocks which is to be updated or idle block, because algorithm will not work for the block which never or seldom update. In an embedded system, most of the data stored on the device seldom update, which makes loss balance algorithm of YAFFS file system will not perform within the scope of the entire NAND FLASH. When the free space of system is very few ,this real imbalance will become more obvious and can't achieve effective balance of block erasing operation.

### 5.1. The design of loss balance algorithm

The design of loss balance algorithm which was presented by this paper is not only to operate the frequently updated blocks and free blocks, but also for the static load. It namely never update or seldom update block, also be included in the scope of recovery. The new algorithm make the system during garbage collection, not only according to block's degree of the "dirty", but also reference block erasing times to choose the erased blocks. The loss balance algorithm plied to the whole NAND FLASH, can greatly improve the loss balance's effect, better to extend NAND FLASH using life.

Hypothesis NAND FLASH total has N block ,the erasing times of n block is  $EC_n$ , the loss balance algorithm's goal is to make each block erasing times same, even if the variance of each block erasing number as small as possible.

$$D(EC) = \frac{1}{n} \sum_{n=1}^N (EC_n - \bar{EC})^2$$

Set  $MAX\{EC_n\}$  for the maximum, and  $MIN\{EC_n\}$  for the minimum in all blocks's erasing times, meanwhile set action triggers threshold for V, set the triggering conditions for:

$$MAX\{EC_n\} - MIN\{EC_n\} \geq V$$

When the triggering conditions was satisfying, based on the basis of the erase number's erasure greedy algorithm to choice the erased blocks. When the triggering conditions is not met, it shows that each block's erasing times in system is little differences, based on the basis of block's "dirty" degree's greedy algorithm to choice the erased blocks. The threshold of V's choicing influence the performance of the loss balance algorithm. If select the higher threshold V, system in favour of based on block's "dirty" degrees greedy algorithm, and the smaller threshold V, system in favour of according to erasing times of greedy algorithm.[6] Due to the static block's effective data is more and page copy workload is big, it influence on performance of system greatly. However, NAND FLASH block's erasure balance better, and FLASH's work life is longer. In practical application, according to the system's total load degree to choose threshold value of V. The improved loss balance algorithm as Fig 2 shows.

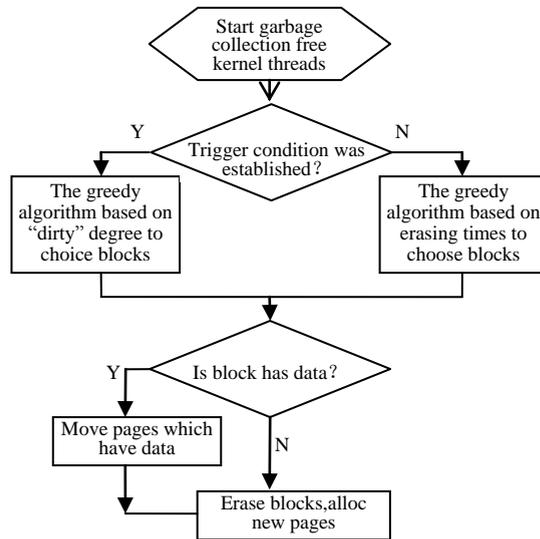


Fig. 2. The improved loss balance algorithm

## 5.2. The implementation of loss balance algorithm

### 1) Data structure changes

This paper designed a new algorithm implement in YAFFS2 file system. YAFFS2 file system defines data structure which was `yaffs_BlockInfo`, which used to record block's related information, including block's effective pages, blocks's the serial number, block state, etc. It adds new members `block_erase_count` in this structure to record the block's erasing times, and considering the block's erasing number limited, setting the member for 4 bytes. The data also record to page's spare areas. the data structure of `Yaffs_ExtendedTags` is layout structure which is in page spare area of block, increasing new member which is `block_erase_count`. These two data structures are defined in a `Yaffs_guts.h`.

### 2) Function changes

Modifying the function that `yaffs_BlockBecameDirty` come first . When recovery algorithm select the blocks erased and move the effective pages on the blocks to other blocks, it will erase the block and at the same time call this function .It also will add member which operate `block_erase_count` and record the code of erasing number. File system will be modified to initialize scanning function `yaffs_Scan` and block information reading function `yaffs_GetBlockInfo` .Read the member of `block_erase_count` which was recorded in page of spare area to block state structure `yaffs_BlockInfo` . In `yaffs_ExtendedTags`,it will add code in initial scanning function `yaffs_Scan`, create a kernel threads,and set its thread priority as low, thus the system will run when it is free and add the times of calling garbage collect function `yaffs_GarbageCollectBlock` in a thread.

Modify garbage collection function `yaffs_GarbageCollectBlock`, and add code according to Fig.2.

## 6. Conclusion

The new loss balance algorithm was implemented in embedded system which based on the ARM—Linux. This system had been used in distributed greenhouse trellis intelligent control. In the apply process, according to the system load condition, setting appropriate threshold  $V$ , make the system balance between loss balance and the running performance, the new loss balance. New loss balance algorithm is simple, practical, it can improve the file system of YAFFS2's loss balance performance and improve NAND FLASH 's using life. It will be of certain reference value in embedded system designing

## References

- [1] QIN Xiao-kang,XU Hui-min . Design and implementation of NAND Flash storage for embedded device [J]. (in Chinese) The computer engineering and design 2010,31(3):514-517.
- [2] WANG Biao , LUO Xin-zhi ,LUO Zhi-ping. The study of Nand Flash write equilibrium in embedded system [J] (in Chinese). Micro computer information 2008(14):8-9.
- [3] PENG Fu-shi,CAO Yong. On NAND flash log file system initialization [J]. (in Chinese) journal of Zhengzhou institute of light industry ,2007,22(5):67-71
- [4] WEI Feng ,LU Zai-qi,LIU Wei. Realization of YAFFS2 in the Embedded System [J]. (in Chinese) The modern electronic technology, ,2010(8):30-37
- [5] SUN Feng,ZHANG Fu-xin . Research and Improvement of YAFFS File System [J]. (in Chinese) The engineering of computer 2008,34(5):257-259
- [6] QI Yi-de, CHEN Guo-feng . A Static Wear-leveling Algorithm Based on NAND Flash [J]. (in Chinese) journal of Chongqing institute of technology,2009,23(7):85-88