

文章编号:1673-0062(2017)03-0086-06

关于 NAND 闪存损耗均衡算法的优化

赵 峰,刘博妍,朱 戈

(安徽工业大学 管理科学与工程学院,安徽 马鞍山 243032)

摘 要:为了改善 NAND 闪存不耐擦写的特性,本文对传统的损耗均衡算法(HWL 算法)进行优化,主要包括触发机制的优化和将现有的静态损耗均衡与动态损耗均衡策略相结合的优化策略.最后根据需求设计了评估损耗均衡算法效果的测试实验.结果表明:与当前算法相比,本优化算法展现了良好的磨损均衡效果,解决了 NAND 闪存不耐擦写的缺陷.

关键词:NAND 闪存;损耗均衡;算法优化

中图分类号:TP333

文献标志码:A

Optimization of NAND Flash Memory Loss Equalization Algorithm

ZHAO Feng, LIU Bo-yan, ZHU Ge

(School of Management Science and Engineering, Anhui University of
Technology, Maanshan, Anhui 243032, China)

Abstract:To improve the characteristics of no resistance to erase NAND flash memory. In this paper, the traditional loss equalization algorithm (HWL algorithm) is optimized. It mainly includes the optimization of the triggering mechanism and the combination of the existing static loss balancing strategy and the dynamic loss balancing strategy. At last, a test experiment is designed to evaluate the effect of the loss equalization algorithm. The results show that compared with the current algorithm, a good wear and balance effect is achieved, which solves the defect of being not resistant to erase NAND flash memory.

key words:NAND flash memory; loss equalization; algorithm optimization

0 引 言

现在越来越多的笔记本电脑、智能手机、固态

硬盘等新型电子设备开始装备 NAND 闪存,凭借特有的存储方式和超高的稳定性、可靠性,NAND 闪存得到越来越多厂商的青睐.配备 NAND 闪存的固

收稿日期:2017-09-16

基金项目:安徽省重点研究与开发计划面上科技攻关项目(1704a0902033)

作者简介:赵 峰(1977-),男,副教授,博士,主要从事于数据挖掘的研究.E-mail:1779236178@qq.com

态硬盘也凭借诸多优势逐渐取代传统机械硬盘^[1]。

采用 NAND 闪存作为存储装置的固态硬盘没有采用像机械硬盘那样的复杂精密结构,读取数据不需要寻道时间,并且也有独特的写入方式,所以能够获得比较快的读取和写入速度。并且由于 NAND 闪存体积小,接口应用广泛,适应各种结构形式,所以可以被手机、笔记本电脑等精密设备采用^[2]。然而 NAND 闪存存储器使用电子管来存储,重复写入擦除数据极易导致损坏,其物理块仅能承受几千次的擦写操作。要想广泛的应用 NAND 闪存就需要采用合适的平均每个物理块磨损的机制,也就是损耗均衡机制,以此才能延长闪存的使用寿命。

在 1994 年,PCMCIA(个人计算机内存卡国际协会)提出了 Flash 转换层(FTL)的概念,FAT 文件系统用于 NAND Flash 时,是由控制器通过管理 NAND Flash 建立一个逻辑的 FAT 文件系统,供上层应用调用,同时它也可以高效的执行损耗均衡^[3]。实现闪存损耗均衡的算法按照类型分为:动态损耗均衡和静态损耗均衡^[4]。Ban 等^[3]针对静态损耗均衡算法的触发机制进行了研究,并提出两种触发机制:确定性触发机制和随机性触发机制。Yared Hailu Gudeta 等^[5]提出了基于概率的静态损耗平均算法。在每个状态下,磨损平均分布使用标准偏差来计算,以确定其是否超过阈。如果它超过阈值,则在所有块中维持损耗平衡通过将热块与冷块交换在闪存中。Lee 等^[6]在静态损耗均衡算法上进行了研究,该算法会在系统初始化时将闪存存储空间划分成若干组,从开始第一次更新之后就记录每一个组的擦写次数。到达损耗均衡的触发条件时统计高于平均擦写次数的组及组内的物理块,然后将这些块的数据搬运至空块,再更新地址映像表。

邢春波^[7]提出一种混合损耗均衡算法(HWL),该法也是在系统初始化时将闪存空间分块,每块的擦写次数由单独分配的一个字节的存储空间记录擦写情况,并采用一个数组来维护所有的块的擦写情况。该算法统计每一个系统更新周期内的各块擦写情况,HWL 算法是目前应用最广发的算法之一,本文也是建立在该算法的基础上进行改进而来。

本文根据闪存的物理块在使用过程中擦写的次数,对闪存数据块进行分类并给出相应的判断标准,同时引入与损耗均衡算法密切相关的阈值的概念;针对损耗均衡算法不能随着存储需求作调整的缺陷,给出根据数据块擦写次数的标准差动态调整阈值的算法;结合传统的动态损耗均衡

算法和静态损耗均衡算法,给出损耗均衡算法的评价标准,将静态损耗均衡算法和动态损耗均衡算法两者结合设计出新的均衡损耗算法;根据需求设计了评估损耗均衡算法效果的测试实验,对损耗均衡算法的效果进行了实验测试。实验结果表明本文提出的改进算法对减小闪存存储器擦写的不均衡性有很好的效果。

1 NAND 闪存损耗均衡机制及策略

1.1 NAND 闪存的工作原理

NAND 是一种计算机闪存设备。随着人们持续不断追求功耗更低、重量更轻和性能更佳的产品,这证明了 NAND 极具吸引力。闪存存储器按照存储单元的存储控制方式可分为单层单元 SLC 和多层单元 MLC 控制方式。单层单元 SLC 控制方式是依靠 MOS 管叠栅上不同的电荷量来区分 0 与 1 两种状态,多个 MOS 管组成叠栅成为一个存储单元。相对应的多层单元 MLC 控制方式由单层单元 SLC 控制方式改进而来,它的一个叠栅可以识别四种存储状态:00、01、10、11。但同时由于该种 MOS 管需要去识别多种电荷量,然而这会造成物理性能不稳定,导致这种闪存的寿命较少。正是因为这种问题,损耗均衡机制的重要性被进一步提高,为了保证多层单元 MLC 控制方式闪存的使用寿命,必须在文件系统中使用损耗均衡和机制。

1.2 损耗均衡评价指标

损耗均衡(wear leveling)^[8]是指用来延长固态存储设备使用寿命的过程。闪存损耗均衡技术的设计目标是:在尽可能减少闪存中所有块的总擦除次数的前提下,尽可能使闪存中的每个块擦除次数趋向一致或在一定范围内。根据统计学的知识我们可以知道平均值描述了一个样本的平均水平,标准差能反映一个数据集的离散程度。根据这个设计目标提出衡量闪存损耗均衡度的两个指标:平均擦除次数 $\bar{E}$ 、擦除次数标准偏差 Deviation,简称为 Dev。

$$\bar{E} = \frac{1}{n} \sum_{i=0}^n E(i) \quad (1)$$

$$Dev = \sqrt{\frac{\sum_{i=0}^n (E(i) - \bar{E})^2}{n - 1}} \quad (2)$$

公式(1)、(2)中 $E(i)$ 表示第 i 块的擦除次数, n 为闪存的总块数。

综合以上因素可以知道,一个较好的闪存系统损耗均衡策略可以使热数据和冷数据的迁移次数

尽可能多,同时总的物理块擦除次数的平均值越小,这样闪存的使用寿命大大延长.另外物理块的擦除次数的标准差越小,表明每一个物理块的擦出次数偏离平均擦除次数较少,损耗均衡机制的执行效果较好.相反,损耗均衡机制的执行效果较差.

1.3 损耗均衡策略

1) 动态损耗均衡策略

动态损耗均衡算法^[9]主要原理是:先将所有的闪存物理块链接到一个动态维护的表格,在表格中将各物理块的擦除次数从大到小进行排序.当数据更新时,要求数据写入到表格最后的物理块,也就是擦除次数最小的物理块,然后再按照擦写次数进行排序.按照这样的方法数据永远都是写入到擦除次数最小的物理块,使得闪存的各块均得到同样的磨损状况.

2) 静态损耗均衡策略

静态损耗均衡算法在动态损耗均衡算法的基础上改进而来,考虑到动态损耗均衡算法中一直优先将数据写入到擦出次数较少的物理块中,原先储备的更新较少的冷数据就会一直占据该物理块,得不到释放.所以静态损耗均衡算法考虑了冷数据的搬运,使得整个闪存存储空间都在不断的进行均衡磨损.静态损耗均衡算法将焦点放在冷数据上,预先设置擦写次数的阈值,采用遍历各块的方法统计得出更新频率低于阈值的方法,筛选出冷数据块,然后将数据搬迁至擦出次数较多的物理块中.用同样的方法筛选出热数据块,将热数据块搬迁至擦除次数较少的物理块中.通过这种方法,闪存中的各物理块的擦除次数比较均衡,并且随着静态均衡机制的不断触发,热数据块的擦出次数增长十分缓慢,冷数据块的擦出次数有一定的提高,从而是十分有效的损耗均衡算法.

2 静态均衡算法设计

本文的静态损耗均衡算法是将现有的动态均衡算法与静态算法相结合,设计出一种更加完善的算法流程.静态损耗均衡算法的设计基本原理在前文介绍过,为此定义擦除次数最多为 $N_{\max}$,擦除次数最少为 $N_{\min}$,两者差值为 Th ,则需要研究的问题^[10]的表述为:

$$\min \left(\bar{E} = \frac{1}{n} \sum_{i=0}^n E(i) \right) \text{ 或 } \min_{Dev} = \sqrt{\frac{\sum_{i=0}^n (E(i) - \bar{E})^2}{n-1}} \quad (3)$$

$$s.t. (N_{\max} - N_{\min}) < Th \quad (4)$$

这里式(4)中的 Th 就是本文所研究的静态损耗均衡算法的核心,即要选择恰当的阈值 Th ,使得物理块的擦除次数上下限相差不会太大.同时又要满足尽可能小的平均值和标准差.

2.1 触发条件的优化

确定性触发的难点在于难以找到合适的系统更新次数,随机性触发只是用于系统更新频率较高的情况^[11].

为此本文设计了一种可以动态选择触发机制的算法,将确定性触发和随机性触发相结合,该算法会统计系统的更新频率.如果更新频率较高则选择随机触发条件,如果更新频率较低则选择确定性触发条件.参考损耗均衡机制的评价标准,选择系统更新次数的均值当做判别条件.流程图如图1所示.

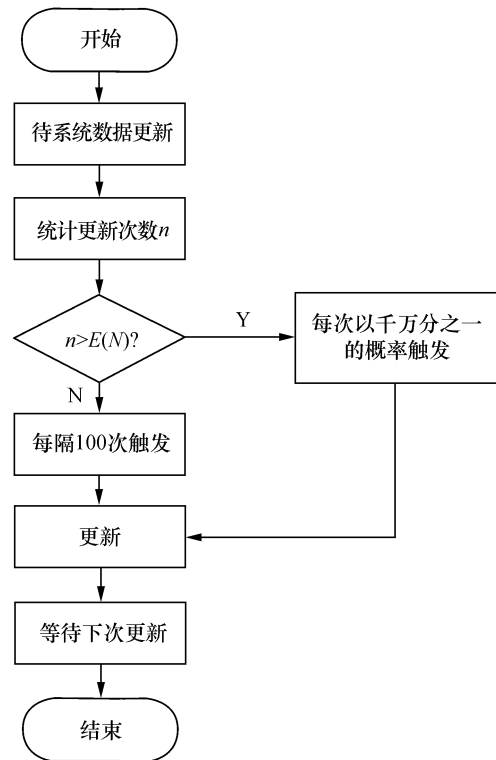


图1 优化的触发条件流程图

Fig.1 Flow chart of optimal trigger condition

2.2 阈值确定数据块的状态

数据块的磨损状态由该块的擦写次数在数据擦写中所占的比例决定,即

$$w_i = \frac{e_i}{a_0} \times 100\% \quad (5)$$

式中 w_i 表示该块的磨损程度, e_i 表示该块在本次系统更新中的擦写次数, a_0 是本次系统更新总的擦写次数。

在磨损均衡过程中维护了一个磨损信息表,表中为每个物理块分配一个字节的存储空间记录磨损状态 w_i 。如果 w_i 达到阈值 w_{high} 就将该块定义为热数据块,同理低于阈值 w_{low} 就将该块定义为冷数据块。

2.3 算法流程

由以上结论可知,要想解决损耗均衡的问题,就必须在尽可能小的影响性能的条件下使擦写任务均匀的分布在每个数据块^[12]。本文在解决这个问题时,通过对以往动态和静态算法的研究下,就此设计了动态静态算法相结合的 NAND FLASH 闪存损耗均衡的算法。

一个完整的静态损耗均衡流程为:在系统更

新操作每过一定的擦写次数 n 时启动静态损耗均衡机制,首先遍历所有参与擦写的数据块,计算出数据块的磨损状态。按照磨损状态从低到高排序,达到阈值 w_{high} 就将该块定义为热数据块,低于阈值 w_{low} 就定义为冷数据块。然后将热数据块中的有效数据搬迁至新的空块或者磨损状态 w_i 最低块,将冷数据块搬迁到原先的最热数据块,重排后依次进行,将搬迁后的数据块重新链接到文件系统。本次损耗均衡结束后计算没有超出阈值的数据块的擦除次数标准偏差,如果过高则标记超出平均擦除次数一定值的块为热数据块,并参与下一次系统更新。这样热数据块就会不断被搬迁新的物理块,不会只在部分块内擦写,有效降低了闪存数据块的磨损。算法流程如图 2, 图 3 所示。

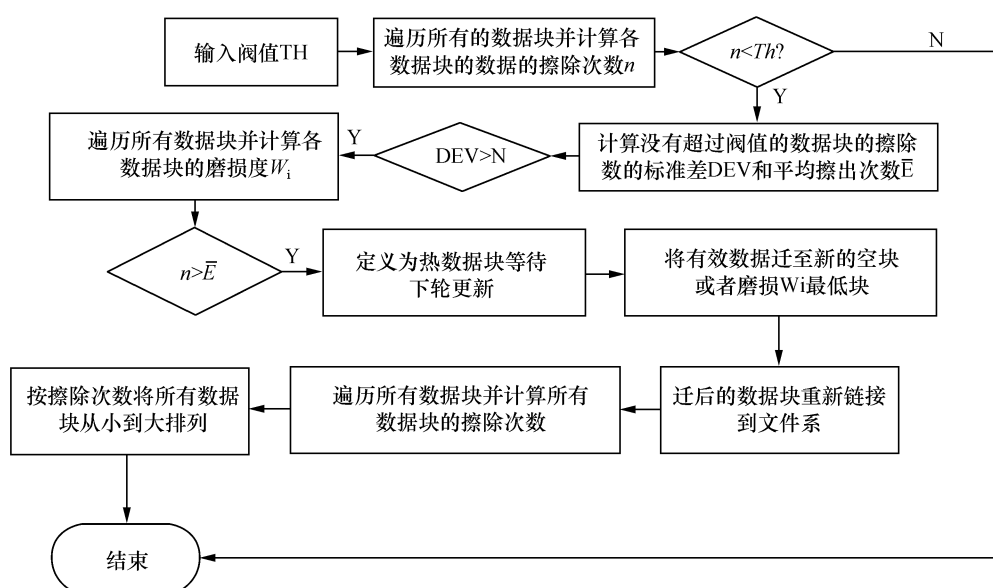


图 2 完整的均衡损耗流程图一

Fig.2 Complete balanced loss flow diagram I

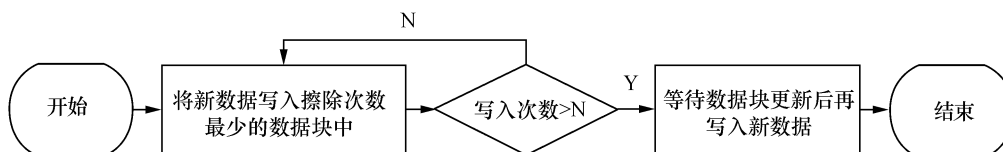


图 3 完整的均衡损耗流程图二

Fig.3 Complete balanced loss flow diagram II

3 仿真测试

3.1 实验平台搭建

为了测试本文设计的损耗均衡算法的效果,

借助 SSD Sim 搭建测试平台。它可以在普通的计算机平台上模拟 NAND 闪存的存储状况,用户可以在 SSD Sim 软件中设置存储容量、存储信道、闪存块大小等参数,根据用户需求可以在没有闪存

硬件的支持下模拟 NAND 闪存的使用情况。

为了直观的对比损耗均衡算法的效果,本文设计了两个 NAND 闪存,每个闪存分配 4 个闪存芯片,它们之间以并行方式连接,其中闪存 1 采用传统的静态损耗均衡算法^[13](HWL 算法);闪存 2 采用本文改进后的损耗均衡算法,两个闪存都写入同样的 1×10^6 次数据量写入请求.两个闪存的其他详细配置如表 1 所示.

表 1 闪存配置表

Table 1 Flash configuration table

性能	NAND 闪存
容量/G	16
块容量/MB	2
页容量/KB	8
写入热数据比例/%	20

3.2 测试内容及结果分析

本节比较了分别采用改进前和改进后的损耗均衡算法的效果.主要通过对比两种磨损状态阈值 w ,系统更新次数 n 和静态损耗周期 M 三个因素对闪存存储的磨损均衡性能的影响,评价的指标用一个周期内的擦除次数标准偏差来描述.

1) 为研究磨损状态阈值上限对闪存存储的磨损均衡性能的影响,设置 HWL 算法下 $w = 1.2\%$ 和 $w = 0.8\%$ 两种情况下记录随着写入请求的增加,物理块擦除次数标准差的变化情况.

从图 4、图 5 可以看出相比 HWL 算法,改进后的静态损耗均衡算法的损耗均衡效果更好一些.随着写入请求不断增加,两者的标准差都是稳定增长.这说明随着写入请求的不断增多在静态损耗均衡机制的控制下标准差的增长趋于稳定,趋于稳定时改进后的擦写次数的标准差比 HWL 算法小约 16.4%,这说明改进后的阈值使得损耗均衡效果更好.

2) 通过物理块擦写次数的标准差变化曲线,比较改进之前和改进之后的静态损耗均衡的周期对静态损耗均衡效果的影响.HWL 算法是采用固定周期,改进后的算法采用变周期.从图中可以明显看出改进后的静态损耗均衡算法效果显著:从写入请求达到 2.2×10^5 次时,两者的擦写次数标准差均达到最大.但是随后 HWL 算法的擦写次数标准差继续增长,直到写入请求达到 7.4×10^5 时才有下降的趋势;然而改进后的擦写次数标准差开始下降,并最终稳定到 10×10^5 左右.由于改进后

的算法采用了动态的周期选择的方法,相比 HWL 的固定周期能更好地适应闪存静态损耗均衡的需求.

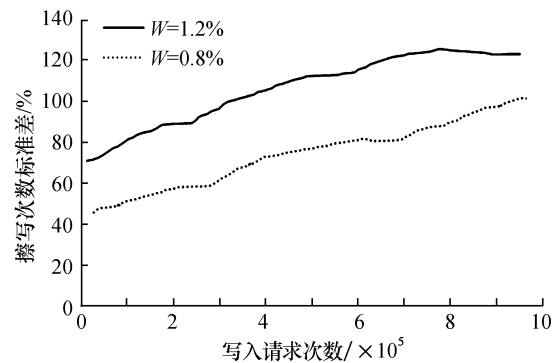


图 4 磨损状态阈值上限对闪存存储的磨损均衡性能的影响

Fig.4 Effect of wear state threshold on wear equalization performance of flash

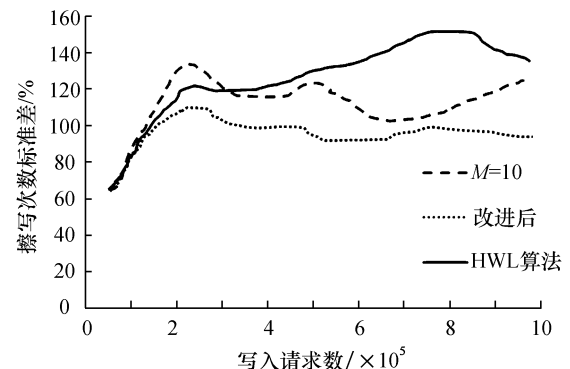


图 5 静态损耗均衡的周期对静态损耗均衡效果影响的对比分析图

Fig.5 Comparison and analysis of the influence of static loss equalization cycle on static loss equalization

同时本文选取了另外一个 $M = 10$ 的变化曲线对比观察,发现即使是将静态损耗及均衡周期设置的比较短,损耗均衡的效果仍然不理想.

4 结 论

对目前较流行的 HWL 损耗均衡算法进行优化.重新定义了数据块的冷热属性,并以高低阈值作为热数据和冷数据的判定条件;对触发机制进行优化和将现有的静态损耗均衡策略与动态损耗均衡策略相结合的优化策略.通过 SSD Sim 软件完成了对比测试,实验证明改进后的损耗均衡算

法有效的降低了物理块擦除次数的标准差,使得损耗均衡效果得到提高。

参考文献:

- [1] 温朝建,严华,文超建,等.Yaffs2 文件系统中对 NAND Flash 磨损均衡的改进[J].电子技术应用,2015,41(6):16-19.
- [2] 江超,周传华,赵伟,等.基于混合云存储多频道视频直播系统方案设计[J].南华大学学报(自然科学版),2017,31(2):62-66.
- [3] BAN A.Wear leveling of static areas in flash memory: US6732221[P].2004-05-04.
- [4] 刘虎生.一种固态硬盘混合损耗均衡算法研究[D].长沙:国防科学技术大学,2012.
- [5] GUDETA Y H, KWON S J, CHO E S, et al. Probability-based static wear-leveling algorithm for block and hybrid-mapping NAND flash memory[J]. Design Automation for Embedded Systems, 2012, 16(4): 241-264.
- [6] LEE C C, CHEN B W, CHOU S K. System and method for managing blocks in flash memory: US20050204187[P]. 2005-09-15.
- [7] 邢春波.闪存磨损均衡算法研究[D].杭州:浙江工业大学,2008.
- [8] 刘智朋,罗洪元,阳小珊.闪存循环位图的损耗均衡机制研究[J].计算机工程与设计,2013,34(2):523-528.
- [9] 赵鹏,白石.基于随机游走的大容量固态硬盘磨损均衡算法[J].计算机学报,2012,35(5):972-978.
- [10] 罗桂娥,康霞.固态硬盘性能优化研究与实现[J].计算机工程与应用,2015,51(1):43-48.
- [11] 丁万霞.高速大容量 NAND FLASH 存储系统[J].数字技术与应用,2016(7):158-158.
- [12] 汤瑞.一种基于 NADA Flash 固态硬盘的坏块管理方法[J].电子科技,2014,27(8):40-42.
- [13] 陈金忠,姚念民,蔡绍滨.基于 NADA 闪存的高性能和可靠的 PRAID-6[J].电子学报,2015,43(6):1211-1217.