文章编号:1671-4598(2018)05-0181-04 DOI:10.16526/j.cnki.11—4762/tp.2018.05.045 中图分类号:TP3 文献标识码:A

Android 存储系统中 NAND 闪存性能的改进

陈 旭,严 华
(四川大学 电子信息学院,成都 610065)

摘要:针对 Android 存储系统在闪存管理上存在较差的磨损均衡效果和较高的垃圾回收额外开销的缺陷,引入冷热数据分离策略,将文件按照不同热度写入对应热度的物理存储单元,同时改进垃圾回收策略,以达到良好的磨损均衡效果并减少垃圾回收额外开销;基于 Android 平台的实验结果表明,改进后的策略在有效减少 NAND 闪存垃圾回收额外开销的同时,还能有效改善其磨损均衡效果。

关键词:安卓存储系统;磨损均衡;垃圾回收;NAND 闪存

Improvements on Performance of NAND Flash in Android Memory System

Chen Xu, Yan Hua
(College of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China)

Abstract: To solve the problem of Android memory system in wear leveling and garbage collection, the policy of cold/hot data separation is adopted to write files to the corresponding physical memory unit according to different heat. Then, the policy of garbage collection is improved to get better performance of wear leveling and garbage collection. Experimental results based on Android platform show that the garbage collection overhead for NAND flash is reduced and the degree of wear leveling for NAND flash is improved in Android memory system.

Keywords: Android memory system; wear—leveling; garbage collection; NAND flash

0 引言

在 Android 设备中,存储系统已经超过网络和处理器成为整个设备性能的瓶颈[1]。因此,Android 设备对大量数据的高速存储需要更高效的存储解决方案。NAND 闪存具有存取速度快、体积小、重量低等优点,因而被 Android 设备广泛作为存储介质。

不同于传统存储介质,NAND 闪存在硬件上具有写前擦除的限制,也就是存储区域被写入数据之后无法再次写入新的数据只有该区域被擦除之后才可以写入新的数据,因此需要采用异地更新策略[2]。该策略是将更新数据写入新的存储区域,然后将原来区域的数据置为无效,存储无效数据的区域在没有经过回收之前无法再次写入新的数据。这些无效数据被称为垃圾,无效数据会随着系统的运行不断积累,从而导致存储系统的有效空间不断减少。为了获得更多有效空间,此时需要对垃圾进行回收。垃圾回收包括一系列额外的读、写、擦除操作,因此会增加额外开销。同时,NAND 闪存的写和擦除操作非常耗时,因此存储系统性能也会受到影响[3]。另外,NAND 闪存由多个块组成,每个块的擦除次数是有上限的。例如,Samsung 840 闪存每个块只有 1 000～3 000 次擦除寿命[4]。如果块的擦除次数达到上限,这个块将无法继续使用,同时闪存的 I/O 性能和可靠性也会受到影响,甚至关系到闪存的使用寿命。因此,需要采用磨损均衡策略使闪存的每个块擦除程度尽可能一致,从而保证闪存良好的 I/O 性能和可靠性并延长设备使用寿命[5]。

Android 系统采用 YAFFS2 (Yet Another Flash File System 2) 文件系统来对 NAND 闪存进行管理[6]。YAFFS2 文件系统是根据闪存固有的特性专门开发的文件系统,可以直接对闪存进行管理,并可以充分发挥闪存的存储优点。但是,YAFFS2 的垃圾回收策略对 NAND 闪存进行管理时,存在较差的磨损均衡效果和较高的垃圾回收额外开销的问题。为此,本文引入冷热数据分离策略,同时改进垃圾回收策略,以降低 Android 存储系统的垃圾回收额外开销并改善其磨损均衡效果。

1 Android 存储体系

Android 基于 Linux 内核开发,内核采用针对闪存固有特性而开发的 YAFFS2 文件系统对存储介质 NAND 闪存进行管理。Android 的存储体系结构如图 1 所示。

Android 存储体系主要包括虚拟文件系统 VFS (Virtual File System),YAFFS2 文件系统、MTD 驱动和 NAND Flash 存储设备。其中,VFS 为上层应用程序提供统一的接口,MTD 则为底层的 NAND Flash 提供统一的读写操作接口,以方便 YAFFS2 文件系统对 NAND Flash 存储设备进行操作。

NAND 闪存由有限个物理块 (block) 组成,每个物理块又由相同数量的物理页 (page) 组成。NAND 闪存最小擦除单位是物理块,最小读写单位是物理页[7]。NAND 闪存中的物理页按照状态可以划分为空闲页,有效页,无效页这三类。有效页存放有效数据,空闲页是经过擦除后暂未写入数据的页,有效页上的数据被置为无效后就变为无效页。NAND 闪

收稿日期:2017-09-07; **修回日期**:2017-10-18。

作者简介:陈 旭(1991-),男,河南新密人,硕士研究生,主要从事模式识别,智能控制方向的研究。

严 华(1971-),男,四川渠县人,教授,博士,主要从事模式识别,智能控制方向的研究。

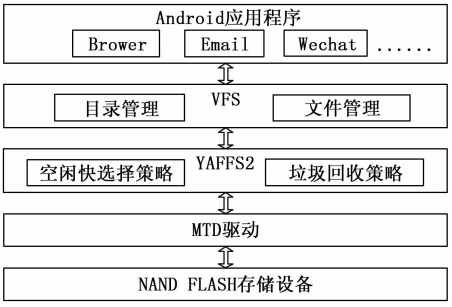

图 1 Android 的存储体系结构图

存中的物理块可以分为两类：空闲块和非空闲块。空闲块上的物理页都是可供写入数据的空闲页，空闲块可以被操作系统分配使用。非空闲块被有效页或者无效页所占据，非空闲块只有在经过垃圾回收后才可以写入新的数据。在 NAND 闪存中，当某个物理页上的数据需要更新时，不是在该页上直接覆写，而是先将更新数据写入空闲页，然后再将原来的有效页置为无效。随着系统的运行，无效页会不断增加，系统有效空间会不断减少。因此，需要启动垃圾回收策略选择回收块进行擦除，擦除后的回收块就变为空闲块，从而达到回收空间的目的。但是，回收块被擦除之前需要将该块上的有效页拷贝到新的空闲块上。

2 Android 存储管理存在的缺陷

在 Android 的存储体系中，系统写入数据时会判断当前分配的物理块上是否存在空闲页。如果当前分配的物理块上存在空闲页就会将更新的数据写入其中，如果当前分配的块上已经写满数据没有空闲页则会跳过当前的物理块，系统会按照块的序号寻找下一个物理块，如此循环顺序的查找，直到找到存在空闲页的物理块为止，在这种分配策略下每个块都有相同的概率被选中为分配块。同时，Android 存储体系采用贪心策略来选择回收块。该策略选择所有脏块（同时存在有效页和无效页的块）中有效页最少的物理块作为回收块，在回收脏块时，可以用较少的额外拷贝次数将有效页的数据拷贝到其它空闲块[8]。在传统的应用领域如传感器网络、数码相机中，由于数据更新操作是均匀分布的，每个物理块被选中做为分配块和回收块的概率相同，可以得到良好的磨损均衡效果，同时可以用较小的垃圾回收额外开销完成垃圾回收，所以贪心策略和顺序分配物理块的块选择策略是简单有效的。

然而，在 Android 系统中，不同文件的操作频率相差很大，因此数据更新通常不是均匀分布的。例如，系统自带的应用程序、配置文件以及核心类库，仅在设备厂商进行远程升级时其中有些数据才会进行更新或者说根本不会被更新，这些典型的很少被更新的数据称为冷数据。又如，在线视频软件中的流媒体视频、图片、音乐，这些数据经常大量地进行更新操作，这些经常被更新的数据称为热数据[9]。

如果某个块上热数据较多，这个块上的数据会被反复更新，块上的空闲页会被写入更新数据，原来的有效页就会被置为无效页，随着块上的无效页不断增加，这样的块就容易被存储系统选中作为回收块，经过擦除之后变为空闲块，重新参与到空闲块的循环分配周期中。当一个块上存储的都是冷数据，冷数据更新较少所以块上的无效页也会很少，那么这样的块根据原有的垃圾回收策略是不会被选为回收块的，也就不会参与到空闲块的循环分配周期中，同时这些块的擦除次数相对于存放热数据的块的擦除次数就会少很多。因此，Android 存储系统中的贪心策略和空闲块选择策略会造成 NAND 闪存物理块的擦除次数呈现两极分化，导致较差的磨损均衡。同时，当更新频率不同的数据在同一个块上存在时，当热数据的快速更新致使这个块被回收时，也将迫使在同一个块上的冷数据被更新，导致垃圾回收额外开销的增加。

综上，Android 存储体系中采用的贪心策略和空闲块选择策略，没有考虑冷热数据不同的更新频率和物理块擦除次数的差异，从而造成了 Android 存储管理的磨损均衡效果较差以及垃圾回收额外开销较高。上述缺陷会使 Android 设备性能、存储可靠性和寿命受到影响。

3 改进策略

针对 Android 存储管理存在的缺陷，采用冷热数据分离的策略进行针对性的改进，对文件进行热度计算，将热数据写入擦除次数少的块，冷数据写入擦除次数多的块。此外，每隔一段时间就对最冷块即擦除次数最少的块进行强制回收。

3.1 逻辑页热度的计算

在 Android 采用的 YAFFS2 文件系统中，文件以逻辑页为基本存储单位，逻辑页分为逻辑地址和数据页两个部分。文件的逻辑地址和闪存的物理地址存在对应关系，通过一张地址映射表来建立准确的映射关系，因此可以将逻辑页作为热度计算的对象，以其更新的时间间隔来表征数据的热度。当文件中的某个逻辑页被修改或者更新时，它的热度 T 可以如式（1）和（2）进行计算：

$$T_{i+1}=\begin{cases}T_i * u, & if \quad 1 \leqslant T_{i+1} \leqslant N_{block}\\ 1, & if \quad T_{i+1}<1\\ N_{block}, & if \quad T_{i+1}>N_{block}\end{cases} \tag{1}$$

$$u=\left(\frac{1}{2}\right)^{\left(\left\lceil\frac{t_{i+1}-t_i}{N_t}\right\rceil-1\right)} \tag{2}$$

T_i 的初始值是 T_{freq}，当 T_i 大于 T_{freq} 时，表示该逻辑页更新操作比较频繁是热数据，反之表示该逻辑页较少得到更新，该逻辑页页存储的是冷数据。$t_{i+1}-t_i$ 表示逻辑页当前更新操作与上一次更新操作的时间间隔。N_{block} 是整个 NAND 闪存的物理块数目，N_t 是调节时间敏感性的参数。

3.2 冷热数据的分离与最冷块的回收

在增加了热度计算机制后，可以对 Android 文件系统采取冷热数据分离的策略，如图 2 所示。当某个物理块被选为回收块时，针对该块中的每一个有效的物理页，按照如下步骤进行冷热分离。

1）计算该有效物理页所对应的逻辑页热度；

2）如果对应逻辑页的热度大于或者等于阈值 T_{freq}，说明该页存储的是热数据，则将该页上的数据拷贝到具有最小擦除次数的空闲块（冷块）；

3）如果对应逻辑页的热度小于阈值 T_{freq}，说明该页存储的是冷数据，则将该页上的数据拷贝到具有最大擦除次数的空闲块（热块）。

4）在逻辑页冷热分离之后，记录逻辑页的更新操作时间，为下次热度计算做准备。

当一个块上同时存在冷数据和热数据时，随着热数据的频繁更新，致使物理块被回收时，冷数据也将被拷贝写入新的空闲块，随着上述过程的进行冷数据将连续多次被拷贝。冷热数据分开存储后，回收脏块时减少了冷数据无意义的拷贝，另外由于同一个块上的页更新频率相近。更新频率相近的页，有较大的概率在同一段时间内被更新，因而也有较大的概率在同一段时间内被置为无效。因此，在该块被回收时只有少量的有效页，甚至没有有效页。这样被贪心策略回收时只需要较少的拷贝次数就能完成块的回收，从而降低垃圾回收额外开销。

另外，Android 系统中存储冷数据的物理块，由于冷数据更新较少或者不更新，块上无效页很少。这些冷数据所在的块很可能不会被贪心策略选作垃圾回收块，那么这些块就不会参与到空闲块的循环分配周期中，这样势必会造成磨损不均衡。为了进一步提升磨损均衡，可以在贪心策略之外，增加最冷块回收策略即每当回收的次数超过指定阈值 G_n 时，就强制选择擦除次数最小且有效页最少的块作为回收块，从而使该块参与到回收中，最终提升整体的磨损均衡效果。

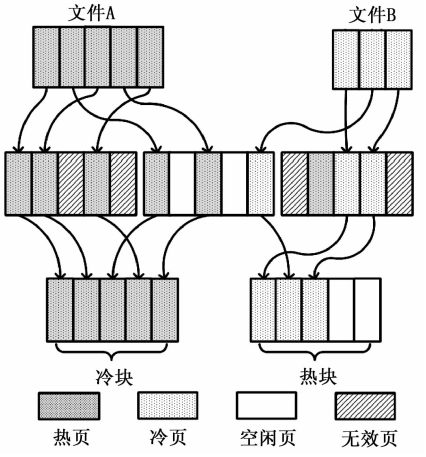

图 2　冷热分离示意图

4　实验测试

4.1　实验环境

实验环境通过在 Ubuntu 12.04 LTS 下使用 Android 模拟器 emulator 22.3.0、Android 2.3.7 版本来搭建，内核使用谷歌基于 Linux 内核定制的 Goldfish 2.6.29，同时将改进后的 YAFFS2 文件系统移植入 Goldfish 内核并重新进行编译。在设备启动后，Android 系统会依次加载编译后的 ramdisk.img、userdata.img、system.img 文件镜像，镜像文件挂载之后 Android 根目录下主要有 system，data，cache 等子目录。system 目录包含了操作系统的内核以及引导程序；data 目录是保存用户数据的区域，包含了用户数据和配置文件的信息，该区域进行擦除，会导致用户数据的丢失；cache 目录是缓存目录，擦除其中的信息，不会影响系统的运行。

测试在 Android 系统的 cache 目录中进行，在 Android 启动时将 cache 目录所对应的 NAND 闪存分区划分为 64 MB，共 512 个物理块。进入 adb shell（Android Debug Bridge shell），执行 adb push 指令将交叉编译后的测试程序传入 Android 的 cache 目录中，使用传入的测试程序随机生成一系列大小在 16 kB～1 024 kB 之间的文件，生成的测试文件大小总和占整个 cache 分区空间的 80%，在文件创建完成后，对其中 15%的文件进行更新，更新次数最多的文件更新次数为 n，剩余测试文件的更新次数依次为 $n/2$，$n/3$，$n/4$，...，1，更新操作满足齐夫分布[10]。在文件更新并触发 Android 垃圾回收机制时分别会调用内核中 yaffs2 文件系统的读函数，写函数，擦除函数，每当调用相关函数一次，对应的计数变量增加 1，在测试程序运行结束后，就可以统计出拷贝次数，擦除次数等信息。相应的实验参数见表 1。

表 1　实验参数

N_{block}	N_p	N_t	T_{freq}	G_n
512	64	50	128	100

4.2　实验结果

实验结果主要对 Android 存储系统改进前后的垃圾回收额外开销和磨损均衡效果两个方面进行对比。垃圾回收额外开销主要考查总的拷贝次数和块擦除次数两个指标，磨损均衡效果则对最大最小擦除次数差值和擦除次数的标准差两个指标进行对比。

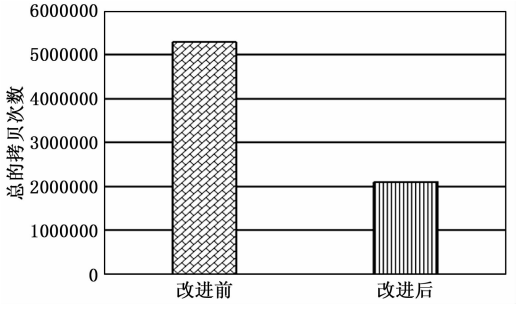

图 3　总的拷贝次数

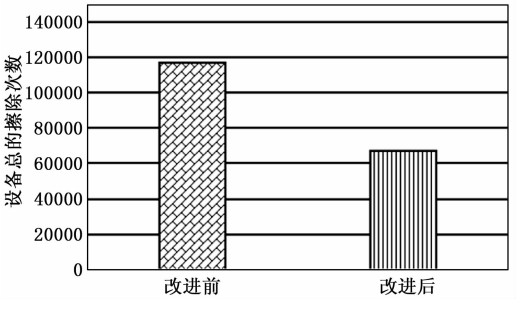

图 4　总的擦除次数

图 3 和图 4 显示的是改进前后总拷贝次数和总擦除次数的测试结果。结果显示，改进后的总拷贝次数和总擦除次数大幅减少。图 5 显示的是闪存中所有块的最大最小擦除次数差值，通过对比可以发现改进后的所有块最大最小擦除次数差值大概为改进前的 1/8。从图 6 中可以看到随着擦除次数的增加，改进前所有块的擦除次数标准差是非收敛的，随着擦除次数的增加而增加。改进后所有块的擦除次数标准差是趋于收敛的，并没有随着擦除次数的不断增加而增加。因此，磨损均衡效果比

改进前有了很大的改善。

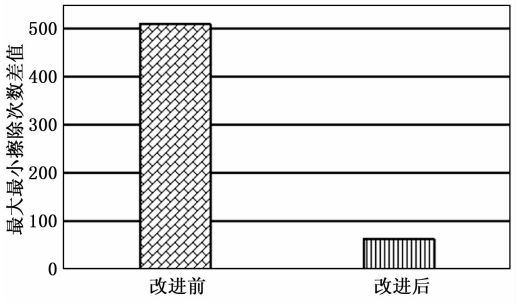

图 5 最大最小擦除次数的差值

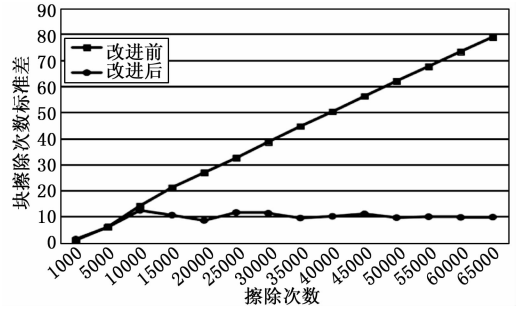

图 6 擦除次数的标准差

改进后的存储管理策略在进行冷热数据分离后，将热度大的逻辑页拷贝到擦除次数少的块中，将热度低的逻辑页拷贝到的擦除次数多的块中，让擦除次数多的物理块尽可能少地被擦除，让擦除次数少的块承担更多的擦除操作，使每个块的擦除次数更加均匀化，同时每隔一段时间强制回收长久得不到更新的块，使其参与到回收与分配的循环中，因而获得了更好的磨损均衡效果。此外，冷热数据的分离，使更新频率相近的页尽可能拷贝到同一个块中，更新频率相近的页有更大的可能在同一段时间内变为无效，从而使该块被回收时具有更少的有效页，用较少的拷贝次数即可完成块的回收，最终减少了总的拷贝次数和擦除次数。

5 结论

在 Android 存储系统采用的贪心策略基础上引入逻辑页热度计算方法，按照热度对逻辑页进行冷热分离，并定期对最冷块进行回收。实验表明，改进后的存储管理策略，垃圾回收额外开销减小，磨损均衡效果得到大幅度提升。实验成果适用于基于 NAND 闪存的 Android 设备，可提升其存储性能和可靠性，延长使用寿命。

参考文献：

[1] Dohee Kim，Eunji Lee，Sungyong Ahn. Hyokyung Bahn Improving the Storage Performance of Smartphones through Journaling in Non－volatile Memory [J]. IEEE Transactions on Consumer Electronics，August 2013，59 (3)：556－561.

[2] Xu Guangxia，Wang Manman，Liu Yanbing. Swap－aware Garbage Collection Algorithm for NAND Flash－based Consumer Electronics [J]. IEEE Transactions on Consumer Electronics，2014，60 (1)：60－65.

[3] 时 正，纪金松，陈香兰，等. 一种基于差分进化的 Flash 文件系统垃圾回收算法 [J]. 电子学报，2011 (2)：280.

[4] Vöttö，Kristian. Samsung SSD 845DC EVO/PRO Performance Preview & Exploring IOPS Consistency [J/OL]. AnandTech，11 June 2017.

[5] Bennett A D，Gorobets S A，Tomlin A，et al. Scheduling of housekeeping operations in flash memory systems [P]. US. 7565478，SanDisk Corporation (Milpitas，CA，US) 2009－7－21.

[6] 曾健平，邵艳洁. Android 系统架构及应用程序开发研究 [J]. 微计算机信息，2011 (9)：1－3.

[7] Chao Sun，Asuka Arakawa and Ken Takeuchi . SEA－SSD：A Stroage Engine Assisted SSD With Application－Coupled Simulation Platform [J]. IEEE Transactions on Circuits and System，January 2015，62 (1)：120－129.

[8] Yan H，Yao Q. An Efficient File－aware Garbage Collection Algorithm for NAND Flash－based Consumer Electronics [J]. IEEE Transactions on Consumer Electronics，2014，60 (4)：623.

[9] Qian Y，Lu J，Xing K. Wear－Leveling Optimization of Android YAFFS2 File System for NAND Based Embedded Devices [J]. Lecture Notes in Computer Science，2014，8491：12－21.

[10] Lin M，Chen S. Efficient and intelligent collection policy for NAND flash based consumer electronics [J]. IEEE Transactions on Consumer Electronics，2013，59 (3)：538－543.

(上接第 180 页)

[4] 董继扬，张军英. 一种简单的椒盐噪声滤波算法 [J]. 计算机工程与应用，2003，20：27－28.

[5] Zhou Wang，David Zhang，Progressive switching median filter for the removal of impulse noise from highly corrupted image [J]. IEEE Trans. On Circuits and Systems－ II：Analog and Digital Signal Processing，1999 ，46 (1)：78－80.

[6] Zhang S，Karim MA. A new impulse detector for switching median filters [J]. IEEE Signal Processing Letters，2002，9 (11)：360－363.

[7] Liu J，Musialski P，Wonka P，et al. Tensor completion for estimating missing values in visual data [A]. IEEE，International Conference on Computer Vision [C]. IEEE，2010：2114－2121.

[8] 刘园园. 快速低秩矩阵与张量恢复的算法研究 [D]. 西安：西安电子科技大学，2013.

[9] Prathap P M J，Brindha G R，Dani W V. A Revised TESOR Algorithm for Perfect Routing [M]. Trends in Computer Science，Engineering and Information Technology. Springer Berlin Heidelberg，2011：355－366.

[10] 石光明，刘丹华，高大化，等. 压缩感知理论及其研究进展 [J]. 电子学报，2009，37 (5)：1070－1081.

[11] 刘 盾，石和平. 基于一种改进的压缩感知重构算法的分析与比较 [J]. 科学技术与工程，2012，12 (21)：5154－5157.

[12] 许光宇，林玉娥，石文兵. 基于局部结构张量的图像三边滤波器 [J]. 计算机工程，2017，43 (4)：269－276. [2017－09－23].

[13] 刘瑜瀚. 基于非负张量分解的医学图像的数据融合方法研究 [D]. 成都：电子科技大学，2015.

[14] 张志伟，马 杰，夏克文，等. 一种应用于高阶数据修复的非负稀疏 Tucker 分解算法 [J]. 光电子·激光，2017，28 (7)：773－779.

[15] 张乐飞，何发智. 基于张量分解的超光谱图像降秩与压缩 [J]. 武汉大学学报 (信息科学版)，2017，42 (2)：193－197.