**Nand Flash管理算法介绍之FTL简介**

和传统磁盘相比，Nand Flash存储设备存储延迟低、功耗低、更高的存储密度、抗震型号更好和噪声低。但是，由于Nand Flash的特性影响(读写擦的单位不一致，每个块有P/E次数限制)，Nand Flash不能直接通过简单的接口转换就拿来使用。所以我们需要在Nand Flash上增加一个管理软件FTL(Flash Translation Layer)进行管理，对外就是一个黑盒子，上层应用可以通过逻辑地址来对这个黑盒子进行访问。

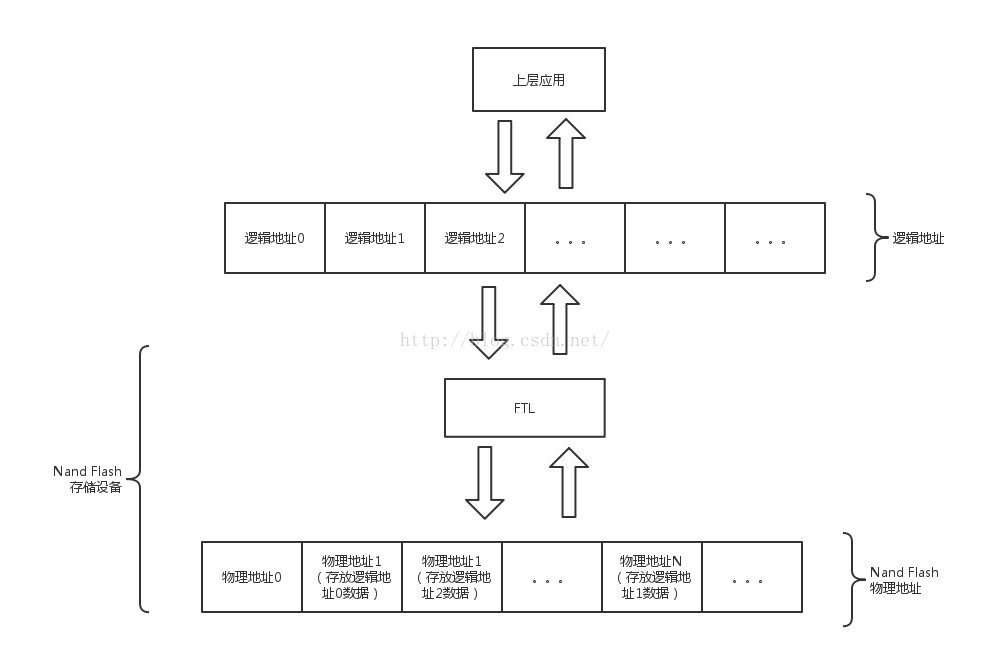


图1

如上图，上层应用通过逻辑地址来访问存储设备，FTL把不同的逻辑地址映射到Nand Flash中的不同位置，FTL简单来说，主要就是映射管理。

固态存储相对于传统磁盘的一些特点：

异处更新(out-of-place Update)。正如之前所说，Nand Flash编程操作只能把存储单元从1变为0，所以在重新编程之前需要进行擦除操作。而且编程以页为单位，擦除以块为单位(一个块包括多个页)。如果使用同处更新(in-place update)，就是把同逻辑地址重复更新到同样的位置上，那么每一次更新，都需要先进行一次擦除操作。由于擦除操作耗费时间和对Flash有损伤，所以一般FTL使用异处更新，把更新的数据映射到一个新的位置上。                                                                                                      如下图，上层应用先写逻辑地址0、1，FTL把数据映射到Nand Flash的物理块0、1页上，然后上层应用又写逻辑地址0，此时物理块0第0页不能重新编程，所以FTL把数据存放在物理块0的第2页上。

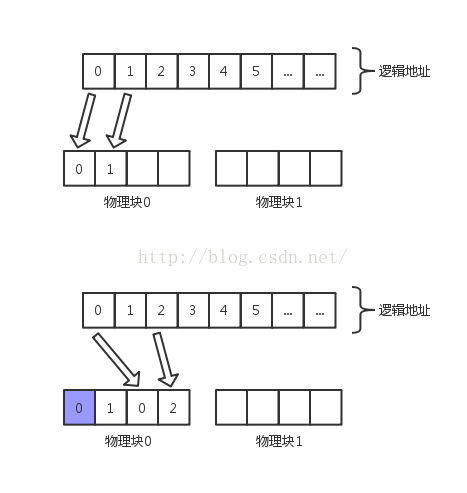


图2

P/E次数有限制。之前有提到，Nand Flash每个块是有擦除次数限制的，在擦除一定次数后，这个块会变得不稳定，编程进去的数据容易出错，甚至会擦除失败。

性能更好。和传统机械硬盘不同，Flash存储是没有机械设备的，比如说不需要寻道，对所有的地址访问开销都一样，特别是在随机读性能上，SSD远远好于传统机械硬盘。按这个道理，Flash设备随机访问和顺序访问的速度是一样，但现实上，Flash支持Cache操作，在顺序访问中可以提前把下一个页的数据读取放到内部寄存器中，可以更快响应读请求。所以在顺序访问上速度要比随机访问要快的。

读、写速度不一致。如之前文章描述的，把电子从浮动门中吸进去（写操作）比检测浮动门电场状态（读操作）要耗时。所以FTL在管理时，尽量减少写和擦除的操作。

FTL功能

地址映射管理。闪存设备对外是一个黑盒子，里面集成了Nand Flash和FTL等，上层应用使用逻辑地址来访问，FTL把逻辑地址映射到不同物理地址上，管理着每个逻辑地址最新的数据存放的物理位置，如图1。

垃圾回收。随着数据的写入，闪存设备上有些块的部分数据已经无效了，需要把有效的数据从块上搬走，然后擦除用来接收新的数据。

磨损均衡和坏块管理。因为每个块的P/E次数是有限的，某些块可能被重复使用而损坏了，而有些块数据很少被访问，所以一直没有进行操作过。为了避免这种情况，FTL加入磨损均衡的功能，大致是通过控制垃圾回收和空块池的管理，从而平衡每个块的使用次数，最理想是所有块一起达到磨损阈值。

由于Flash本身就存在部分坏块，在使用的过程中部分块会变坏，所以FTL在管理的时候需要避开这些无用块，把使用后变得不稳定块上的数据及时拷贝到稳定位置。

FTL研发关注点

1. **映射管理性能。**映射分为两种：逻辑地址到物理地址映射，成为直接映射；物理地址到逻辑地址映射，称为逆映射。大部分时候使用的都是直接映射，直接映射可以可能保存在SRAM中，也可能存放在Flash中。逆映射需要通过扫描Flash来建立映射关系，因为逆映射关系存放在Flash页的冗余空间中。可见，映射关系存放在SRAM中时获取对应关系速度是最快的，但由于SRAM资源比较紧张，所以会把所有映射关系存放在Flash中，软件只会加载使用的那段映射关系。逆映射需要扫描Flash来建立映射关系，所以速度最慢，一般用于特殊情况映射关系的恢复，比如说操作过程中断电了，或者某张映射表格所在的物理页坏了，需要重建。

映射影响着我们数据的存放，比如说需要区分顺序数据和随机数据，随机数据可能区分为热数据、冷数据。如果把热数据和冷数据存放在同一个块中，由于热数据更新频繁，导致物理块有效页利用率不高，把此块进行垃圾回收时，要把其中冷数据有效页拷贝到其它地方，导致写放大。如果冷热数据区分开，可以避免在垃圾回收中回收到冷数据。

映射管理是FTL中最重要的一点。

1. **垃圾回收效率**。垃圾回收需要考虑到冷热数据、磨损平衡和读写延迟。回收过程中把热数据和冷数据混合在一个块中，会导致冷数据的频繁读、写，加剧Flash损耗和影响性能。而且，热数据会被经常更新，在垃圾回收流程中应该避免回收热数据块。所以垃圾回收流程，需要综合考虑几大方面。
2. **读写延迟。** 读写延迟主要是指上层应用发送读写指令，设备处理、回复数据的时间。理想情况是，收到指令，软件在RAM中找到映射关系，返回数据。事实上影响的因素很多，比如说存在后台操作，在指令执行中间，软件可能进行垃圾回收,不能及时响应命令，如果垃圾回收流程控制不好，会造成上面这种情况，写速度波动很大，给人感觉就是在打CS，时而流畅，时而卡顿。
3. **SRAM需求**。SRAM价格比较高昂，所以一般产品会严格控制SRAM的大小。如果SRAM足够大，可以把映射信息都保存在SRAM上，也把Flash中的数据都读取到SRAM上，性能是最好的（此外最好得有个备用电源或者超级大电容，意外掉电时可以把SRAM中被更新的数据写回Flash~）。事实上SRAM大小远远不够的，所以有块管理算法、混合管理算法和把映射关系存放在Flash中的方案，缓存上层应用数据也要考虑把部分冷数据回写到Flash中。
4. **Flash并行操作**。闪存设备底层，每片Flash可以进行Multi Plane操作，多片Flash间同时操作，也可能有多通道可以并行操作。这些操作可以大大加速对Flash的访问速度，有点类似于RAID技术。 以上提到的FTL基本功能是相互关联的。比如说，上层应用写数据到某个逻辑地址上，地址映射管理在空块池中获取一个物理块来接收数据，然后触发了垃圾回收。垃圾回收在选择物理块的时候需要考虑到块的磨损情况、是否坏块，在回收这个块的时候需要使用并行操作，提高回收速率。
5. **异常处理**。在Nand应用场景中，比如说SSD、U盘、eMMC，均存在一些突然断电的情况。如何在突发掉电的情况后恢复数据，保持映射关系正常。比如说，RAM中缓存着最新的映射关系，没有保存在Flash中，此时掉电后，如何在下次上电后把RAM中的映射关系恢复；如何保证当前找到的映射表格是最新的等等。还有规避Flash特性造成的错误，如驻留错错误（Data Retention）、Disturb。