**Micro-benchmarking Flash Memory File-System Wear leveling and Garbage Collection : a Focus on Initial State Impact**

**Abstract**

闪存的优点：低能耗/I/O性能好/数据存储密度高

闪存的缺点：寿命有限/写前擦除规则/写入擦除操作粒度不对称

在专用FFS(闪存文件系统)的帮助下，这些缺点可在硬件(专用控制器)或软件中处理

根据不同初始状态，本文全方位的研究了不同FFS实现的磨损均衡/垃圾回收对于闪存性能/寿命的影响

←通过添加闪存特定功能微基准技术推动研究

**Introduction**

NVM日益兴起→闪存是NVM最成熟广泛的技术→闪存优缺点兼具

相比于HDD：

闪存中没有机械部分→闪存的顺序读/随机读性能更优

大部分情况，闪存顺序写性能更优

闪存缺点→随机写性能随生产批号不同/状态(如填充率)不同变化

为处理上述特性，闪存需具备下述功能：

1. 逻辑到物理地址的映射→实现异地更新
2. 磨损均衡机制→平衡全部闪存单元的磨损→解决时空局限性(若不采取具体行动会导致快速磨损)
3. 垃圾回收算法→将无效页面恢复为可用空间
4. 错误矫正码ECC 5. 电源故障修复机制

→→其中本文涉及1/2/3

→若由硬件实现：FTL层→FTL模拟块设备→使得能用标准文件系统格式化基于闪存的外围设备

如：裸闪存芯片直接由操作系统内核控制←通过专用FFS实现

→→为实现全部闪存管理服务→FFS需执行全部标准文件系统服务

本文利用微基准工具，研究不同闪存初始状态的磨损均衡/垃圾回收

→调查填充率标准/写入请求频率标准

**Background on NAND flash memories**

闪存是基于NVM的浮栅晶体管→根据基本组成元素的逻辑门种类

→分为NOR闪存和NAND闪存两种

①NOR闪存：更可靠/不需ECC/支持字节随机访问/密度低/成本高

→适用于代码存储

②NAND闪存：块寻址/存储密度高/成本低/读写性能优

→适用于存储数据→→本文仅讨论NAND闪存

闪存结构：晶粒→面板→块→页

页：用户数据空间+元数据空间(又称带外空间OOB)

带外空间OOB：包含ECC信息/页面状态等

读写操作单元：页 擦除操作单元：块

闪存限制：

①性能不对称：基本闪存单元的电气特性→写操作时间长于读操作

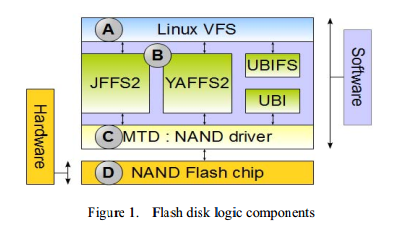
②写/擦除操作粒度不平衡：写操作单元是页，擦除操作单元是块

③写前擦除

④擦除/写操作周期有限：影响闪存寿命←需磨损均衡机制辅助

**Flash file-systems**

**A. Kernel File-system Organization**



内核文件系统

管理层

图1中：

1. 最高层-虚拟文件系统(VFS)-允许用户级对不同文件系统有独一无二的界面-将FFS的复杂性及结构抽象化-维持系统缓存以加速I/O
2. 专用文件系统-请求文件信息时，VFS请求B创建必要的目标
3. 存储技术设备(MTD)-为使低层能够执行原始闪存I/O-MTD提供NAND驱动器-对不同NVM设备(如NOR/NANAD闪存)提供通用界面

FFS的特点：

1. 数据压缩→节省存储空间/降低I/O负担
2. 坏块管理：块磨损→文件系统将其标记为“不可用”
3. 磨损均衡/垃圾回收机制
4. 日志功能→系统崩溃/掉电时保持数据一致性

**B.The Journaling Flash File System (JFFS2)**

最常用FFS

JFFS2节点：数据/元数据存储单元

(一个节点代表一个文件或文件的一部分，其大小在一个闪存页和半个块大小之间)

安装时，JFFS2扫描整个分区→给闪存JFFS2节点创建直接映射表

JFFS2相关闪存块列表：

1. 空闲列表：包含已准备好被写入的块←用于JFFS2需要新块写入节点时
2. 洁净列表：包含有有效节点的块
3. 脏列表：包含有大于一个无效节点的块→空闲空间不足时，垃圾回收器擦除脏列表中的块

(考虑磨损均衡，JFFS2常从洁净列表中选择块使用，而非空闲列表)

←空闲列表准备好被写入即常被擦除磨损多

→选用洁净列表中的块，要事先将其中有效数据复制到其他块

**C. Yet Another Flash File-System (YAFFS2)**

通过应用补丁实现YAFFS2与内核的集成

YAFFS2在chunk结构中存储数据：

用一个包含元数据(类型/存储权限等)的头chunk和包含文件/用户数据的多个数据chunks表示各文件

chunk大小=底层闪存页大小

闪存中顺序写入chunk

YAFFS2也使用数据chunk的带外区域(如用于将更新数据置为无效)

安装时，YAFFS2扫描整个分区

YAFFS2不支持压缩

发生写入且包含旧数据的块已完全无效时→执行GC→擦除

空闲空间低于阈值→GC选择某些包含有效数据的块→复制有效chunks到其他位置→擦除被选择的块

**D. Unsorted Block Image File-System (UBIFS)**

用于解决JFFS2可扩展性的相关问题

UBIFS依赖于附加层-UBI层(UBI执行逻辑到物理的块映射并将UBIFS从磨损均衡及坏块管理服务中卸载)

对于文件索引，UBIFS使用树结构：

索引树通过索引节点存储在闪存

树叶指向包含闪存数据或元数据的闪存位置

UBIFS使用标准数据节点存储文件数据

UBIFS将闪存量分区：

1. 包含数据和索引节点的主要区域
2. 包含块元数据(GC的擦除/无效计数)的逻辑擦除块属性树区域

闪存不允许就地更新→更新树节点→其祖先均移动至其他位置←“漫游树”

为减少闪存访问的数量→文件数据/元数据修改缓存于主存并定期在闪存媒介中刷新

UBIFS记录文件系统的每一次修改→维持掉电时文件系统的一致性

**Performance evaluation methodology**

研究存储系统可用的三种基准的主要类型：

1. Macro-benchmarks：测试众多文件系统操作→纵观系统行为
2. Trace-Based：重跑代表性应用的I/O traces→面对专用负载时的存储系统行为
3. Micro-benchmarks：允许为测量而隔离系统的某些专用部分

FFS运用不同的磨损均衡和GC结构/算法→尝试在全部性能测试中描述最流行FFS的行为

1. **Performance Metrics**

性能指标：擦除操作数量/磨损均衡/平均响应时间

响应时间通过gettimeofday()系统调用收集，微秒精度

基准测试工具：uFilp/由scratch发展而来的补充基准测试工具

uFilp由使用标准文件系统的黑匣子闪存FTL发展而来

实验：uFilp在多种请求大小下随机/顺序访问文件

在I/O请求中插入timeouts→测试异步垃圾回收

Flashmon工具：擦除操作数量

允许在MTD级跟踪全部闪存访问(读/写/擦除)→在所有FFS上工作

其日志存储于RAM(不会干扰测试)

YAFFS2不支持压缩→为公平，测试时禁用JFFS2/UBIFS压缩

1. **Tested Hardware Platform**

基于OMAP 3530 EVM 的Mistral tmdsevm3530 开发板，集成有OMAP3530(720MHz ARM Cortex-A8 / 520MHz c64x+ DSP)处理器和256 MB LPDDR / 256MB NAND闪存

闪存：美光SLC/页大小2KB/块大小128KB

闪存芯片延迟：读延迟25us/写延迟200us/擦除延迟500us

Linux内核版本：2.6.37，补丁包含YAFFS

1. **Flash Specific Benchmarking Tests**

假设闪存处于空闲状态且每一次测试前都被格式化是不现实的

→需要分两步初始化闪存

1.将测试分区用一个大型随机数据文件填满

2.在测试前删除文件

→→闪存处于无效状态

基准测试在文件删除后立即启动→防止FFS的异步垃圾收集操作

第二组微基准测试根据可用的清洁空间精确测试垃圾收集和磨损均衡性能→每个测试的初始状态调整不同←不同初始状态不同性能

1. **Simple sequential and random tests with saturated flash memory：**

简单负载上研究FFS垃圾回收/磨损均衡

三个参数：1.I/O请求类型：读/写 2.I/O大小：2-512KB

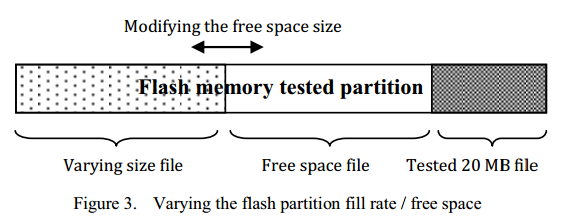
3.访问模式：顺序/随机

调整其中一个参数的值不同→多个实验

各FFS有20MB大小的洁净分区→填充随机数据(部分空间用于存储元数据和执行垃圾回收，不会100%填充)→擦除数据文件→写操作时需垃圾回收→使相关操作有延迟→直接影响I/O响应时间

1. **Varying flash memory fill rate / free space：**

根据可用洁净空间的大小测量垃圾回收性能→调整分区填充率大小



分区大小100MB→对其中20MB应用微基准(即I/O请求测试文件)→创建一大型数据文件(占用空闲空间，大小30-75MB)→就空闲空间而言，垃圾回收器空间小

测试文件空间为脏状态

1. **Varying I/O request inter-arrival times：**

可用空间完成写入操作→同步启动垃圾回收机制

I/O超时期间→异步启动垃圾回收机制

本组测试目的：测试各FFS的上述特性

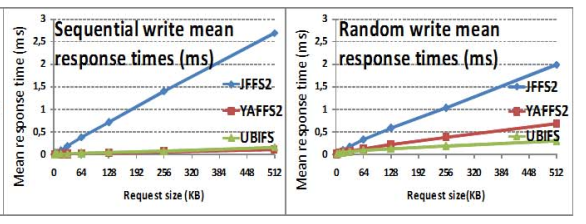
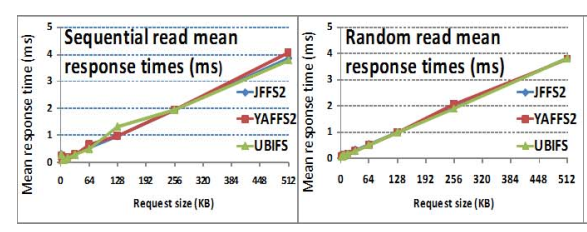
执行不同请求大小(同一测试I/O请求大小相同)的随机写操作

在各连续写请求(10ms、100ms和1000ms )间插入休眠调用→不同测试的内部到达时间不同

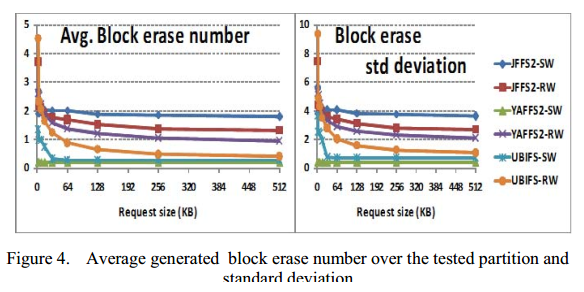
其余初始状态同1）

**Results and discussion**

**A. Simple tests with saturated flash memory**



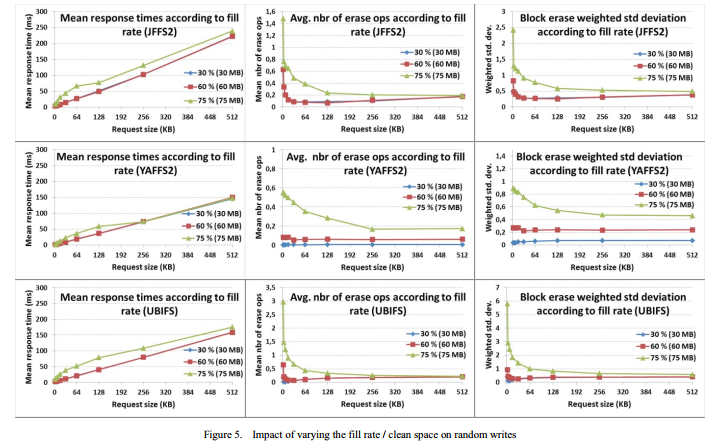
①



**B. Varying flash memory fill rate / free space**

洁净空间填充率对写性能的影响→仅考虑随机写

洁净空间填充率对读性能无影响→不考虑读性能



(30%和60%曲线重复)

**C.Varying I/O request inter-arrival times**

写请求间插入超时→异步GC

若响应时间没有变化→FFS未使用内部达到时间来擦除块

可选的内部到达时间：0ms(基准)/10ms/100ms

