ParaFS: A Log-Structured File System to Exploit the Internal Parallelism of Flash Devices.

`Log-structured` `FileSystem` `Flash` `Parallelism` `Open-channel`

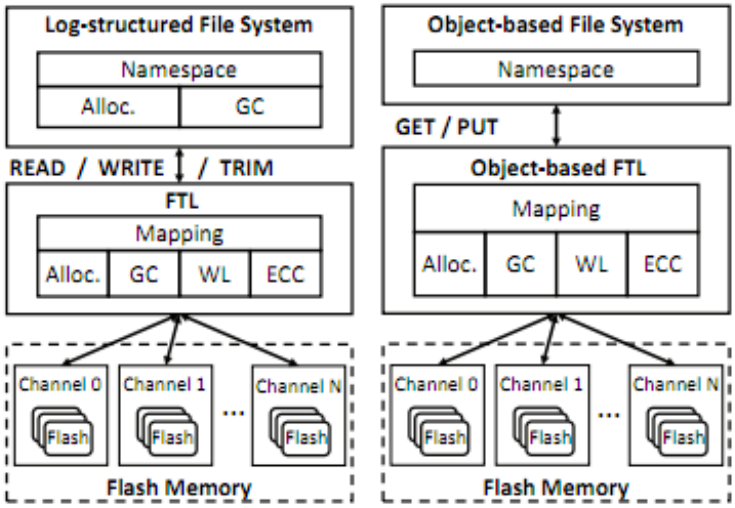
Zhang, Jiacheng, Jiwu Shu, and Youyou Lu. "ParaFS: A Log-Structured File System to Exploit the Internal Parallelism of Flash Devices." USENIX Annual Technical Conference. 2016.

背景

为了发挥闪存存储的性能优势，文件系统设计经历了快速的革命式的发展。然而现有的设计没有很好地利用起闪存内部并发的特性，并且即使是一些针对闪存做优化的文件系统，与闪存管理之间在垃圾回收机制上存在许多不协调的地方，带来较大的性能降低，特别是在写负载大的情况下尤为严重。ParaFS提出2D的数据布局，保持数据冷热聚集的前提同时进行数据的条带化分布，发挥闪存并发特性；另外，对文件系统层和FTL层的GC进行协同设计，提高GC效率;最后，对闪存不同channel的读写擦除操作进行调度以提供稳定的性能。

动机与挑战

- 闪存文件系统架构
  现有的文件系统通过FTL进行闪存设备的访问。日志被认为是闪存友好的访问，在文件系统层如F2FS,NILFS或FTL层都会采取此方式。FTL向文件系统呈现类似HDD的访问接口，因而传统的文件系统可以运行在闪存设备上。传统架构下的FTL提供了很好的存储抽象，但也扩大的文件系统和存储之间的语言隔离；甚至由此带来一些功能冗余，如空间分配，地址映射，垃圾回收等，从而导致性能降低，闪存耐久性降低。



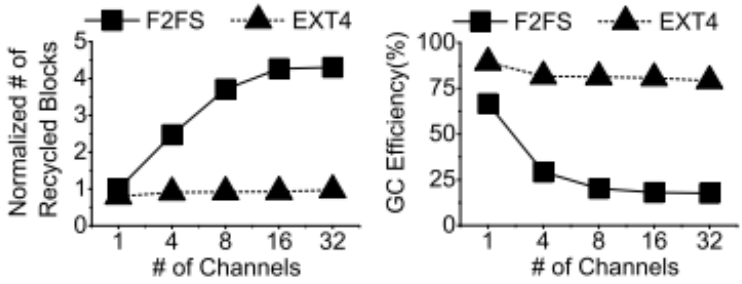
(a) Block Interface (b) Object Interface

Figure 1: FS Architectures on Flash Storage

- 发挥内部并发面临的挑战

  对NAND闪存的IO请求会分布到不同的channel上，从而获得很高的并发性能，现有的文件系统并没有完全发挥出此内部并发特性。从以下几个方面得以体现：

  - 冷热分组问题。文件系统按照一维线性空间进行分配页，FTL再进行地址的映射。对闪存来说，冷热数据聚集能减少垃圾回收过程中有效数据的移动。文件系统层会进行冷热数据的分组，而FTL的地址映射又会将数据进行条带化，从而打破了冷热数据分组，造成不协调。
  - 垃圾回收问题。F2FS在写负载很大的时候GC的效率降低；当channel增大的时候，F2FS中块的擦除会增加，GC效率降低。



(a) # of Recycled Blocks (b) GC Efficiency

Figure 2: GC Statistics under Heavy Traffic

  - 稳定的性能。 写和擦除操作会给读带来延迟增大的问题，可以进行动态的IO调度以缓解此问题。

设计思路

- ParaFS架构

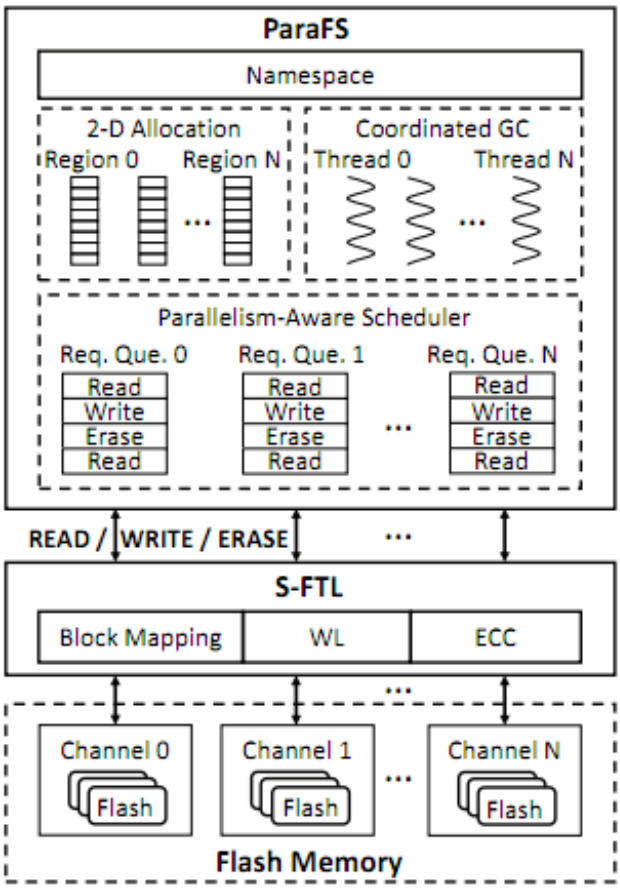


Figure 3: The ParaFS Architecture

  - ParaFS使用一个简单的S-FTL，进行块级别的地址映射；垃圾回收功能简化为对闪存块的擦除操作；S-FTL向文件系统提供闪存的物理布局抽象。
  - 文件系统层使用Log-Structured的结构，为避免“wandering tree”问题，文件系统元数据不采用Log-Structured结构，维护一个page-level FTL进行原地更新。
- 2-D数据分配（2-D Data Allocation）
  - 通道（channel-level）维度。ParaFS按页对请求进行划分，然后条带化到不同的区域。 这些区域与闪存channel一一对应，能很好利用闪存的并发访问。
  - 热度（hotness-level）维度。对每个区中的数据进行热度聚集。区域中不同热度对应不同的日志头，从而保证热度相近的数据能够聚集。实现中ParaFS借鉴F2FS的做法使用了6个日志头。

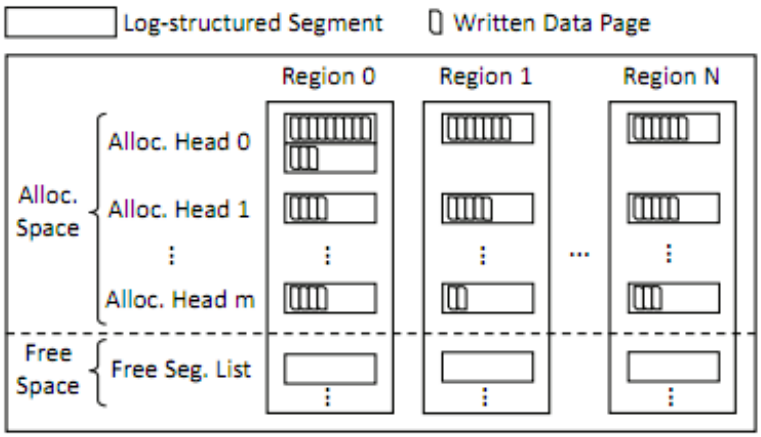


Figure 4: 2-D Allocation in ParaFS

- 协同设计垃圾回收（Coordinated Garbage Collection）
  - 文件系统层paraGC，在空闲空间低于某个阈值时，触发前台GC；在系统处于idle状态时，进行后台GC。前台GC使用贪婪策略，后台GC使用cost-benifit策略。
  - FTL层GC。S-FTL不需要进行闪存页状态和闪存块利用率的记录，GC得到很大的简化。
  - ParaFS还提供多线程的GC操作，提高GC效率。
- 并行感知的IO调度（Parallelism-Aware Scheduling）
  - 在请求派遣阶段（dispatching phase）。读写请求的channel是固定的，而写请求有调整的空间；ParaFS选择channel中相对来说最不忙的来响应写请求。调度器使用不同权值表示读写请求的繁忙代价（W_{read}, W_{write}），由于读写的不对称性，写延迟一般是读的8x，可设置$W_{read} = 1, W_{write} = 8$。可得：

$$W_{channel} = \sum(W_{read} * Size_{read}, W_{write} * Size_{write})$$

  - 在请求调度阶段（scheduling phase）。考虑到公平性，调度器为读写擦除请求使用相同的时间片（Slice）大小。在读时间片中，进行读请求的调度；当时间片用完或者队列中没有读请求，调度器将有策略的进行写或者擦除请求的调度。

$$e = a * f + b * N_e$$

其中f为channel中的空闲块比例，N_e是正在响应擦除请求的channel比例，a, b表示上述两个参数的权值。如果$e > 1$,在时间片中进行写请求的调度，否则在这个时间片进行擦除请求调度。读或擦除时间片之后，调度器开始读时间片。当前设计下，$a = 2, b = 1$,即当空闲空间在50%以下是才开始擦除请求的调度；这种策略下，当空闲空间不足时，擦除请求的优先级也更大。

测试与结果

主要回答：（1）ParaFS与其他文件系统相比，不同写负载下的效率 （2）ParaFS的优化带来哪些好处？
在不同的负载下与EXT4，BtrFS和F2FS进行了比较。

Table 3: Workload Characteristics

Name	Description	# of Files	I/O size	Threads	R/W	fsync
Fileserver	File server workload: random read and write files	60,000	1MB	50	33/66	N
Postmark	Mail server workload: create, delete, read and append files	10,000	512B	1	20/80	Y
MobiBench	SQLite workload: random update database records	N/A	4KB	10	1/99	Y
YCSB	MySQL workload: read and update database records	N/A	1KB	50	50/50	Y

作者还测试了ParaFS在使用不同优化策略下，长期运行过程中，闪存空间使用情况不断变化，所呈现的性能特征。

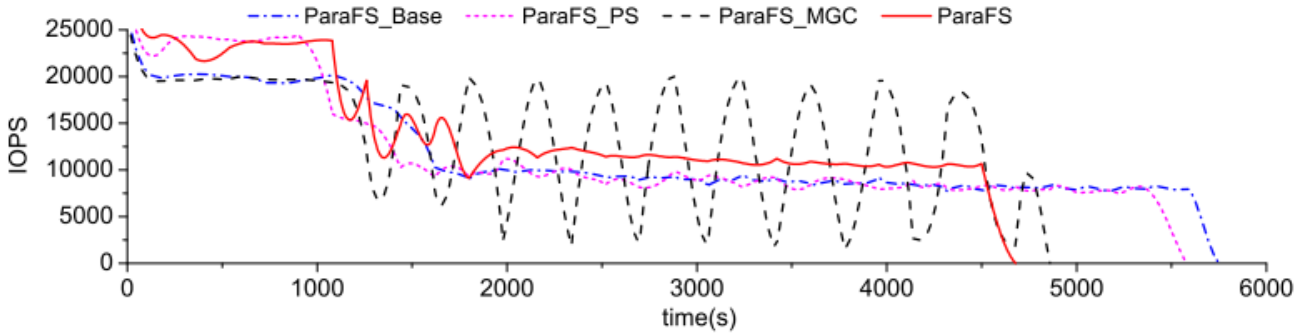


Figure 8: Performance Consistency Evaluation

结论

ParaFS提出了文件系统层和FTL层在闪存管理上的不协调的问题，并提供了一种文件系统和FTL协同的高效利用闪存内部并发的文件系统设计思路，为Open-channel SSDs的运用提供了范例。