新型分区存储研究综述

宋郅炜[1)]

[1)](华中科技大学计算机学院 武汉市 中国 430074))

摘 要 分区命名空间 SSD 将逻辑块分组到分区中，分区随机读取顺序写入，将 SSD 的 FTL 层功能移到主机侧。因此需要在主机端实现垃圾回收，但是存在写放大和垃圾回收进程与主机进程竞争的问题。此外，现有的软件系统无法充分利用 ZNS SSD 的新特性。本文将对 ZNS 主机侧垃圾回收和段压缩策略进行综合性分析，并且着重讨论如何对现有软件系统进行更改。

关键词 ZNS SSD 垃圾回收 软件系统 写放大

Literature review of NVM-based file system research

Zhiwei Song[1)]

[1)](Huazhong University of Science and Technology of Computer School, WuHan China 430074)

Abstract The SSD of the zoned namespace groups logical blocks into partitions, and the partitions read and write data in sequence at random. The FTL layer function of the SSD is transferred to the host. Therefore, garbage collection needs to be implemented on the host side, and there are problems such as write amplification and competition between garbage collection process and host process. In addition, existing software systems are unable to take full advantage of the new features of the ZNS SSD. This article provides a comprehensive analysis of ZNS host-side garbage collection and segment compression strategies, and focuses on how to make changes to existing software systems

Key words ZNS SSD; garbage collection; software systems; write amplification

1. 引言

传统 SSD 存在性能和写放大问题，因此 ZNS SSD 出现 (Zoned Namespaces SSD)受到了工业界和学术界的广泛关注，ZNS SSD 是 2019 年由西部数据和三星公司推出的新一代 SSD。由于 HDD 的广泛应用，块接口提供的语义对应用程序必不可少，这造成了操作与 SSD 底层闪存介质性质不匹配，只有在块被完全擦除以后才能重写，传统的 SSD 通过使用闪存转换层(flash translation layer，FTL)实现动态的逻辑到物理页面映射，使用大量 DRAM 来隐藏这种不匹配，但由于闪存块存在物理限制，因此经常需要进行垃圾回收(garbage collection GC)同时也带来了预留空间(over provisioning OP)、写放大(write amplification WA)、吞吐率受限、性能不可预测等问题，ZNS SSD 可以有效提升 SSD 的读写吞吐，降低写入时的写放大，减少 SSD 本身的预留空间，并且还能解决传统 SSD 在空间占用达到一定程度时由于内部垃圾回收导

收稿日期：2023年1月12日

致的性能不稳定的问题，因此利用 ZNS SSD 来构建存储系统是一个趋势。于此同时，ZNS 中依然存在空间回收的问题，以及如何在现有的文件系统应用 ZNS SSD。本文针对 ZNS 中垃圾回收问题和文件系统设计进行调研。

2. ZNS 垃圾回收

为了为新的写入释放空间，由 SSD 控制器执行的垃圾收集器(GC)回收无效的块，并将多个擦除块中仍然有效的块合并到一个新的擦除块中，然后擦除释放的擦除块。此操作需要对驱动器中的闪存介质进行过度供应(OP)，以减少 GC 操作期间的副本数量。ZNS SSD 将设备内的 GC 转移到主机侧，消除写放大(WA)来提高 I/O 性能，但是使用 ZNS SSD 也必须执行区域(zone)清理。但是依然存在写放大的问题，并且主机端 GC 进程直接与常规应用程序竞争这些主机资源，需要以尽量低的成本将 GC 从设备端加载到 CPU 上。

2.1 交换技术

空间回收过程自然涉及到设备上逻辑块的迁移,这对于 SSD 端 GC 来说不是问题，因为用户可见的逻辑块地址(LBA)保持不变。然而，将这种解决方案应用于主机端 ZNS GC 将在主机中引起不可接受的空间开销。Bergman 在 ZNSwap 中提出一种 ZNGC 的垃圾回收策略[1]，设备中为每个 4KiB 块维护反向映射，ZNSwap 通过将反向映射信息存储到逻辑块元数据中，从而避免了主机中的这些开销。将碎片区域中的有效块合并为新块，然后擦除释放的区域，因为区域中一部分页是有效的。由于交换区的作用是改善性能，而不是作为内存不足时的备选，因此交换区容量较大，与 ZONE 绑定不仅可以有更高的性能，还能节省相当的内存空间。

ZNGC 分为 4 个阶段，Gather：将 ZONE 中缓存的页面的交换槽移除，且交换槽无效；并将还在使用的交换槽收集起来，为读取做准备。Read：将收集的交换槽读取放入缓存。Write：根据剩余的 ZONE 的情况分配并写入目标区域。Activate：之前收集的交换槽暂时还允许进程访问，待到映射表和页表更新完成之后，再将之前的交换槽真正的清理。ZNSwap 动态调整也存储在交换设备中的换入页的数量，从而提高主要读和混合读写工作负载的性能。操作系统在交换设备中保留未修改的交换内存页的副本，ZNSwap 允许轻松地自定义磁盘空间管理策略，以根据特定系统的交换需求定制 GC 逻辑。

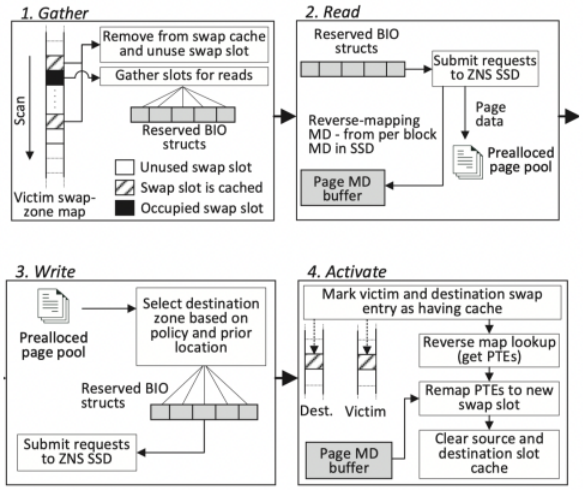

图 2-1 ZNGC 垃圾回收

2.2 压缩优化

区域清理过程中，如果选中作为 zone cleaning 的 zone 中仍然保留有效数据，则由于复制了有效数据，WA 问题仍然存在。Lee 提出一种应用感知的数据放置决策[2]，通过最小化区域清理期间复制的数据量来减少 WA。设计了一种新的压缩感知区域分配算法(CAZA)，该算法的设计基于观察到一组 sstable 删除/失效是由 LSM-tree 的压缩算法决定的。压缩作业选择一组相邻级键范围重叠的 sstable，将它们合并为一个或多个新的 sstable，并删除用于合并的压缩输入的 sstable。考虑到压缩过程，CAZA 将新创建的 SSTable 放在与其关键范围重叠的 SSTable 最多的区域中。然后当将来触发压缩时，选择作为合并受害者的 SSTable 将与已经在同一区域的 SSTable 合并，并一起删除/失效。这将使保留在区域中的有效数据量最小化，从而最大限度地减少在擦除区域之前复制有效数据所造成的开销。

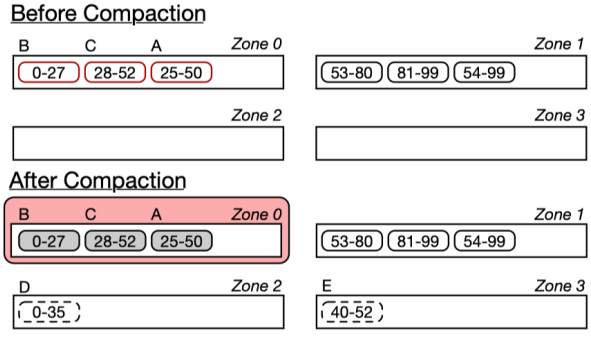

图 2-2 CAZA 区域清理

使用 ZNS SSD 的 RocksDB 通过 ZenFS(一种用户级文件系统)使用基于寿命的 zone Allocation 算法(LIZA)将具有相似寿命的数据放置在同一个 zone 中，并在回收 zone 时最大限度地减少有效数据复制的 GC 开销。LIZA 通过根据 lsm -树的层次结构级别预测每个 SSTable 的生命周期来分配区域，由于 SSTable 生命周期的不准确预测，在最小化写放大(WA)问题方面非常低效。lsm 树中 sstable 的删除时间完全由压缩过程决定，CAZA 算法允许新创建的 sstable 在合并后一起删除。RocksDB 的 ZenFS 中实现了 CAZA，我们的广泛评估表明，与 LIZA 相比，CAZA 显著降低了 WA 开销。

3. ZNS 文件系统

ZNS 文件系统通过嵌入更多闪存芯片来增加 SSD 的带宽。ZNS 设备的区域大小将被确定为足够大，以利用 ssd 内部闪存芯片的并行性。ZNS SSD 的带宽越高，分区大小就越大，文件系统使用的段大小也就越大。然后，主机会更严重地遭受段压缩开销，因为开销通常会随着段大小 的增加而增加。为了提高 IO 性能并克服收益递减的问题，需要主机-设备协同设计，而不是简单地将 GC 开销从 SSD 转移到主机。

3.1 LFS优化

ZNS 采用顺序 write-only zone 模式，需要 LFS (log-structured file system)访问 ZNS 的 ssd 盘，对应的 LFS 必须承担段压缩开销。Han 提出支持 lfs-aware ZNS 接口的系统 ZNS+[3]，设备级支持来减轻 LFS 的段压缩开销，可以考虑两种方法:压缩加速和压缩避免。

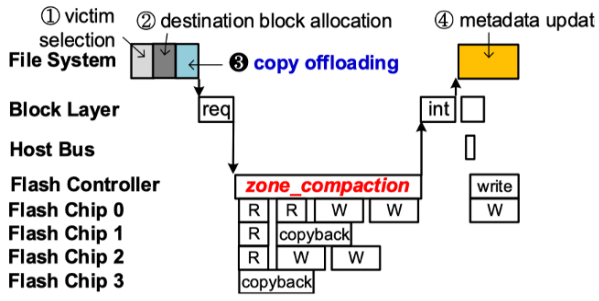

图 3-1 ZNS+段压缩

通过 zone_compaction 和 TL_open 两个新命令支持内部区域压缩(IZC)和稀疏顺序覆盖。段压缩需要四个子任务:victim 段选择、目标块分配、有效数据复制和元数据更新。尽管所有其他任务都必须由主机文件系统执行，但最好将数据复制任务卸载到 SSD，因为设备端数据复制比主机端数据复制快。

对于压缩加速，ZNS+ 允许主机通过 zone_compaction 将数据拷贝任务卸载到 SSD 上。对于通过 TL_open 打开的区域，线程日志记录允许稀疏顺序覆盖。ZNS+ SSD 将稀疏的顺序写请求转换为密集的顺序请求，方法是在同一段中用未接触的有效块堵塞请求之间的漏洞(内部堵塞)，并将合并的请求重定向到新分配的 flash 块。内部堵塞是在写请求之间处理的，因此与批处理风格的段压缩相比，它提高了写请求的平均响应时间。虽然线程日志可以避免段压缩开销，但它有几个缺点，考虑到线程日志记录的优点和缺点，我们为 ZNS+提出了混合段回收技术，它根据回收成本和收益选择线程日志记录或段压缩。

3.2 软件接口

块接口和当前存储设备特征之间存在显著的不匹配，主机负责以擦除块的粒度管理数据，存在块设备接口税的问题。Bjørling 认为 FTL 转移给主机的效果不如将存储软件的数据映射和放置逻辑集成[4]，提出了调整主机软件层以利用 ZNS SSD，并且对包括 Linux 内核、f2fs 文件系统、Linux NVMe 驱动程序和分区块设备子系统、fio 基准测试工具进行更改以支持 ZNS。在对 Linux 内核修改中，对 ZBD subsystem 修改了 NVMe 设备驱动程序，Zone Capacity zone 中，在区域描述符(zone descriptor)数据结构增加新的区域容量(Zone Capacity)属性，Limiting Active Zones 中为修改 F2fs 将 limit 设为 6，分区块设备不支持足够的活动分区，则将打开的段的数量配置为与设备可用的数量一致。

4. 总结与展望

本文首先从对 ZNS 性能有重要影响的垃圾回收出发，介绍了 ZNSwap，引入了一个主机端 ZN GC，与交换逻辑共同设计，以减少垃圾收集开销并提高系统性能。随后针对区域清理问题，分析了 ZNS 的 lsm 树一种新的压缩感知区域分配算法(CAZA)，最大限度地减少区域清理期间的写放大开销。

其次当前的软件系统无法充分利用 ZNS 新特性，ZNS 专注于 ssd 端的好处，而没有考虑主机复杂性的增加。ZNS+支持存储区压缩和稀疏顺序覆盖，将块拷贝操作卸载到 SSD，还提出了支持 ZNS+

的文件系统技术，即支持回拷的块分配和混合段回收。和传统块设备不同，针对如何存储软件在应用ZNS 接口，介绍了对现有软件层的更改方案，包括对 Linux 内核、f2fs 文件系统、Linux NVMe 驱动程序和分区块设备子系统、fio 基准测试工具等。

但是关于 ZNS 特性的讨论又不仅仅限于本文所讨论，Shin 对 ZNS 的并行性、隔离性、可预测性进行试验分析[5]，提出建议包括可以设计算法使用更大的请求来触发 IO，暴露更多关于区域隔离性的信息，将频繁访问的数据放在更快的区域内。ZNS 和主机协同方面依然存在很多问题，可以考虑应用程序级的 I/O 行为，并尝试将它们与 ZNS ssd 的特性相匹配。

参 考 文 献

[1] Bergman S, Cassel N, Bjørling M, et al. {ZNSwap}:{un-Block} your Swap[C]//2022 USENIX Annual Technical Conference (USENIX ATC 22). 2022: 1-18.

[2] Lee H R, Lee C G, Lee S, et al. Compaction-aware zone allocation for LSM based key-value store on ZNS SSDs[C]//Proceedings of the 14th ACM Workshop on Hot Topics in Storage and File Systems. 2022: 93-99.

[3] Han K, Gwak H, Shin D, et al. ZNS+: Advanced zoned namespace interface for supporting in-storage zone compaction[C]//15th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 21). 2021: 147-162.

[4] Bjørling M, Aghayev A, Holmberg H, et al. {ZNS}: Avoiding the Block Interface Tax for Flash-based {SSDs}[C]//2021 USENIX Annual Technical Conference (USENIX ATC 21). 2021: 689-703.

[5] Shin H, Oh M, Choi G, et al. Exploring performance characteristics of ZNS SSDs: Observation and implication[C]//2020 9th Non-Volatile Memory Systems and Applications Symposium (NVMSA). IEEE, 2020: 1-5.