软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2022,33(12):4851−4868 [doi: 10.13328/j.cnki.jos.006359]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

一种新型瓦记录磁盘的高可靠数据存储方法*

吴坤尧[1,2], 柴云鹏[1,2], 张大方[1,2], 王 鑫[3]

[1](数据工程与知识工程教育部重点实验室(中国人民大学), 北京 100872)

[2](中国人民大学 信息学院, 北京 100872)

[3](天津大学 智能与计算学部, 天津 300350)

通信作者: 柴云鹏, E-mail: ypchai@ruc.edu.cn

摘 要: 近年来, 传统磁记录的存储密度增长已经达到极限, 为了满足快速增长的数据容量需求, 多种新型存储技术不断涌现, 其中瓦记录(shingled magnetic recording, SMR)技术已实现商业化, 在企业实际应用. 由于瓦记录磁盘的叠瓦式结构, 磁盘在随机写入时会引起写放大, 造成磁盘性能下降. 这一问题在部署传统的高可靠存储方案(如 RAID5)时会变得更加严重, 原因在于校验数据更新频率很高, 磁盘内出现大量的随机写请求. 研究发现瓦记录内部其实存在具有原位更新能力的“可覆盖写磁道 (free track)”, 基于“可覆盖写磁道”, 提出了一种专门针对瓦记录盘的高可靠数据存储方法——FT-RAID, 以替代经典的 RAID5 方法, 实现一种廉价、大容量、高可靠的存储系统. FT-RAID 包含两个部分: “可覆盖写磁道映射(FT-mapping)”和“可覆盖写磁道缓冲区(FT- buffer)”. FT-mapping 实现了一种瓦记录友好的 RAID 映射方式, 将频繁更新的校验块数据映射至“可覆盖写磁道”; FT-buffer 实现了一种瓦记录友好的两层缓冲区结构, 上层确保了热数据能够原位更新, 下层提高了缓冲区的容量. 基于真实企业 I/O 访问记录的实验结果表明, 与传统 RAID 5 相比, FT-RAID 能够减少 80.4%的写放大率, 显著提高存储系统整体性能.

关键词: 瓦记录; RAID; 磁盘; 存储; 容错

中图法分类号: TP302

中文引用格式: 吴坤尧, 柴云鹏, 张大方, 王鑫. 一种新型瓦记录磁盘的高可靠数据存储方法. 软件学报, 2022, 33(12): 4851−4868. http://www.jos.org.cn/1000-9825/6359.htm

英文引用格式: Wu KY, Chai YP, Zhang DF, Wang X. Highly Reliable Data Storage Method Based on Novel Shingled Magnetic Disks. Ruan Jian Xue Bao/Journal of Software, 2022, 33(12): 4851−4868 (in Chinese). http://www.jos.org.cn/1000-9825/6359.htm

Highly Reliable Data Storage Method Based on Novel Shingled Magnetic Disks

WU Kun-Yao[1,2], CHAI Yun-Peng[1,2], ZHANG Da-Fang[1,2], WANG Xin[3]

[1](Key Laboratory of Data Engineering and Knowledge Engineering of the Ministry of Education (Renmin University of China), Beijing 100872, China)

[2](School of Information, Renmin University of China, Beijing 100872, China)

[3](College of Intelligence and Computing, Tianjin University, Tianjin 300350, China)

Abstract: In recent years, traditional HDDs' areal density is reaching its limit. To extend the capacity of disk drives, several new storage techniques were proposed, including shingled magnetic recording (SMR), which is the first one to reach market among those new technologies. However, the shingled track structure of SMR disks encounters serious write amplification and performance declining when processing random write requests. Furthermore, constructing RAID5 based on SMR drives worsens the write amplification (WA) because the parity updating of RAID5 is very frequent to produce many random writes. This study, for current SMR disks' structure, finds that the first track of each band can be overwritten without impacting other tracks, because the wide write head can be moved a bit to cover both

* 基金项目: 国家重点研发计划(2018YFB1004401); 国家自然科学基金(61972402, 61972275, 61732014)
收稿时间: 2020-09-28; 修改时间: 2021-01-05, 2021-04-12; 采用时间: 2021-04-28; jos 在线出版时间: 2021-05-20

the first track and the guard region. In other words, the first track of each band can be called the free track, because it can be overwritten freely without causing any write amplification. Therefore, a new free-track-based RAID system (FT-RAID) is propose based on SMR drives, to fully develop the potentials of the overwriting-free region in SMR disk drives. FT-RAID is consisted of two key techniques, i.e., FT-Mapping and FT-Buffer. FT-Mapping is an SMR-friendly data mapping manner in RAID, which maps the frequently updated parity blocks to the free tracks; FT-Buffer adopts an SMR-friendly two-layer cache structures, in which the upper level can support in-place updating for hot blocks and the lower level can supply higher capacity for the write buffer. Both of them are designed to mitigate the degradation of performance by reducing SMR WA, leading to an 80.4% lower WA ratio than CMR-based RAID5 based on practical enterprise I/O workloads.

Key words: SMR; RAID; disk; storage; fault tolerance

随着社交网络、电子商务、人工智能等领域的快速发展, 数据总量高速增长. 根据 2019 年的《大数据白皮书》, 全球数据总量已经接近 41 ZB. IDC 预测全球数据总量到 2025 年将达到 175 ZB[1]. 与固态硬盘(solid state drive, SSD)相比, 磁记录(hard disk drive, HDD)具有成本低、存储稳定、寿命长、容量大等优点[2,3], 目前仍是主流的廉价大容量存储介质, 是大数据存储不可或缺的主流介质.

近几年, 受到物理学上超顺磁效应极限的影响[4], 传统磁记录的存储密度达到了极限(1 Tb/in^2), 工业界迫切需要新技术的支持来提高磁盘的存储密度. 包括瓦记录(shingled magnetic recording, SMR[5])、热辅助磁记录(heat-assisted magnetic recording, HAMR[6])、比特模式磁记录(bit-pattern magnetic recording, BPMR[7])在内的多种高密磁盘存储技术都出现并逐步产业化. 在这些技术中, 瓦记录存储技术目前最为成熟, 已经出现商业化产品[8], 与现有系统兼容度高.

写放大是瓦记录盘性能不佳的主要原因. 写放大指实际写入磁盘的数据量与用户写请求数的比值. 对于传统磁盘, 由于没有额外的写入数据, 写放大一般是 1 倍. 瓦记录的固有结构导致处理随机写请求时写放大显著. 由于写入的粒度较大(一般为 MB 级别), 而写请求的粒度较小(一般为 4KB), 所以在面对局部性不好的场景时, 瓦记录的写放大往往会很大, 达到几十倍. 当使用瓦记录组成磁盘阵列时(例如 RAID 5), 由于校验数据频繁更新, 因此写放大现象更为严重, 造成存储系统严重的性能下降. 目前关于瓦记录磁盘的研究多集中于单个磁盘的内部结构设计优化, 对于多个瓦记录磁盘构成的磁盘组(如 RAID 方式)的研究还很少.

虽然传统方法认为瓦记录磁盘进行随机写操作会引起严重的写放大, 但是本文发现了瓦记录磁盘中存在一些对随机写友好的“可覆盖写磁道”. 如图 1 所示, 通过观察瓦记录的结构, 可以发现在瓦记录磁盘中, 并非所有的磁道都不能随机写. 而是在每一个磁道分区(也被称为“带”, 即 Band)中, 第 1 个磁道之上由于存在 Band 之间的隔离空间, 写入时可以不用覆盖后续磁道. 每个 Band 的第 1 个磁道就是本文发现适合随机写的“可覆盖写磁道(free track, FT)”, 这些磁道可以在不引起写放大的情况下执行原位更新操作.

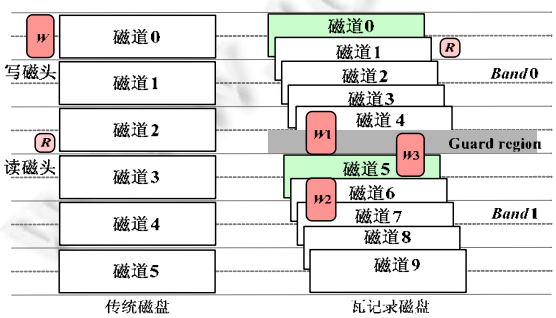

图 1 传统磁盘与瓦记录的结构示意图

为了缓和瓦记录 RAID 引起的显著写放大现象, 本文提出了基于可覆盖写磁道的 RAID 系统(即 FT-RAID). FT-RAID 充分利用了“可覆盖写磁道”的特性, 将写频繁的数据映射到可以覆盖写磁道上, 从而显著减少瓦记录 RAID 中的写放大现象, 提升系统性能.

FT-RAID 具体包含两个模块: 基于 FT 的新的 RAID 地址映射方式(FT-mapping)和基于 FT 的新的持久化

缓冲区管理方法(FT-buffer). 本文使用了软硬件结合的方式模拟瓦记录 RAID, 在传统磁盘的基础上用软件模拟瓦记录磁盘的读写逻辑; 然后基于5块模拟的瓦记录磁盘构建了瓦记录RAID的实验环境, 并使用来自企业的真实磁盘访问记录进行了实验. 实验结果证明, 在 10 种不同的企业磁盘访问记录的测试中, FT-RAID 与瓦记录传统 RAID 5 方案相比, 能够平均降低 80.4%的写放大, 对写主导的应用优化程度最明显.

FT-RAID 主要包括以下几点贡献:

(1) 发现瓦记录磁盘中的“可覆盖写磁道”, 使上层应用可以在可覆盖写磁道进行原位更新操作而不引起写放大, 这种新型瓦记录磁盘可以进行部分的随机写, 有助于减小写放大, 提升性能. 同时, 可以在不改动现有瓦记录硬件结构的情况下, 通过拓展当前指令集实现该功能. 可覆盖写磁道在当前瓦记录磁盘的架构中, 可以支持部分随机写; 不仅可以在本文提出的瓦记录RAID中有直接减小写放大的作用, 而且具有启发性, 在其他领域和方法中也可以应用来提升性能.

(2) 提出一种基于可覆盖写磁道的、对瓦记录友好的 RAID 方案 FT-RAID, 包括新的地址映射方式 FT-mapping 和瓦记录写缓冲区的新型管理方法 FT-buffer. FT-mapping 将 RAID 中频繁更新的校验块映射至可覆盖写磁道, 使得 RAID 系统中校验块的频繁更新和触发的随机写操作不会引起额外的写放大, 因此可以显著提升瓦记录 RAID 系统的性能; FT-buffer 新型的基于可覆盖写磁道的双层结构具有两方面的优势, 定位在可覆盖写磁道的 FT-buffer 上层空间可以存储频繁更新的热数据, 利用可覆盖写磁道的随机写能力减少写放大; 瓦记录磁道构成的高密度下层空间可以在同样的空间条件下提高写缓冲区的存储容量, 为更多的写数据提供“二次机会”, 减少瓦记录磁盘的写放大现象.

(3) 经过真实企业级访问记录的测试, FT-RAID 能够显著降低写放大, 平均比 SMR RAID 5 方法要好 80.4%. 不仅 FT 的结构在针对瓦记录时能发挥效果, 这种在交错堆叠的复杂结构中寻找特殊磁道的思想, 对其他结构的新型高密度磁记录的研究也有启发作用.

本文第 1 节介绍瓦记录磁盘和 RAID 的相关研究背景, 并分析瓦记录 RAID 面临的主要挑战. 第 2 节描述提出的 FT-RAID 的基本思想. 第 3 节和第 4 节分别介绍 FT-RAID 中的 FT-mapping 和 FT-buffer 两个关键模块. 第 5 节展示 FT-RAID 的实验结果并进行分析. 第 6 节介绍相关研究工作. 最后, 第 7 节总结全文.

1 研究背景

1.1 瓦记录磁盘

受到超顺磁效应的限制, 传统磁盘的存储密度将达到极限(1 Tb/in^2). 瓦记录在现有磁盘的硬件技术上做了一些改动来提高存储密度. 在磁盘中, 每一个盘片都有一个读磁头和一个写磁头. 由于向磁盘中写入信息时需要翻转比特, 写磁头需要制作得更宽以提供更强的磁场. 通过利用磁盘中读磁头的宽度小于写磁头这一特质, 可以让磁道之间部分重叠, 磁道宽度减小到与读磁头宽度相同. 这样一来, 就可以保证读磁头能准确无误地读取数据, 同时在同样的盘片上可以放置更多数据量的磁道, 从而提高存储密度.

如图 1 所示, 左图为传统磁记录(conventional magnetic recording, CMR)方式的结构示意图. 在 CMR 内部, 磁道的宽度要大于读磁头, 基本等于写磁头宽度, 以保证每一次写数据时不会影响其他磁道, 也不会在读取数据时读到邻近磁道的信息. 图1的右图则展示了瓦记录方式的结构. 瓦记录将磁道互相堆叠, 形成瓦片一样的结构. 每个磁道都保证暴露出一部分空间, 以允许读磁头正常读取数据.

由于磁道部分重叠, 瓦记录存在固有的写入限制, 这一限制是影响瓦记录性能的根本原因. 当向一个磁道写入数据时, 会把相邻磁道的数据修改, 以防止数据丢失[9−12]. 如图 1 的右图所示, 当磁盘试图向 6 号磁道写入数据时, 写磁头会处于 $W2$ 的位置, 同时覆盖 6 号磁道和 7 号磁道, 写入 6 号磁道的数据会同时会存储到 7 号磁道中. 因此, 为了防止原来存储于 7 号磁道的有效数据丢失, 瓦记录磁盘会在写入之前, 将可能影响到的磁道内的数据全部读取到缓冲区中(一般为 RAM), 在缓冲区中将新的数据写入, 然后再全部刷回磁盘的相应位置. 这一过程称为读改写(read-modify-write, RMW),而这一过程会引起写放大(write amplification, WA). 文献[13]提出瓦记录的写放大问题主要由随机写操作和数据更新操作引起. 如果不做限制, 对一个磁道的写

入可能经过RMW过程影响磁盘上后续的全部磁道. 为减小RMW操作波及的磁道范围, 瓦记录磁盘将若干个磁道划分为一个带, 每个 Band 之间空出一些不使用的磁道, 这些空出的区域被称为隔离区(guard region). 这样的设计保证了所有写覆盖产生的影响都局限于 Band 内部, 减少了每次写入时覆盖的磁道数量. 例如, 图 1 右图中的磁道 0 至 4 号构成 0 号 Band, 磁道 5 至 9 号构成 1 号 Band, 两个 Band 之间会有一个隔离区. 这样, 当写入 1 号磁道时, 只会影响后续的 2–4 号磁道, 而不会对 5–9 号磁道产生任何影响.

瓦记录磁盘可以分为以下 3 种类型: 驱动管理型瓦记录(drive-managed SMR, DM-SMR)、主机管理型记录(host-managed SMR, HM-SMR)和主机感知型瓦记录(host-aware SMR, HA-SMR).

- 驱动管理型瓦记录在硬件层面内置了瓦记录转换层(shingled translation layer, STL), 以实现地址转换, 使得对上层的文件系统的兼容. 与固态硬盘的闪存转换层(flash translation layer, FTL)类似, STL 处在上层操作系统与底层硬件之间, 用于接受并转换上层发出的读写请求. 因此, 驱动管理型瓦记录可以在不做改动的情况下直接兼容现有的操作系统和文件系统, 这是其最大的优点. 其缺点在于全部转换的过程由不可见的硬件实现, 无法预测在不同工作环境下的性能表现.
- 主机管理型瓦记录允许主机系统感知瓦记录磁盘内部结构. 由于瓦记录具有叠瓦式限制, 随机写入操作不能直接进行, 所以就要求在主机系统层面实现 STL 或近似的功能, 以实现地址转换的功能. 主机管理型瓦记录的最大优点在于主机掌握的信息最丰富, 可根据工作环境的不同调整管理策略, 以提高整体系统的效率. 这种硬件并不能直接在常见的系统上部署, 而是需要使用区域块指令集(zone block commands, ZBC)[14]. ZBC 是国际信息技术标准委员会提出的基于主机管理型瓦记录的接口标准. 该标准提供了一系列以区域(zone)为粒度的操作, 包括查看 Zone 的类型和大小、打开和关闭 Zone、读取 Zone、写入 Zone 和复位 Zone 内写指针. 为了支持需要相应指令集的硬件设备, 目前日立公司已经实现 C 语言的 libzbc 库.
- 主机感知型瓦记录是驱动管理型瓦记录和主机管理型瓦记录的整合体, 兼具二者的特性. 它既可以在已有的文件系统上直接部署, 又可以将控制权交由主机, 由主机系统掌控读写策略.

1.2 磁盘阵列

磁盘阵列, 即 redundant array of independent disks (RAID), 是一种由多块磁盘构成、逻辑上表现为统一接口的存储方案, 目的是在容量、性能和可靠性之间取得平衡. 磁盘阵列存在不同级别, 如 RAID 0、RAID 1、RAID 01、RAID 10、RAID 4 和 RAID 5 等. RAID 4 将用户数据平均分布在数据盘中, 并将用户数据生成的校验信息存储在校验盘中. 当其中一块磁盘发生故障时, RAID 4 系统可以通过其余硬盘的信息恢复故障盘的数据. RAID 5 在 RAID 4 基础上做了改动, 将校验信息平均分布在每块盘上. 由于良好的读写性能、容单盘错特性和较大的容量, 大规模存储系统中广泛部署 RAID 5.

在 RAID 5 中, 存储数据时以条带(stripe)为基本单元. 条带是一个逻辑概念, 包含各个磁盘内对应区域的一个数据块(chunk). 图 2 展示了一个由 5 块磁盘构成的 RAID 5 磁盘组的例子, 这个磁盘组由若干个跨越 5 块磁盘的条带构成. 其中每一个条带都由 5 个数据块组成, 这 5 个数据块分布在五块磁盘的相同逻辑位置. 例如图 2 中 0、1、2、3 号数据块就构成了一个条带, *P*0 为这个条带的校验块. 每当更新条带中的任何一个数据块时, 都会引起对条带内校验块的同步修改, 所以 RAID 5 的实际写入量为逻辑写请求量的两倍. 比如修改数据块 9 时, 会伴随产生对校验块 *P*2 的修改请求. 修改数据块 7 时, 会同时修改校验块 *P*1. 换句话说, RAID 5 磁盘组本身自带 2 倍的写放大率. 由于 RAID 5 本身具备并发读写的特点, 读写请求能够同时下发至各块磁盘中, 即使 RAID 5 需要承担更高的读写请求数量, 其依然能够提供很高的性能.

1.3 瓦记录RAID的挑战

由于瓦记录磁盘具有很高的存储密度, 将会逐渐替代现有磁盘技术, 成为大数据存储的主流存储介质. 另一方面, RAID 在磁盘存储中具有非常广泛的应用, 具体提高数据可靠性和并发访问性能的双重优势. 因此, 在大数据存储时代, 基于瓦记录磁盘的 RAID 将逐渐成为企业大数据存储的主要形式之一.

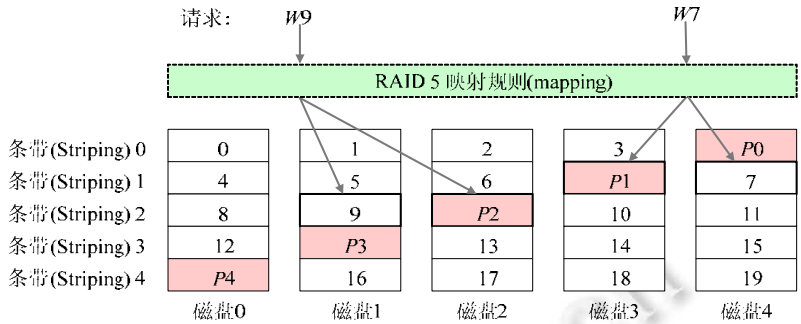

图 2 RAID 5 数据地址映射示例图

但是, 由于瓦记录磁记录方式内在的写放大性质, 以及RAID中校验数据引入的额外数据更新操作, 会使瓦记录 RAID 遇到更为严重的写放大现象, 影响大数据存储系统的性能. 根据 Liu 等人的研究, 在面对空间局部性不好的磁盘访问时, 与传统 RAID 5 相比, 瓦记录 RAID 5 性能最多相差 33 倍[15]. 我们初期的模拟测试显示, 瓦记录 RAID 5 的写放大可以达到传统情况的 23 倍. RAID 5 中的奇偶校验块经常被更新, 原因在于当一个条带中的任何块被更改时, 奇偶校验块都必须更新. 在瓦记录上构造 RAID 5 磁盘组, 每当数据块更新时, 都会伴随着奇偶校验块的更新, 瓦记录要处理的写请求量翻倍, 导致瓦记录的写放大倍数至少是不部署 RAID 5 时的两倍. 严重的写放大会造成瓦记录 RAID 磁盘组的性能显著下降, 成为其广泛应用在大数据存储系统的主要障碍和挑战, 迫切需要新的方法能够解决严重的写放大问题.

目前已有相关工作主要围绕单个瓦记录磁盘进行设计和优化[16−19], 还有部分工作探讨不同上层应用或数据结构与瓦记录结合时的场景[20−23]. 另外, 对传统磁盘的顺序写入的优化也对瓦记录的优化有着参考价值[24,25]. 对于瓦记录磁盘构成的 RAID 系统的优化, 只有少数工作提出在瓦记录磁盘组中引入一些 SSD 或 CMR 磁盘构建混合存储来加速[26−29], 非常缺少在 RAID5 系统上进行数据映射等基本方法层面的研究和优化工作, 这正是本文研究工作的定位.

2 基于可覆盖写磁道的 RAID 系统(FT-RAID)

观察瓦记录的内部结构, 我们发现传统瓦记录内仍然有一些资源没有被充分利用. 如图 1 所示, 瓦记录内相邻的 Band 之间设置隔离区的初衷是避免引起写放大的级联影响, 覆盖过多的磁道. 一个直观的想法是通过向隔离区写入数据来覆盖每个 Band 的第 1 个磁道. 例如, 如图 1 所示, 需要写入 Band 1 的 5 号磁道时, 可以将写磁头移得更高一些(如图 1 的 *W*2), 确保其仅覆盖 5 号磁道暴露出来的部分和 Band 之间的隔离区. 在这种情况下, Band 0 和 Band 1 内的所有其他磁道都不会被这次覆盖写操作影响. 每个 Band 的第 1 个磁道可以命名为“可覆盖写磁道(free track, FT)”, 例如图 1 中的 0 号磁道和 5 号磁道, 它们可以用于执行原位更新而不引起写放大, 适合用来处理随机写操作.

为了充分利用可覆盖写磁道便于进行随机写而不引起写放大的特点, 可以将 RAID 5 中的校验数据块强制映射到可覆盖写磁道来减少写放大. 由于 RAID 5 中每个条带中任意一个数据块更新, 该条带中的校验数据块都需要配套进行更新, 因此, RAID 5 中的校验数据块属于频繁进行随机写的数据. 如果将 RAID 5 中的校验数据块放到上述可覆盖写磁道上, 那么 RAID 5 校验数据块的任何更新, 都不会引起瓦记录层面的写放大, 这将非常有助于提升系统性能.

因此, 为了解决瓦记录 RAID 5 写放大严重的问题, 本文提出了一种基于可覆盖写磁道的 RAID 5 系统设计——FT-RAID. 能够在不改变现有瓦记录结构的情况下, 通过发掘“可覆盖写磁道”的潜力, 优化瓦记录 RAID 5 的存储方案, 以提高系统性能. 图 3 展示了 FT-RAID 系统架构. 如图 3 所示, 瓦记录磁盘空间会被划分为若干个 Band, 每个 Band 的第 1 个磁道都属于“可覆盖写磁道”, 可以支持随机写, 不会影响其他磁道; 其他磁道属于瓦记录磁道, 只能支持连续写入. FT-RAID 会把瓦记录磁盘上的 Band 分为两个空间, 绝大部分

Band 作为 RAID 数据的存储区域; 另一小部分 Band 是对用户不可见的持久化写缓冲区, 通过缓存随机写数据块, 减小写放大. 一般的产品中, 持久化缓冲区的 Band 数量约为数据存储区的 1%.

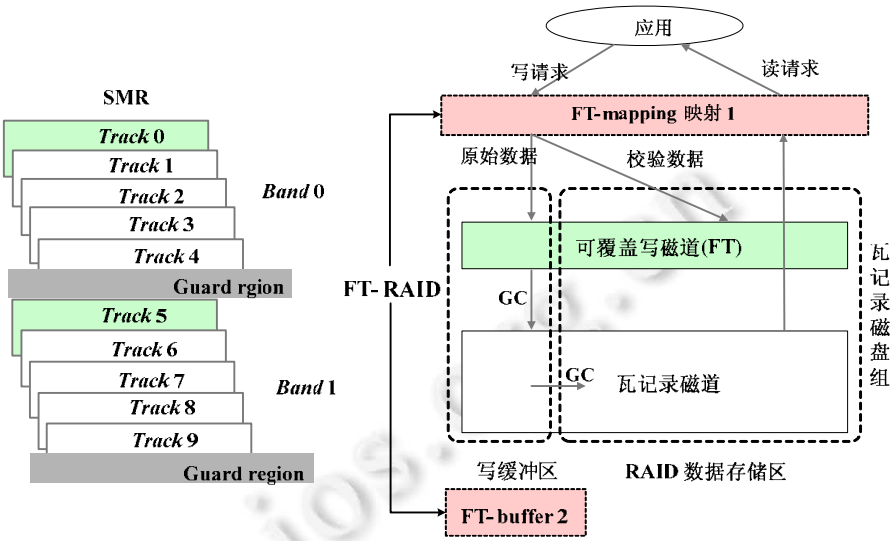

图 3 FT-RAID 架构图

如上所述, FT-RAID 主要包括两个部分: 一是面向可覆盖写磁道的 RAID 地址映射方法 FT-mapping, 二是基于可覆盖写磁道的持久化缓冲区管理方法 FT-buffer. 上层的应用程序下发的 I/O 请求会依次经过 FT-mapping 和 FT-buffer 的一系列处理, 最后存储到瓦记录磁盘中.

首先, FT-mapping 会将 I/O 请求作地址转换, 产生一个数据块地址和一个校验块地址, 具体产生方法在第 3 节中介绍. 随后, 数据块和校验块会根据不同的逻辑处理. 校验块会被映射到并直接写入可覆盖写磁道中. 由于可覆盖写磁道能够进行原位更新, 所以该过程不会影响其他磁道, 也不会产生额外的写放大. 数据块会先写入到 FT-buffer 的上层.

其次持久化写缓冲区采用 FT-buffer 方法进行管理. 持久化写缓冲区与 RAID 数据存储区的结构相同, 均由若干 Band 构成, 每个 Band 内部都有一个可覆盖写磁道和若干个叠瓦式磁道, 所有可覆盖写磁道构成缓冲区上层空间, 而叠瓦式磁道构成缓冲区下层空间. 当持久化缓冲区的上层写满时, 会触发垃圾回收(garbage collection, GC), 根据上层 GC 策略将数据块淘汰至下层. 当下层数据满时, 会根据下层 GC 策略将数据块淘汰回数据存储区的相应位置. FT-buffer 的具体细节会在第 4 节中详细介绍.

3 基于可覆盖写磁道的 RAID 数据映射方法(FT-mapping)

FT-mapping 负责将文件系统提供的数据原始逻辑块地址(logical block address, LBA)映射到瓦记录磁盘上的物理块地址(physical block address, PBA). 本节将详细介绍 FT-mapping 的具体映射方法.

3.1 FT-mapping概述

FT-RAID 以 RAID 5 的数据布局为基础, 最主要的区别从 RAID 5 确定的原始位置开始进行奇偶校验块的移位, 把所有奇偶校验块移到可覆盖写磁道. 在瓦记录磁盘内的所有 Band 里, 数据块都存放于叠瓦式磁道, 而校验块都位于可覆盖写磁道上.

与 RAID 5 不同, 在 FT-RAID 中, 可覆盖写磁道具有不影响其他磁道原位更新的特殊性质. 因此, 可以将频繁更新的奇偶校验块放在特殊的可覆盖写磁道. 在这种情况下, 奇偶校验块更新不会导致任何写放大, 因此能够明显减少总体写放大. 例如图 4 中, 磁盘 2 的 Band 上的块 2、6、13 和 17 按顺序依次存放在部分重叠的磁道上. 在该示例中, 除了奇偶校验块 $P0$ 外, 其他所有奇偶校验块(即 $P1$–$P4$)被向上移动到可覆盖写磁道.

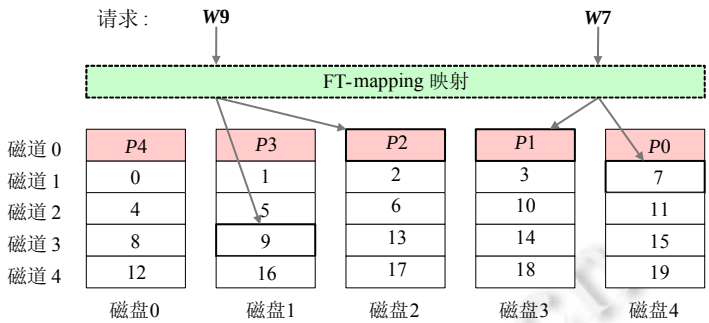

图 4 FT-RAID 中数据地址映射示例图

根据设计, 每个 Band 包含两部分区域: 可覆盖写磁道(free tracks)和叠瓦式磁道(shingled tracks). 可覆盖写磁道是一个对随机写友好的区域, 非常适合用于存储频繁更新的数据, 例如 RAID 奇偶校验块、写缓存中的热数据等. 叠瓦式磁道受到瓦记录重叠磁道引起的约束, 其中的数据必须以粗粒度单位(即 Band)进行更新. 可覆盖写磁道的大小占比由每个 Band 内包含的磁道数决定. FT-RAID 必须满足 Band 内磁道数等于磁盘数这一约束条件, 可以通过调整 RAID 磁盘组中磁盘数量或调整瓦记录盘中 Band 大小来实现. 假设瓦记录一个 Band 包含 5 个磁道, 可覆盖写磁道的总容量为整个 Band 容量的 1/5.

对于主机管理型瓦记录磁盘, 可以添加一个新操作(即 FTWrite)来扩展 ZBC 集. FTWrite 表示将数据写入 FT 区域中的逻辑块地址(LBA); 它与其他瓦记录操作的区别在于 FTWrite 不会将数据放入持久化缓冲区, 也不会在将来触发读改写操作. 相反, FTWrite 可以直接将数据写入或覆盖到可覆盖写磁道的目标位置上.

3.2 FT-mapping映射公式

根据前面介绍的 FT-RAID 的设计, FT-mapping 会将全部校验块数据映射到可覆盖写磁道来减小写放大, 这一映射关系由公式(1)–公式(10)说明. 公式中涉及符号的含义以及每个公式的意义, 本节给出详细解释.

如图 5 所示, 假设每个磁道能够存储 N_{col} 块数据, 对于每个逻辑地址为 LBA 的目标块 T, FT-mapping 的目标是计算出一个物理块的全部信息: 数据块和校验块的实际的磁盘号(I_{data} 和 I_{parity})和地址(A_{data} 和 A_{parity}). 为了计算出 A_{data} 和 A_{parity}, 首先要确定数据块所在的 Band 号(I_{band})、磁道号(I_{track})和列序号(I_{col}). 这些符号的具体含义在表 1 符号表中展示.

表 1 FT-mapping 符号表

符号	描述
N_{disk}	RAID 组中的总磁盘数
N_{track}	一个 Band 内包含的磁道数
N	一个磁道内包含的数据块的数量
I_{parity}	校验块所在的磁盘序号
I_{disk}	数据块所在的磁盘序号
I_{band}	目标块所在的磁盘内的 Band 序号
I_{track}	目标块所在 Band 内的磁道序号
I_{col}	目标块所在磁道内的块序号
n_{stripe}	根据 RAID 5 规则得出的目标块所在条带的序号
n_{chunk}	根据 RAID 5 规则得出的目标块序号
B	在瓦记录 Band 组内目标块的序号
A_{data}	数据块的物理地址
A_{parity}	校验块的物理地址

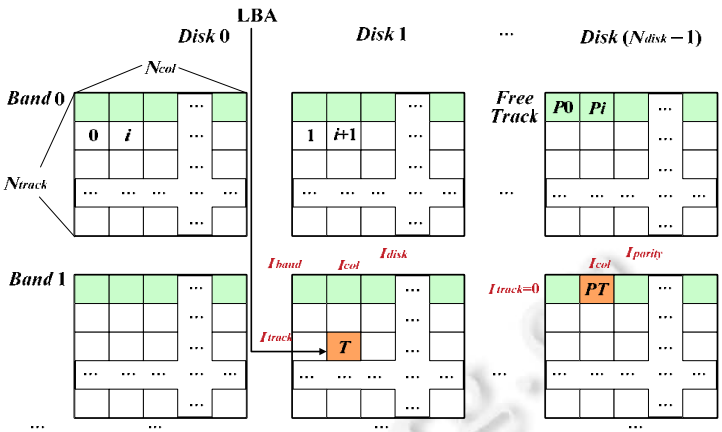

图 5 FT-Mapping 映射规则示意图

首先, Band 所在的位置 I_{band} 可以按照公式(1)进行计算, 其中, $N_{track}\times(N_{disk}-1)\times N_{column}$ 表示一个瓦记录 Band 组(由 N_{disk} 块磁盘构成)中总共含有的块数量. 用 LBA 整除总的块数量, 即可得到该 Band 所对应的序号.

$$I_{band}=\frac{LBA}{N_{track}\times(N_{disk}-1)\times N_{column}} \tag{1}$$

值得注意的是, FT-mapping 必须满足一个 Band 内的磁道数 N_{track} 等于 RAID 磁盘组内的磁盘数 N_{disk} 的限制. 由于瓦记录 Band 内含有磁道数的多少是由磁盘的读写磁头移动决定的, 可以灵活设置, 因此可以把 N_{track} 的值设置为 N_{disk}, 以满足这一约束.

然后, 把上面公式(1)的除法改为取模运算, 就可以得到目标块 T 在 Band 组中的偏移量, 计算方法由公式(2)给出.

$$B=LBA \bmod (N_{track}\times(N_{disk}-1)\times N_{column}) \tag{2}$$

利用公式(2)的结果, 在磁道中的列序号 I_{col} 可以由公式(3)确定. 列序号 I_{col} 意味着当前块 T 位于 Band 中的第几列.

$$I_{col}=\frac{B \bmod (N_{col}\times(N_{disk}-1))}{N_{disk}-1} \tag{3}$$

另外, 目标块 T 的条带号 n_{stripe} 和所在 chunk 的序号 n_{chunk} 可以被确定. 需要注意的是, 无论是传统 RAID 5 还是 FT-RAID, 这两个参数值都是一样的. 其中, n_{stripe} 表示目标块 T 处在 Band 内的第几行, 而 n_{chunk} 代表该块处在某个 *stripe* 中的第几份. 具体计算方式由公式(4)和公式(5)给出.

$$n_{stripe}=\frac{B}{N_{col}\times(N_{disk}-1)} \tag{4}$$

$$n_{chunk}=B \bmod (N_{col}\times(N_{disk}-1)) \bmod (N_{disk}-1) \tag{5}$$

校验块所在的磁盘号 I_{parity}, 可以由公式(6)确定. 同样, 这一参数与传统 RAID 5 一致.

$$I_{parity}=N_{disk}-1-n_{stripe} \tag{6}$$

接着, 数据块所在的磁盘号和磁道号(I_{disk} 和 I_{track})由公式(7)和公式(8)给出. 其中, 当条件 $n_{chunk}\geqslant I_{parity}$ 成立时, 意味着数据块 T 会移动至下一块磁盘, 原因是需要给校验块上移留出空间.

$$I_{disk}=\begin{cases}n_{chunk}, & n_{chunk}<I_{parity}\\ n_{chunk}+1, & n_{chunk}\geqslant I_{parity}\end{cases} \tag{7}$$

$$I_{track}=\begin{cases}n_{stripe}+1, & n_{chunk}<I_{parity}\\ n_{stripe}, & n_{chunk}\geqslant I_{parity}\end{cases} \tag{8}$$

校验块的位置与数据块相似. 如图 4 所示, 校验块 PT 的磁道号永远为 0(因为校验块必须位于可覆盖写磁道), 而 Band 号和列序号与数据块完全一致(即分别等于 I_{band} 和 I_{col}).

最后, 数据块 T 和校验块 PT 的地址可以被确定, A_{data}、A_{parity} 分别代表数据块和校验块所在磁盘内的总偏移量. 它们的计算方式如公式(9)和公式(10)给出.

$$A_{data}=(I_{band}\times N_{track}+I_{track})\times N_{col}+I_{col} \tag{9}$$

$$A_{parity}=I_{band}\times N_{track}\times N_{col}+I_{col} \tag{10}$$

按照以上方法, 就实现了 FT-RAID 中逻辑地址到物理块地址的映射. 给定任意一个逻辑地址 LBA, 最终都能得到数据块和校验块的位置信息: I_{disk}、I_{parity}、A_{data} 和 A_{parity}.

4 基于可覆盖写磁道的瓦记录缓冲区管理方法(FT-buffer)

瓦记录磁盘中一般都设置对用户不可见的持久化写缓冲区, 用户的写入数据先临时存储于写缓冲区, 然后在合适的时机一批一批地将同一 Band 的数据刷回磁盘, 以减少总体的写放大. 如果没有写缓冲区, 用户写入数据直接写入由于无法直接在目标 Band 上进行随机写操作, 需要触发代价非常高的“读取-修改-写回(read-modify-write, RMW)”操作, 造成比较明显的写放大.

传统瓦记录通常使用传统磁盘结构(即非叠瓦式)缓冲区, 以支持缓冲区内的随机写操作; 这样当对一个数据块进行充分写入时, 可以直接进行写覆盖. 基于前面的讨论, 可以了解到在相同的空间内, 瓦记录的存储容量大于传统磁盘, 尽管瓦记录磁道的随机写入并不自由. 受到可覆盖写磁道的启发, 在瓦记录内部也能有支持可随机写的区域. 因此, 为了增大缓冲区的空间, 以减少瓦记录磁盘的写放大率, 本文提出了一种新的基于叠瓦式结构的写缓冲区管理方法——FT-buffer.

FT-buffer 管理的也是一个瓦记录磁道的区域. 结构上, FT-buffer 与非缓存区域完全一致, 也由若干个 Band 和它们之间的隔离区组成. 每个 Band 内部, 也同样包括瓦记录磁道和可覆盖写磁道. 在功能上, FT-buffer 与非缓存区域不同, 其用于接收和处理文件系统和上层应用发出的请求. 如图 3 所示, 一个 FT-buffer 在逻辑上分为上层空间和下层空间. 上层空间由缓冲区所有 Band 的可覆盖写磁道组成, 它们在逻辑地址上连续. 例如在写入数据时, 会先写入 0 号 Band 的可覆盖写磁道, 接着再写入 1 号 Band 的可覆盖写磁道, 以此类推. 上层空间相对下层空间来说较小, 但是可以支持随机写操作而没有写放大. FT-buffer 的下层空间是由缓冲区所有 Band 的瓦记录磁道构成, 因此不能支持小粒度的随机写操作, 必须整个 Band (除去可覆盖写磁道部分)整体执行写操作. 下面将分别介绍 FT-buffer 上层空间和下层空间的管理方法, 以及 FT-buffer 的写放大分析.

4.1 FT-buffer上层空间管理

由于 FT-buffer 的上层空间可以支持随机写操作, 因此用户的数据写入先写入 FT-buffer 的上层. 但是由于可覆盖写磁道较少, FT-buffer 上层空间非常有限; 当上层空间满了之后, 需要将其中部分数据淘汰到空间更大的 FT-buffer 下层. 其中最为关键的策略是上层空间的垃圾回收(garbage collection, GC)算法, 目标是尽可能减小瓦记录磁盘的写放大率.

FT-buffer 上层空间的垃圾回收过程(FT-GC)的策略主要是尽可能释放更多的上层空间, 同时尽可能减小将来缓冲数据写回瓦记录磁盘时的写放大率. 由于FT-buffer下层都由瓦记录磁道组成, 需要以Band为单位进行整体更新, 因此 FG-GC 将从上层数据块以其在 RAID 数据存储区域的物理地址所在 Band 为单位进行排列, 优先选择包含最多缓冲区数据块的一个或多个 Band, 淘汰其中的所有上层数据块. 这些数据块将一次性写入下层空间的一个空闲 Band 中. 由于这些数据块都属于同一个或尽可能少的 Band, 未来下层数据写回 RAID 数据存储区域时, 将触发尽可能少的瓦记录磁盘写放大.

图 6 展示了 FT-buffer 的上层空间释放的一个具体实例. 首先, 当用户发出了若干写入请求时, 最终目的地为不同 Band 的数据(如图中的 $bd1$、$bd2$ 等)写请求发出时, 会首先存入 FT-buffer 的上层. 这些数据存储在

不同 Band 的可覆盖写磁道上. 由于系统和应用发出的请求具有局部性, 这些最新写入的数据短时间内有可能被再次写入, 可覆盖写磁道能提供无写放大的随机写覆盖操作.

当上层数据满时, 系统会触发 FT-buffer 上层的垃圾回收过程(FT-GC). 值得注意的是, 与传统的垃圾回收过程不同, FT-GC 不会将被淘汰的数据直接刷回磁盘的相应位置, 而是采用“二次机会”的策略, 将这些数据先存储在下层中. 最后, 当下层也装满时, 叠瓦式磁道垃圾回收(shingled-GC)才会将这些数据刷回磁盘的相应位置. FT-GC 的具体工作过程分为 3 步: (1) 目标区域的选择; (2) 淘汰块的选择; (3) 写入目标区域. 下面将结合图 6 所示的例子来阐述 3 个步骤的具体工作.

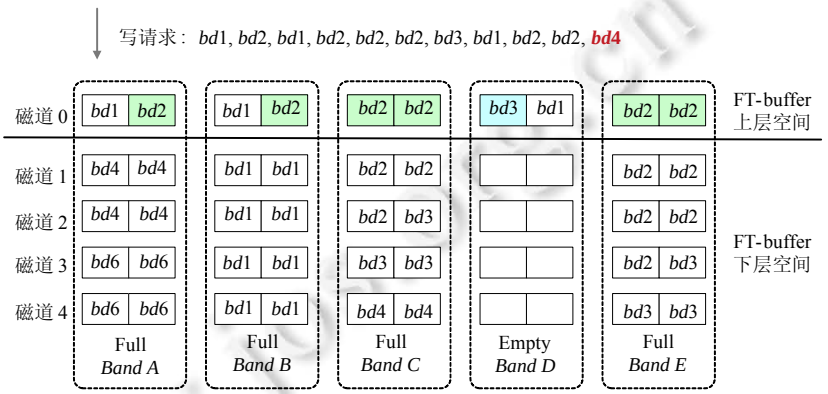

图 6 FT-buffer 上层空间的 GC 操作示例

如图 6 所示, 假设在执行一系列的写请求后, FT-buffer 的上层空间已满. 当请求 *bd4* 到达时, 由于上层已经没有空间接收, 此时会触发 FT-GC. 首先, 系统会从缓冲区的下层内任意选择一个空的 Band 作为存储的空间, 图 6 中的 Band *D* 在此时被选中. 然后, 会选择 FT-buffer 上层空间中地址属于同一个 Band 数量最多的块, 作为即将淘汰的数据. 这么做主要基于两方面的考虑. 其一, 是尽可能地释放更多的 FT-buffer 上层空间, 以减少 FT-GC 被触发的次数. 其二, 是将同一 Band 的数据聚集在一起“打包存储”, 为将来集体写入提供支持, 以减少写放大. 图中上层中 Band 2 的数据最多, 因此 6 个位于 Band 2 的块全部被选中. 由于还剩下 2 个位置, Band 3 的 1 个块被选中(因为 Band 1 的块共有 3 个, 无法全部放入, 因此没有选择 Band 1). 最后, 将上层被选中的 7 个块写入下层的第 4 个 Band 中. FT-GC 过程结束, 成功释放了上层的缓冲空间, 同时, 同一 Band 的数据被成块地存储在叠瓦式磁道中, 有利于进一步的整理.

4.2 FT-buffer下层空间管理

经过上述 FT-GC 的过程, 下层的瓦记录磁道会逐渐填满, 当 FT-buffer 下层叠瓦式磁道区域的空间满了, 会通过 Shingled-GC 操作将其中数据淘汰回到瓦记录磁盘原始的存储位置, 释放 FT-buffer 下层空间. Shingled-GC 的具体工作过程分为两步: (1) 淘汰 Band 的选择; (2) 写回磁盘数据存储区. FT-buffer 下层淘汰数据时, 会选择整个 FT-buffer 下层中 Band 号相同的数量最多的数据块作为淘汰的依据.

如图 7 所示, 假设在下层的叠瓦式磁道中, 各 Band 号的数量如图 7 右边数字所示(*bd*1:8, *bd*2:14, *bd*3:7, *bd*4:6, *bd*6:4), 此时将选择拥有 14 个数据块的 Band 2 作为本次淘汰的数据. 由于下层需要以 Band 为单位淘汰, 所以含有 *bd*2 的 Band *C*、Band *D* 和 Band *E* 会被选中; 而且其他未包含位于 Band 2 数据块的 Band 不在本轮 Shingled-GC 范围内(如图 7 中的 Band *A* 和 Band *B*). 接下来, 这 3 个选中的 Band 里的全部数据(分别是 14 个来自 *bd*2 的数据块、7 个来自 *bd*3 的数据块和 2 个来自 *bd*4 的数据块)都将会被刷回数据存储区的相应位置, 这一过程将执行 RMW 操作. 这样的淘汰策略能够显著减小写放大. 由于 Band2 是整个下层中最多的数据块写回的目标, 所以在这一次写回过程里, 额外写入数据的比例降低, 写放大也随之降低.

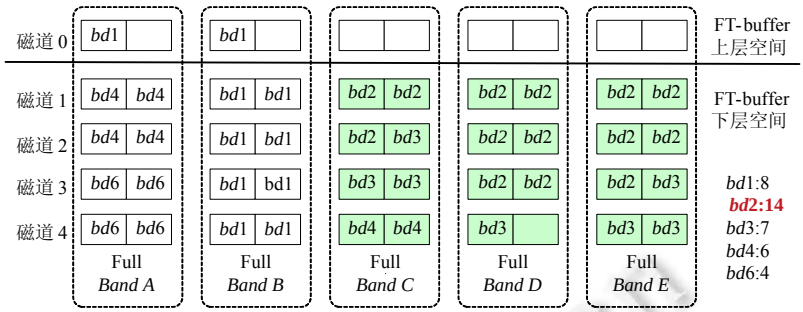

图 7 FT-Buffer 下层空间 GC 操作示例

4.3 FT-buffer写放大分析

与基于传统非瓦记录磁道的持久化写缓存区方式相比, FT-buffer 能够减少整个瓦记录磁盘的写放大率, 原因有三: 其一是获得了更多的缓冲区空间. 由上面的讨论知道, 下层拥有更高的存储密度. 缓冲区的空间更大, 意味着能够装载的数据量也更大. 这样一来, 触发 GC 的总次数减少, 在淘汰数据时, 也更容易聚集位置相近的数据一同写回. 其二是拥有原位更新的能力. 尽管只有最上方的可覆盖写磁道能够执行原位更新, 但由于数据具有局部性, 频繁修改的数据能够原位更新, 缓冲区填满的速度减慢, GC 的次数减少. 其三是对淘汰数据的"二次机会". 对于传统瓦记录的缓冲区而言, 被淘汰的数据会被直接写回原位.例如, 当缓冲区中只有一个数据块来自 Band 3, 那么这个数据块也必须要经过 RMW 过程写回数据存储区的相应位置. 如果一个 Band 的大小是 5 MB, 一个数据块大小为 4 KB, 那么这一过程引起的写放大就等于 5 MB/4 KB=1.25K, 磁盘需要多承担 1.25K 倍的负载. 而采用 FT-buffer 的结构, 这一来自 Band 3 的数据会被存储在下层内. 将来触发 Shingled-GC 时, 下层内其他来自 Band 3 的数据会与其一同写回, 写放大能明显减少.

为了量化写放大情况, 可以用如下公式估算 FT-buffer 减少写放大的效果. 公式(11)说明了传统方案中写放大, 其中数字 2 表示数据块和校验块加在一起一共是两倍的写入量, wa 表示未配置 RAID 5 时瓦记录的写放大. 公式(12)说明了 FT-buffer 引起的写放大, 其中, $1_{(parity)}$表示一个校验位直接写入磁盘需要一次写操作, $1_{(data)}$表示一个数据位写入 FT-buffer 可覆盖写磁道需要一次写操作. 在 FT-buffer 中, 数据从上层淘汰至下层时需要多执行写操作, 由于在写入数据时存在缓存命中, 所以平均会引起 k 次写操作(k 小于等于 1). 因为叠瓦式磁道的空间比传统缓冲区空间大, 所以这一空间的基本写放大 wa' 小于 wa. k 次写操作乘上$(1+wa')$表示 FT-buffer 两层总共带来的写放大. 比较这两个公式, 可以发现传统 RAID 5 的写放大会倍增, 而 FT-RAID 的写放大只会以加法的形式提高. 因此, FT-RAID 能显著降低整体系统的写放大.

$$WA_{\text{RAID 5}} = 2_{(data+parity)} \times wa \tag{11}$$

$$WA_{\text{FT-RAID}} = 1_{(parity)} + 1_{(data)} + k(1+wa'),\ k \leqslant 1 \tag{12}$$

5 实 验

5.1 瓦记录磁盘模拟器

本文中提出的 FT-RAID 是一种新的瓦记录盘上的数据组织和管理方法, 而目前市场上主要存在的瓦记录盘都还不支持本文提出的可覆盖写磁道, 以及相应的优化方法. 因此, 本文在实际的传统磁盘之上构建了一个能够支持可覆盖写磁道基本操作的瓦记录模拟器, 依照已有论文研究中探测的瓦记录内部结构和读写逻辑, 利用主机程序全面模拟瓦记录的读、写、地址映射、缓存管理等各种行为; 并拓展了现有的瓦记录指令集 ZBC. 为了实现对可覆盖写磁道的写操作, 需要在 ZBC 中新增了一个操作: FT-write. 与其他写入操作不同, FT-write 会将相应数据写入可覆盖写磁道. 由于可覆盖写磁道支持原位更新操作, 所以 FT-write 不会引起额外的写放大.

在模拟器的参数配置上, 一个磁道的大小被设置为 1 MB. 在瓦记录内部, 一个 Band 包含 5 个磁道, 总大小 5 MB, Band 之间空出一个磁道作为隔离区(即 1 MB). 每个 Band 内的第 1 个磁道就是可覆盖写磁道, 可以支持随机写操作.

在模拟实验中, 一共设置了两种缓冲区结构: 传统缓冲区结构(persistent buffer, PB)和 FT-buffer. PB 采用传统磁记录(CMR)磁盘结构, 整个缓冲区都支持随机写操作. FT-buffer 结构如上文介绍, 只有可覆盖写磁道支持随机写操作. 需要注意的是, FT-buffer 的总空间是 PB 的 5/3(相同空间下叠瓦式结构能够获取更多存储量, 这一设置能保证比较的公平性). 同时根据实验的惯例, 传统缓冲区 PB 的大小设置为 workload 不重复访问量的 1%.

5.2 实验环境

为了实现瓦记录磁盘模拟器, 部署了五块传统磁盘(西部数据 WD10EZEX 1 TB), 并在上层部署了 FT-RAID 系统. 硬件参数和模拟器设置见表 2. 所有的实验都在一台装载 Linux 系统的机器上进行. 操作系统内核版本为 Linux3.10.0-1062.el7.x86-64 (CentOS 7), 内存大小 8 GB, 同时开启 PageCache.

本文采用了微软剑桥研究院提供的真实 I/O 访问记录(trace). 其中, 每个访问记录都通过地址平移和叠加的方式放大了 10 倍, 以保证测试数据的规模. 本次实验出现的 10 个访问记录的名称和特性都列在表 3 中. 由于瓦记录的性能主要与写请求有关, 所以需要关注写请求的比例. 这些 trace 中, 写请求占比的范围在 28%–89%, 覆盖了读主导、写主导和读写平衡的应用情景.

表 2 瓦记录 RAID 模拟器参数

参数	描述
CMR 磁盘	西部数据 WD10EZEX 1 TB
Block 大小	4 KB
Track 大小	1 MB
Band 大小	5 MB
写磁头宽度	2 磁道
CMR 缓冲区大小	该 trace 实际访问量的 1%
FT-Buffer 大小	5/3 倍的 CMR 缓冲区大小

表 3 实验用到的 I/O 访问记录信息

访问记录名称	写请求占比(%)	请求总量	实际访问量(GB)
*hm*_0	67	89 854 870	23.29
*mds*_0	70	29 166 620	31.21
*prn*_0	80	176 357 660	148.26
*rsrch*_0	89	32 542 780	3.58
*src*1_2	83	140 248 600	19.87
*stg*_0	68	60 986 670	63.66
*ts*_0	74	42 164 570	9.13
*usr*_0	28	128 732 740	24.68
*wdev*_0	73	26 548 240	5.27
*web*_0	46	96 423 980	73.08

本次实验共比较 4 种方案, 它们的名称和配置见表 4. 对于 RAID 5, 我们采用传统 RAID 5 的映射规则、CMR 结构的 PB 和 FIFO 的淘汰策略. 后面 3 种方案都采用 FT-mapping 的映射规则. 不同的是, FT-C-F 采用 CMR 结构的 PB 和 FIFO 的淘汰策略, FT-C-M 采用 CMR 结构的 PB 和 MOST[30]的淘汰策略. 其中, FIFO 是瓦记录磁盘 PB 中最为常用的缓存淘汰算法, 每次淘汰最早到来的数据; MOST 是专门针对瓦记录磁盘提出的、有助于减小其写放大率低缓存淘汰算法, 每次淘汰包含相同 Band 号最多的一组块. 而 FT-RAID 则采用上文介绍的 FT-buffer 方法.

本文的实验基于 10 种来自企业的实际 I/O 访问记录进行测试, 比对 5 种方案在每一个 trace 文件(即一种工作环境)下的性能. 性能的评价指标有 3 种, 即 I/O 时间(I/O time)、写放大率(write amplification rate)、触发的垃圾回收次数(GC count).

表 4 比较对象信息

RAID 方案	RAID 地址映射方法	持久化写缓存(PB)管理方法
RAID 5	RAID 5 Mapping	CMR PB+FIFO
FT-C-F	FT-mapping	CMR PB+FIFO
FT-C-M	FT-mapping	CMR PB+MOST
FT-RAID	FT-mapping	FT-buffer
CMR	RAID 5 Mapping	N/A

5.3 实验结果与分析

5.3.1 整体性能比较

这一部分选择了 3 个典型的 I/O 访问记录作为代表, 其中, *rsrch*_0 代表写主导的 trace, *usr*_0 代表读主导的 trace, *web*_0 代表读写平衡的 trace. 上文提到的 5 种方案在这 3 种 trace 上的 I/O 时间如图 8 所示. 其中, CMR 相对 SMR 而言具有存储密度小、能随机读写、价格高的特点, 在实验配置时 CMR 与 SMR 的容量相等, 意味着现实中使用 CMR 磁盘需要付出更高额的成本. FT-RAID 在不同 trace 下均表现优于除 CMR 外的 3 种方案, 性能与 CMR 相近. 与传统的 RAID5 方案相比, FT-RAID 提高了 1.46–11.15 倍的总体性能. 由于 FT-RAID 设计的主要目的是提高写请求的性能, 所以性能提高的比例与 trace 中写请求的比例有关, 写请求占比较大的 trace 中 FT-RAID 的优化程度高. 同时注意到, 除了传统 RAID5 方案和 CMR 方案, 余下 3 种方案由于均采用了可覆盖写磁道的设计和 FT-mapping 的映射方法, 所以它们的性能都优于传统 RAID 5. 在我们的 3 组 trace 测试中, SMR RAID 5 的 I/O 任务完成时间是 CMR RAID 5 的 1.79–10.42 倍. FT-RAID 的 I/O 任务完成时间减小为 CMR RAID 5 的 0.93–1.22 倍, 性能相对于 SMR RAID 5 有了显著提升, 已经达到接近 CMR 的性能.

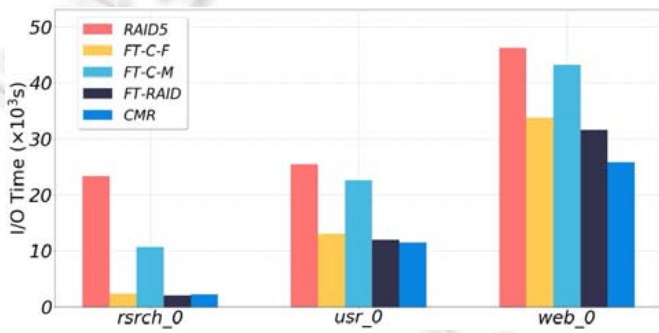

图 8 I/O 时间比较

5.3.2 写放大分析

CMR RAID 5 的写放大固定为 2 倍, 相当于基于瓦记录磁盘的 RAID 5 方案优化的上限, 越接近 2 倍就表示写放大优化效果越好. 为了解释上述性能测试的结果, 图 9 展示了 CMR 方案以外的四组方案在各组测试下的写放大率结果, 如图 9 所示. 对于全部 10 组用于测试的 trace, 可以发现传统瓦记录磁盘 RAID 5 拥有最大的写放大率和最差的整体性能. 而 FT-RAID 在 8 组 trace 中拥有最低的写放大率, FT-C-M 在另外两组中有最低的写放大率. 比较写放大率的平均减少量, FT-RAID 比 RAID5 减少了 80.42%, 比 FT-C-F 减少了 63.81%, 比 FT-C-M 减少了 26.12%. 值得注意的是, 即使 FT-C-F 的写放大率比 FT-C-M 更大, 但 FT-C-F 的整体性能更佳. 原因在于 MOST 算法虽然能够有效地减少写放大, 但 FIFO 总是能淘汰旧的数据, 保证 PB 能够持续接受连续 I/O. 因为连续 I/O 的速度比随机 I/O 快两个数量级, 所以 FT-C-F 总体达到了更好的性能.

回顾前文提到的可覆盖写缓冲区结构, 由于采取了叠瓦式结构, 所以 FT-RAID 的缓冲区空间是 CMR 结构缓冲区的 5/3, 能够容纳更多的数据. 更大的容量意味着在执行垃圾回收时, 可以在更大的数据范围内选择 Band 号相同的数据块, 以进行大批量的刷回, 带来更小的写放大. 另外, 缓冲区内的双层结构给到了上层被淘汰数据“二次机会”, 数据块在刷回时拥有再次组织的可能, 这让 FT-RAID 在多数 trace 的测试结果中达到了最低的写放大.

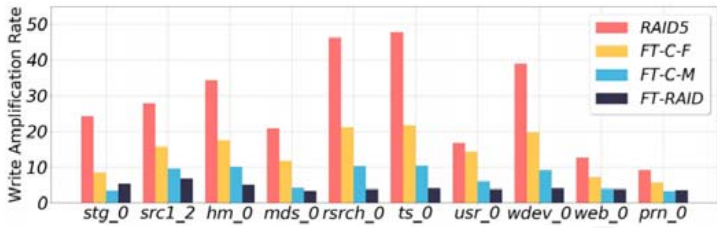

图 9 写放大率比较

5.3.3 垃圾回收次数分析

CMR 不需要缓冲区就可以进行随机写操作, 因此不存在垃圾回收过程. 其余 4 种方案的缓冲区触发的垃圾回收次数统计如图 10 所示. 很明显, 与其他 3 种方案相比, FT-RAID 的垃圾回收次数在全部 10 组实验中均达到了最小. 注意, FT-RAID 的垃圾回收次数只统计了下层的垃圾回收(由于持久化缓冲区上层的垃圾回收对写放大影响很小, 具体影响参见上文的公式(12)). 对于 *rsrch*_0 这一 trace, FT-RAID 方案的垃圾回收次数相比 FT-C-F 和 FT-C-M 均减少了至少 50%, 所以 FT-RAID 拥有更低的写放大率. FT-RAID 给予了所有缓冲区内被淘汰的块“二次机会”, 允许其暂时被存储于下层, 因而可以减少垃圾回收的次数、增加每次垃圾回收时找到同一 Band 的块的机会.

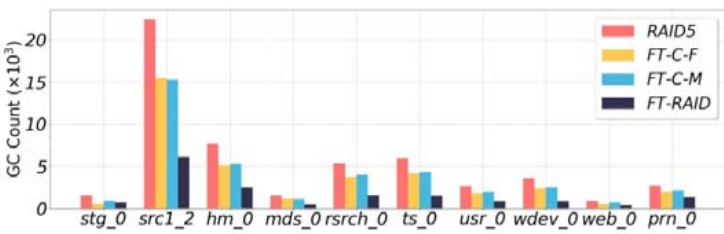

图 10 垃圾回收次数比较

6 相关工作

6.1 瓦记录磁盘相关研究

Abutalib 等人提出 Skylight[16]技术, 用于发掘瓦记录磁盘的性质. 与传统的纯软件测试技术不同, Skylight 额外加入了硬件技术, 通过在瓦记录磁盘盘片的顶部钻开的小口向内部植入一个微型高速摄影机, 拍摄磁盘内部磁头的移动情况. 在软件层面, 通过测量 I/O 操作的延迟等指标来了解磁盘的运转情况. 结合上述的软件、硬件测试结果, 作者推测出了驱动管理型瓦记录的性质. Skylight 揭示了瓦记录厂家未公开的多项指标, 包括持久化缓冲区的结构和性质、垃圾回收的机制、瓦记录转换层的映射策略和 Band 的大小. 这样软硬件结合的方式不仅揭开了瓦记录的“黑箱”, 也为后续基于瓦记录的研究铺平了道路. 其揭示的瓦记录结构和读写流程是本文开发的瓦记录磁盘模拟器的主要参考依据之一.

为了探究主机感知型瓦记录的性质, Wu 等人[17]基于主机感知型瓦记录样盘进行了测试, 得出了一系列结果. 主机感知型瓦记录具有独特的性质, 包括开放区域问题(open zone issue)和非连续区域问题(non-sequential zone issue). 根据作者对主机感知型瓦记录样盘(目前未商业化)的测试, 如果对上述两个参数的设置不合理, 会导致严重的性能下降.

SMaRT[18]是一个基于驱动管理型瓦记录的管理系统, 其主要创新点在于使用了磁道级别的管理策略. 与传统的基于数据块(block)的磁盘管理方案不同, SMaRT 将管理的粒度放大为磁道. SMaRT 采用了原位更新和异位更新的混合策略. SMaRT 还基于磁道级别的管理方案提出了划分冷热数据的优化方式. 磁盘内部划分出

冷磁道和热磁道. SMaRT 能够显著减少写放大率, 提高瓦记录的整体性能. 该文以磁道为单位的组织方式、冷热分离的存储策略非常具有启发性.

He 等人[19]提出了一种新型的瓦记录磁道映射机制, 提高了在低空间利用率时的读写性能. 该方案改变了逻辑块到物理块地址的映射方式. 该方案考虑了瓦记录的结构, 通过改变逻辑地址到物理地址的映射方案, 规避了瓦记录的内在缺点. 然而, 这一方案只在磁盘空间占用率较小(50%以下)时有明显的性能提升, 当磁盘空间占用率较大(超过 50%)时性能与传统方案差不多. 该论文说明了瓦记录中磁道的性质并不完全相同, 因此需要依据不同性质制定策略.

Caveat-Scriptor[20]描述了一种基于瓦记录的读写策略. 策略的重点在于为每个 Band 维护两个参数. 这两个参数分别是磁盘隔离距离(drive isolation distance, DID)和磁盘前置距离(drive prefix isolation distance). 通过这两个参数, 主机可以实现受限制的随机写入操作, 提高瓦记录磁盘的性能. 这两个参数表征了每个 Band 内部“不会覆盖有效数据”的写入情况. 这一方法相比传统的管理方式更加灵活, 松弛了叠瓦式结构的限制, 允许主机在合适的情况下进行随机写操作, 提高了系统的灵活性, 减小写放大率.

另外, Yao 等人[21]提出了一种基于主机管理型瓦记录和 LevelDB 的键值对(key-value)存储系统——GearDB. 作者将主机管理型瓦记录和 LevelDB 结合, 规避瓦记录的内在缺点, 使整个系统性能提升. 瓦记录对连续写操作和读操作非常友好, 这一性质与 LevelDB 相吻合. 作者提出一种新的 LSM-Trees 结构的存储方案和新的压缩(compaction)算法, 在不用垃圾回收的情况下也能保证数据被正确处理. 结果显示, GearDB 性能是传统的基于瓦记录的 LevelDB 性能的 1.71 倍.

在瓦记录的缓存优化方面, 文献[22]提出了一种基于 Flash-cache (FC)和瓦记录混合架构的缓存策略. FC 能够有效地提高性能, 在未触发垃圾回收时, 混合存储系统拥有接近 NAND flash 的性能. 同时, FC 可以减轻大规模垃圾回收总量, 并提升垃圾回收的性能. 实验显示, 在平均响应时间上, FC 比传统方法缩短了 73%, 垃圾回收的效率能够提升 11.1 倍.

文献[23]提出了 SW-B+tree——一种利用 B+树结构优化瓦记录索引管理的方法. 优化瓦记录性能的一个通常的方法是通过引入索引来管理写入的数据, 然而很少有论文提到对索引管理的优化. 如果索引管理的策略不佳, 索引的访问将成为整体系统的瓶颈. SW-B+tree 包含一个基于瓦记录的重定向地址的数据结构和与频率相关的垃圾回收策略. 实验表明 SW-B+tree 能够平均提高 55%的瓦记录存储性能.

6.2 瓦记录RAID相关研究

S-RAID5 和 Ripple-RAID 是针对顺序写操作对传统磁盘的优化. 由于顺序存储系统(如视频监控、虚拟磁带库)不需要很高的数据传输带宽, 当前的 RAID 方案存在能耗浪费和高磁盘损耗的问题. 文献[24]提出了一种针对顺序数据访问的节能磁盘阵列 S-RAID 5. S-RAID 5 采用局部并行策略, 将磁盘阵列中的存储区域分组, 以达到节能的目的. 文献[25]提出了一种面向连续数据存储的高效能磁盘阵列——Ripple-RAID. Ripple-RAID 综合运用了地址转换、异地更新等策略. 基于 80%顺序写操作负载的测试表明, 该方法的写性能比多数常见 RAID 策略好, 同时能耗最低.

HiSMRfs[26]是一个针对瓦记录的文件系统, 对瓦记录友好. 这一系统由若干瓦记录磁盘和一块固态硬盘构成. 该系统使用文件系统级 RAID 替代传统的驱动级 RAID. 文件系统级别的 RAID 可以在主机的文件系统层实现地址映射和重定向, 无需磁盘内部的参与. 由于元数据(如 Band 的使用情况)会频繁更新, HiSMRfs 决定使用 SSD 来存储元数据. 在结构方面, 为了加快元数据的查找速度, HiSMRfs 使用树状结构存储、管理元数据并使用哈希算法. 而普通数据会存储于瓦记录的 Band 内. HiSMRfs 也采用冷热数据分离的管理策略, 热数据会被安排在专门的 Band 内, 尽可能方便其原位更新. 文件系统级的 RAID 在文件层面实现条带. 一个文件会被拆解为若干个文件块, 存储于不同的磁盘中. 测试结果显示, HiSMRfs 在更节省 SSD 空间的情况下, 性能比 Flashcache 方案快 11%.

DVS[27]是一种针对瓦记录 RAID 的优化方案. 整体系统由若干块瓦记录磁盘和一块固态硬盘构成混合存储. 与传统 RAID 5 方案不同, DVS 改变了数据更新的规则. DVS 从不会原位修改数据, 相反, 它总会将更新的

数据写在一个新的条带里. 为了保证每次都能写满一个条带, DVS 在接收到数据更新操作时不会立即写入, 而是会等待一小段时间, 积攒一部分更新操作. 之后, 这一堆更新操作会被重构为新的条带, 根据 RAID 5 的校验规则生成校验块, 并以严格追加的方式将这一条带写入磁盘. 由于这个新条带改变了传统的地址映射规则, 所以这些数据的原地址位置会被记录下来. 同时, 为了减少写放大的开销, DVS 还设计了一种专为瓦记录服务的新的写缓冲区管理结构. 从测试结果看, 在全部 6 个 trace 中, DVS 的平均响应时间小于传统瓦记录 RAID 5.

RAID 4S[28]同样采用混合存储的方案来优化瓦记录 RAID. 与 DVS 不同, RAID 4S 采用两块传统磁记录与 3 块瓦记录构成 RAID4 磁盘阵列. 传统的 RAID 4 方案需要一块专门的磁盘存储校验位. 为了优化整体系统性能, RAID 4S 决定采用传统磁记录(CMR)作校验盘. RAID 4S 使用另一块传统磁记录, 来缓存数据盘的更新操作, 并将原来的地址位置记录下来. 这一方案虽能够提升磁盘阵列的性能, 但需要引入混合存储, 且比通常情况下的 RAID 4 还要多使用一块传统磁记录. 这样的混合方式不一定适用于企业的生产环境, 泛用性并不高.

文献[29]提出了一种基于捎带回收的瓦记录 RAID 5——PRaid 5. 为了减小瓦记录磁盘阵列引发的高昂写放大开销, PRaid 5 使用高速持久缓存持久化写请求, 利用循环日志管理瓦记录中的条带, 将随机写请求转化为顺序写, 并把相应信息存储在地址映射表中. 在处理小粒度的读操作时, 系统会顺带回收附近的无效数据, 以减轻垃圾回收过程引起的开销. 同时, 在出现单盘故障需要进行数据恢复时, 该方案可以避免对所有条带进行读写, 从而提高数据恢复的性能. 在 PRaid 5 和传统 RAID 5 的 6 组对比测试中, PRaid 5 的磁盘损坏重构数据量均小于传统 RAID 5. PRaid 5 在 3 组 trace 中的写带宽相对于 RAID 5 有提升, 最高达到 29.4%. 而在另外 3 组 trace 里, PRaid 5 出现写带宽性能下降, 对于局部性较差的 tracehm_0, 其写带宽下降达到了 38.1%.

目前已有工作中, HiSMRfs、DVS 和 PRaid 5 均使用了 SMR-SSD 混合存储, 依赖 SSD 作为缓存以提高读写速率; RAID 4S 在 RAID 4 的基础上, 利用传统磁盘构建了混合磁盘组来进行系统优化. 本文提出的 FT-RAID 不依赖于其他设备构建混合存储, 直接针对瓦记录 RAID 的核心方法进行设计, 因此不适合与已有工作进行直接比较. PRaid 5 对磁盘损坏重构有不错的优化, 但循环日志的写方式会引入地址映射表的更新和持久化的额外开销, 因此在写带宽方面相比传统 RAID 5 提升不明显, 甚至在有些情况会下降. 本文提出的 FT-RAID 充分利用了瓦记录磁盘中可覆盖写磁道的优势, 在各种典型 trace (包括 hm_0)下的表现都比 RAID 5 有显著提升.

7 结 论

瓦记录技术和 RAID 5 都会引起写放大现象, 这一现象会引起存储系统性能下降. 为了缓解瓦记录在构建磁盘阵列时的严重性能下降问题, 本文提出了一种基于瓦记录磁盘的高可靠数据存储方法 FT-RAID, FT-RAID 利用瓦记录盘中可覆盖写磁道可以原位更新的性质, 降低整体写放大率. FT-RAID 包含两个模块: FT-mapping 和 FT-buffer, 分别处理逻辑地址到物理地址的映射和实现瓦记录写缓冲区的管理. 实验结果表明, 与瓦记录传统 RAID 5 相比, FT-RAID 提高了 1.77–18.5 倍的总体性能, 平均能够降低 80.4%的写放大率. 由于该方案只需要拓展 ZBC 指令集, 无需修改现有硬件结构, 所以可以在各种企业环境下广泛部署. 可覆盖写磁道的结构和优化方法具有通用性和启发性, 对于其他新型结构的高密度磁盘(如 IMR), 虽然其内部包含了不同结构, 拥有不同性质, 但同样可能存在适合随机写的局部空间, 未来将在其他结构的磁记录上继续探索和研究.

References:

[1] Patrizio A. IDC: Expect 175 zettabytes of data worldwide by 2025. Network World, 2018.

[2] Riley D. Samsung's SSD Global Summit: Samsung: Flexing Its Dominance in The NAND Market. 2013.

[3] DRAMeXchange. NAND Flash Spot Price. 2014. http://dramexchange.com

[4] Thompson DA, Best JS. The future of magnetic data storage technology. IBM Journal of Research and Development, 2000, 44(3): 311–322.

[5] Wood R, Williams M, Kavcic A, *et al*. The feasibility of magnetic recording at 10 terabits per square inch on conventional media. IEEE Trans. on Magnetics, 2009, 45(2): 917–923

[6] Kryder MH, Gage EC, McDaniel TW, *et al*. Heat assisted magnetic recording. Proceedings of the IEEE, 2008, 96(11): 1810–1835.

[7] Albrecht TR, Arora H, Ayanoor-Vitikkate V, *et al*. Bit-patterned magnetic recording: Theory, media fabrication, and recording performance. IEEE Trans. on Magnetics, 2015, 51(5): 1–42.

[8] Westerndigital. Ultrastar dc hc600 smr series. https://www.westerndigital.com/products/datacenter-drives/ultrastar-dc-hc600-series-hdd

[9] Gibson G, Polte M. Directions for shingledwrite and two-dimensional magnetic recording system architectures: synergies with solid-stateDisks. Technical Report, CMU-PDL-09-104, CMU Parallel Data Laboratory, 2009.

[10] Amer A, Long DDE, Miller EL, *et al*. Design issues for a shingled write disk system. In: Proc. of the 26th IEEE Symp. Mass Storage Systems and Technologies (MSST), MSST 2010. Washington: IEEE Computer Society, 2010. 1–12.

[11] Gibson G, Ganger G. Principles of operation for shingled disk devices. Technical Report, CMU-PDL-11-107, CMU Parallel Data Laboratory, 2011.

[12] Feldman T, Gibson G. Shingled magnetic recording: Areal density increase requires new datamanagement. USENIX, 2013, 38(3).

[13] Wang GH, Du DHC, Wu FG, *et al*. Survey on high density magnetic recording technology. Journal of Computer Research and Development, 2018, 55(9): 2016–2028 (in Chinese with English abstract).

[14] T10 zoned block commands (zbc). 2014. http://www.t10.org/ftp/zbcr01.pdf

[15] Liu W, Feng D, Zeng L, *et al*. Understanding the SWD-based raid system. In: Proc. of the 2014 Int'l Conf. on Cloud Computing and Big Data. IEEE, 2014. 175–181.

[16] Aghayev A, Shafaei M, Desnoyers P. Skylight a window on shingled disk operation. ACM Trans. on Storage (TOS), 2015, 11(4): 16.

[17] Wu F, Yang MC, Fan Z, *et al*. Evaluating host aware SMR drives. In: Proc. of the 8th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage 16). 2016.

[18] He W, Du DH. Smart: An approach to shingled magnetic recording translation. In: Proc. of the 15th USENIX Conf. on File and Storage Technologies (FAST 17). 2017. 121–134.

[19] Novel address mappings for shingled write disks. In: Proc. of the 6th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage 14). 2014.

[20] Kadekodi S, Pimpale S, Gibson GA. Caveat-scriptor: write anywhere shingled disks. In: Proc. of the 7th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage 15). 2015.

[21] Yao T, Wan J, Huang P, *et al*. Geardb: A GC-free key-value store on HM-SMR drives with gear compaction. In: Proc. of the 17th USENIX Conf. on File and Storage Technologies (FAST19). 2019. 159–171.

[22] Ma C, Shen Z, Han L, *et al*. FC: Built-in flash-cache with fast cleaning for SMR Storage. In: Proc. of the 2019 IEEE Int'l Conf. on Embedded Software and Systems (ICESS). IEEE, 2019. 1–7.

[23] Liang YP, Chen TY, Chang YH, *et al*. Enabling sequential-write-constrained B+-tree index scheme to upgrade shingled magnetic recording storage performance. ACM Trans. on Embedded Computing Systems (TECS), 2019, 18(5s): 1–20.

[24] Li YZ, Sun ZZ, Ma ZM, *et al*. S-RAID 5: An energy-saving raid for sequential access based application. Chinese Journal of Computers, 2013, 36(6): 1290−1302 (in Chinese with Englisha abstract).

[25] Sun ZZ, Zhang QX, Tan YA, *et al*. Ripple-RAID: A high-performance and energy-efficient RAID for continuous data storage. Ruan Jian Xue Bao/Journal of Software, 2015, 26(7): 1824−1839 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4606.htm [doi: 10.13328/j.cnki.jos.004606]

[26] Jin C, Xi WY, Ching ZY, *et al*. Hismrfs: A high performance file system for shingled storage array. In: Proc. of the 30th Symp. on Mass Storage Systems and Technologies (MSST). IEEE, 2014. 1–6.

[27] LuocD, Yao T, Qu X, *et al*. DVS: Dynamic variable width striping raid for shingled write disks. In: Proc. of the 2016 IEEE Int'l Conf. on Networking, Architecture and Storage (NAS). IEEE, 2016. 1–10.

[28] Le QM, Amer A, Holliday J. SMR disks for mass storage systems. In: Proc. of the 23rd IEEE Int'l Symp. on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems. IEEE, 2015. 228–231.

[29] Zhang Q, Li SL, Zhang XY. A method of SWD Raid5 write sequentialization based on piggyback GC. Chinese High Technology Letters, 2018, 28(5); 383−391 (in Chinese with English abstract).

[30] Xiao W, Dong H, Ma L, *et al*. HS-BAS: A hybrid storage system based on band awareness of shingled write disk. In: Proc. of the 34th IEEE Int'l Conf. on Computer Design (ICCD). IEEE, 2016. 64−71.

附中文参考文献:

[13] 王国华, 杜宏章, 吴凤刚, 等. 高密度磁记录技术研究综述. 计算机研究与发展, 2018, 55(9): 2016−2028.

[24] 李元章, 孙志卓, 马忠梅, 等. S-RAID 5: 一种适用于顺序数据访问的节能磁盘阵列. 计算机学报, 2013, 36(6): 1290−1302.

[25] 孙志卓, 张全新, 谭毓安, 等. Ripple-RAID: 一种面向连续数据存储的高效能盘阵. 软件学报, 2015, 26(7): 1824−1839. http://www.jos.org.cn/1000-9825/4606.htm [doi: 10.13328/j.cnki.jos.004606]

[29] 张强, 李素玲, 张翔宇. 一种基于捎带回收的瓦记录 Raid 5 写顺序化方法. 高技术通讯, 2018, 28(5): 383−391.

吴坤尧(1997－), 男, 硕士生, 主要研究领域为新硬件, 分布式数据库.

张大方(1998－), 男, 硕士生, CCF 学生会员, 主要研究领域为计算机系统.

柴云鹏(1983－), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为分布式系统, 存储系统, 云计算.

王鑫(1981－), 男, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为数据库, 知识图谱, 大数据.