doi:10.3772/j.issn.1002-0470.2018.05.001

一种基于捎带回收的瓦记录 Raid5 写顺序化方法①

张 强②*** 李素玲*** 张翔宇***

(*中国科学院计算技术研究所 北京 100190)

(**中国科学院大学 北京 100049)

(***中华全国总工会 北京 100085)

摘 要 瓦记录技术(SMR)采用部分叠加相邻磁道的方式增大磁盘碟片的面密度,其磁道相互叠加,使得瓦记录磁盘无法进行原地更新,是因为直接更新某一磁道上的数据可能会覆盖掉相邻磁道上的有效数据,导致瓦记录磁盘的非顺序写需要引入额外的写放大开销。对于由瓦记录磁盘组成的 Raid5 来说,在非顺序写场景下,最终落盘的 IO 也是非顺序的,因此单盘性能下降,会极大地影响瓦记录 Raid5 的非顺序写性能。针对瓦记录 Raid5 非顺序写性能低的问题,提出了一种基于捎带回收的瓦记录 Raid5。该方法改进了传统 Raid5 的小写过程,在小写读的时候捎带读取附近的可被回收的数据或校验,通过将落盘写 IO 日志化,实现了落盘写 IO 的顺序化,避免了瓦记录磁盘的写放大开销。系统维护了地址映射表等元数据,在重构时只需对应用访问过的条带进行读写和计算,这减少了重构时需要读写的数据量。测试显示,在某些非顺序写场景下,写带宽超过了以 Bcache 作为缓存的 MD Raid5。从计算结果看,重构时的读写开销相比普通磁盘组建的 MD Raid5 也有明显减小。

关键词 瓦记录(SMR), 瓦记录 Raid5, 回收, 写顺序化

0 引 言

据权威市场调查机构 IDC 预测,到 2020 年,整个世界的数据总量将会增长 44 倍,达到 35.2ZB。数据量的爆炸式增长对存储系统的容量、性能等都提出了巨大的挑战。而作为主要存储介质的磁盘单碟片的面密度却很快将到达 1Tb/in^2[1] 的上限,这其中的主要限制因素就是超顺磁效应(super-paramagnetic effect)[2]。为了保证磁盘容量的增长速度,业内提出了多种新的存储技术:晶格介质(Bit patterned media)[3]、热辅助磁记录(heat assisted magnetic recording)[4]、微波辅助记录(microwave assisted magnetic recording)[5,6] 以及分离磁道介质(discrete track media)[7] 等。但是上述这些技术的实现都需要大幅改变现有磁盘的物理结构,实现成本高,短期内无法实现产品化。而一种称为瓦记录(shingled magnetic recording, SMR)[8-10] 的技术已经被业内公认为是下一代磁盘最可行的实现方式,其可在最小化对现有磁盘进行物理改进的前提下实现磁盘容量的扩增,可将现在磁盘的面密度提升 2 ~3 倍[11]。国际信息技术标准委员会(INCITS)的 T10[12] 和 T13[13] 研究组也正在研究制定瓦记录磁盘(shingled write disk, SWD)的相关接口标准,标准化 SWD 的磁盘接口。

SMR 技术通过部分叠加相邻磁道[14,15] 的方式来增加磁盘面密度。虽然 SMR 技术可大幅提升磁盘存储容量,但磁盘的磁道重叠引起了磁道间的干

① 中国科学院战略性先导科技专项(XDA06010401)和中国科学院重点部署(KG8D-EW-103-517)资助项目。

② 男,1982 年生,博士生;研究方向:瓦记录,磁盘冗余阵列,分布式系统;联系人,E-mail: zhangqiang@ict.ac.cn

(收稿日期:2018-02-23)

涉,即在一条磁道上写数据时会覆盖后续相邻的若干条磁道的数据,因此瓦记录磁盘(SWD)不能像传统磁盘(hard disk drive,HDD)一样在任意位置更新数据,即无法原地更新[16-18]。要想保证数据写入的正确性,需避免 SMR 的原地更新,会引入额外的读写开销。一种简单的方案是引入读修改写技术(read-modify-write,RMW)[19],即在需要写磁盘上某一物理位置时,先将可能被覆盖的相邻连续磁道上的数据读出,修改后将新数据和读出的数据一起写回到磁盘中。但是在写回时又会产生新的数据覆盖问题,导致重复执行 RMW 过程,直到磁盘的末尾。因此对磁盘内任意数据的修改会导致平均 1/2 磁盘数据的读写,极大地影响了 SWD 的非顺序写性能。为减小瓦记录磁盘的写放大程度,业内一般认为应该将 SWD 按带(band)连续划分,每个带是一段连续的物理空间,带之间预留一段不可访问空间,确保更新一条带上的数据不会覆盖相邻带上的数据,即将原地更新的写放大程度控制在一个带内。

对于 SWD 的非顺序写,业内的许多学者提出了各种可能的解决方案,主要分为两类:in-place update 和 out-place update[19]。

In-place update,即数据写入前需要先将可能会被覆盖的数据读出,合并后再写回,即读修改写(RMW)。Luo[20]等提出的垂直编址方案即属于此类,通过将磁盘编址方式由横向改为纵向,并使用高速持久缓存,从而减小了 RMW 带来的开销,提升了瓦记录磁盘的非顺序写性能。因为改变磁盘的编址方式会改变现有磁盘的基础架构,因此存在难以产品化的问题。

Out-place update,即采用写顺序化技术来避免 SMR 的写覆盖问题。写顺序化技术将非顺序写转换成了顺序写,因此能提升非顺序写性能。但是写顺序化一方面会带来元数据管理的开销,即需要维护数据块逻辑地址与物理地址之间的映射信息;另一方面需要引入回收(garbage collection,GC)的开销。对于更新操作,写顺序化技术会使旧的数据无效掉,但是因为 SMR 不能直接原地写入数据的特性,无效掉的物理空间是不能直接使用的,因此需要将零散的无效空间整理成连续的物理空间,即回收。而回收操作必然会引入额外的读写开销。Cassuto[21]等提出的 Indirection Systems 及 Lin[22]等提出的 H-SWD 都属于此类。

在瓦记录 Raid 方面 Liu[23]等人实现了一个瓦记录 Raid 模拟器,从测试结果来看,在非顺序写场景下瓦记录 Raid5 的性能要远低于标准 Raid5。本研究对希捷 8TBSWD[24](由持久磁盘缓存区和瓦记录区组成,写请求优先顺序写入持久缓存区,当持久缓存区写满或到达水位时通过 RMW 将持久缓存区中的数据移动到瓦记录区中)组成的标准 Raid5 进行了 4kB 和 64kB 随机写测试,写请求先写入持久缓存,由于写顺序化的原因,性能有大幅提升,但是当希捷 SWD 的持久缓存被写满启动回收后,写性能急剧下降。

由于影响瓦记录 Raid5 非顺序写性能的主要原因是瓦记录本身的写放大开销,本文为了避免瓦记录磁盘非顺序写带来的写放大开销,我们提出了一种基于捎带回收的瓦记录 Raid5(piggyback GCRaid5,PRaid5),PRaid5 使用循环 Log 来管理瓦记录磁盘中的带,实现瓦记录磁盘的写顺序化,同时改造了传统 Raid5 的小写过程,在小写读的时候捎带读取附近可回收的数据或校验,回收读出的数据与应用和要写入的数据一起顺序写入带中,在小写的过程中顺带实现了回收和写顺序化,从而有效提升瓦记录 Raid5 的非顺序写性能。由于系统记录了地址映射表等信息,在重构时可以避免对所有条带做读写和计算,减少了重构时的读操作,从而也提升了重构性能。

1 PRaid5 介绍

1.1 PRaid5 布局

PRaid5 保持了传统 Raid5 的布局方式,只是添加了一个高速持久缓存层(如 SSD),图 1 所示是由 5 块瓦记录磁盘组成的 PRaid5。本研究将不同盘上具有相同带号的带组成一个组(group)。由于应用具有一定局部性,使用高速持久缓存可以持久化写请求,并减少写阵列的频率,从而提升系统性能。

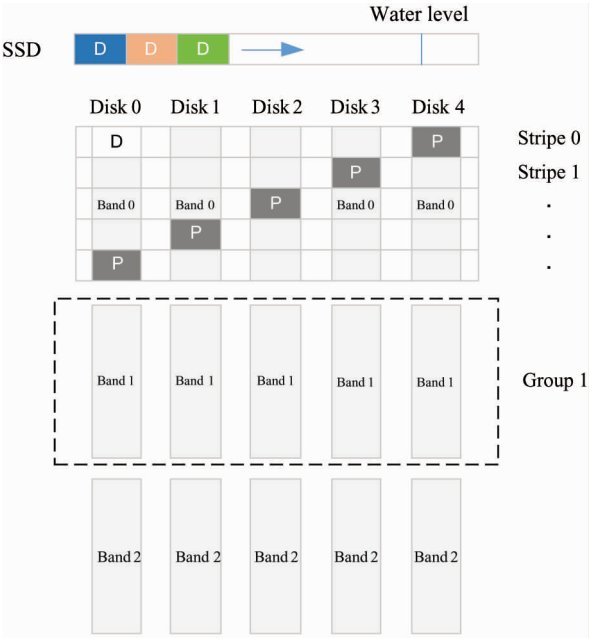

图 1　PRaid5 布局

本文采用循环 Log 管理带,保证落盘的 IO 是带内顺序的,如图 2 所示,循环 Log 是将带看成一个循环队列,数据只能从一头写入。头指针永远指向下一个数据块要写入的物理位置,尾指针指向当前 Log 的末尾有效数据块的物理位置,当带中有数据写入后头指针向前移,当尾指针处的数据块无效时,可将尾指针前移到有效数据块的位置。头尾指针之间需要保持一个最小距离(min _ dist),避免瓦记录带内产生写覆盖。当有数据更新时,旧的数据被无效掉,图 2 中阴影部分是数据更新产生的无效空间,即洞(hole)。由于瓦记录相邻磁道部分重叠的特性,使得产生的洞不能直接写入数据,因此当数据持续写入时,头尾指针之间的距离最终会到达 min _ dist,导致带内可用物理空间小于实际物理空间,因此需要回收,从而将写顺序化产生的洞整理成为可写入的连续物理空间。

采用循环 Log 管理磁盘中的带,实现了带内的写顺序化。本文以 group 为单位进行下刷,即缓存到达一定水位后选取缓存中属于同一个组的,访问热度最低的数据块进行下刷。数据通过缓存聚集后,下刷时会形成带内大块的写 IO,从而一定程度上利用磁盘顺序写性能优势,提升写盘性能。

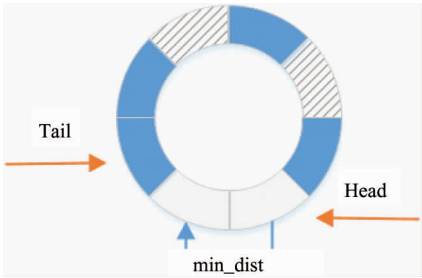

图 2　带内循环 Log

1.2　写顺序化

由于带内以循环 Log 方式写入数据,PRaid5 可以将随机写转换为带内的顺序写。以 5 块瓦记录磁盘组成的 PRaid5 为例,本文用 C_{xy} 表示 Stripe x 中的 Chunk y,P_x 表示 Stripe x 的校验 Chunk。其中,x 表示 Stripe 在 Group 内的编号,y 表示磁盘编号。

图 3 对标准 Raid5 和 PRaid5 数据的写盘位置

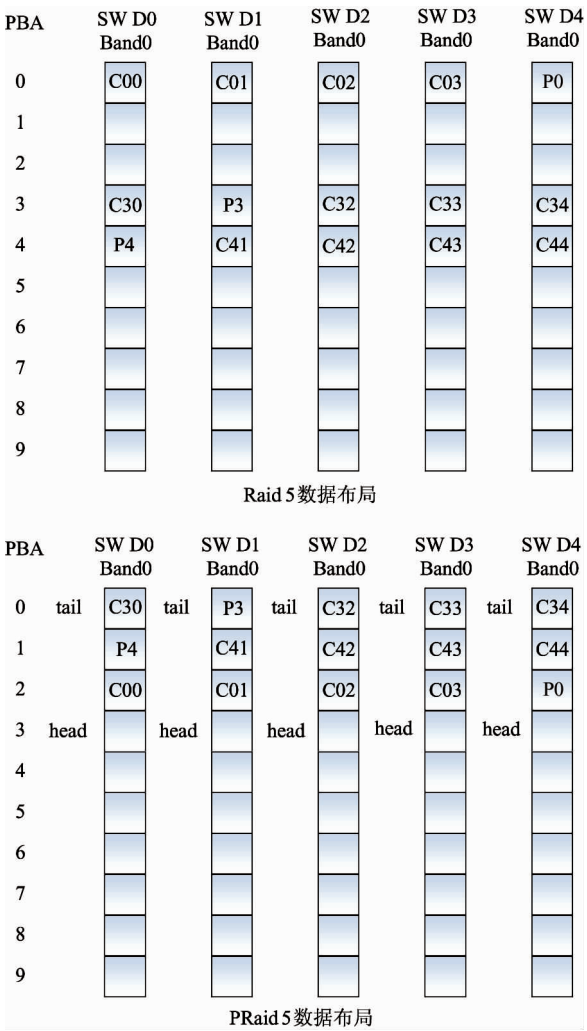

图 3　带内写顺序化

进行了对比,其中 PBA 是带内物理 Chunk 的编号。假设 stripe3, stripe4 和 stripe0 的数据和校验按时间顺序依次写入阵列中。图 3 上方是标准 Raid5 中数据和校验应当写入的位置。图 3 下方是 PRaid5 经过带内写顺序化后数据的写入位置。由于 group0 中每个带内都是追加写,因此写入过程中可以有效避免瓦记录的写放大问题。

1.3 捎带回收

虽然 PRaid5 采用写顺序化可以避免瓦记录的写放大问题,但是数据更新会导致 group 内实际可使用物理空间小于对外提供的存储空间,图 4 所示为 stripe4 和 strip0 更新后 group0 的磁盘快照。

从图 4 中可以看到,stripe4 和 stripe0 的数据更新后之前写入的 stripe4 和 stripe0 的数据和校验所在的位置变为无效,用阴影来表示无效的数据空间,并记为“洞”。由于瓦记录写覆盖的原因,“洞”是不能直接写入数据的,因此需要将 group 内的离散的数据和校验 整理成物理连续的,从而将“洞”占用的无效物理空间变为可写入的有效空间。

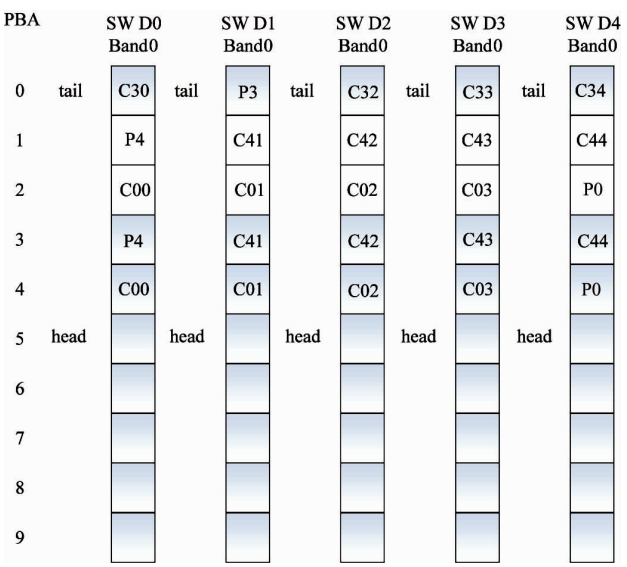

图 4 stripe4 和 stripe0 数据更新

由于以带为单位读取带内所有离散数据再写回带中会导致系统长时间无响应,因此采用捎带回收的方式来尽量减少系统阻塞的时间。捎带回收是利用 Raid5 小写读开销来顺便读取可以被回收的数据和校验块,将整带的回收读转换为数据正常写入过程中的小粒度回收读,从而将大的回收读开销分散到每次小写更新中,通过适当增大小写更新读开销来均摊整带回收开销,从而达到降低整体回收开销的目的。

还是以 stripe4 和 stripe0 的更新为例。在 stripe4 和 stripe0 小写读的时候顺便也读取 group0 中每个带的尾指针指向的数据和校验 Chunk,即读取 stripe3 的数据和校验。在写入 stripe4 和 stripe0 的时候也将 stripe3 中所有的数据和校验写回到带中,引入捎带回收后的磁盘快照如图 5 所示。

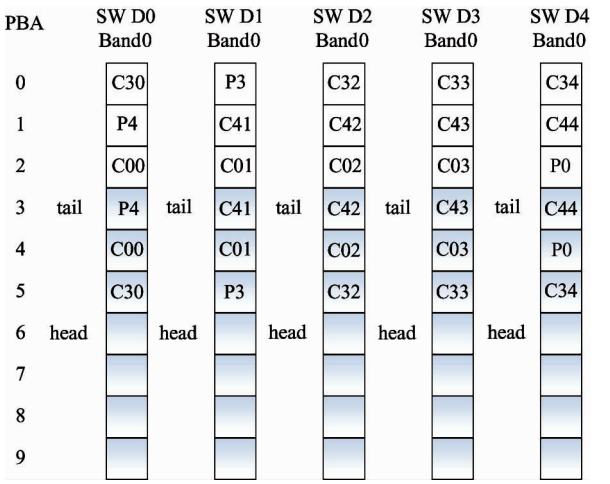

图 5 捎带回收 stripe3 后的磁盘快照

从图 5 中可以看出,通过捎带回收 stripe3,在小写更新的基础上引入了一次读写 stripe3 数据和校验的开销,但是却在 stripe4 和 stripe0 的写入过程中顺便将旧的 stripe4 和 stripe0 占用的空间释放了出来,即 group0 中每个带内的 tail 都向前移动了 2 个 Chunk 的距离,此时每个带内的实际可用物理空间与能提供的存储空间一致,在数据写入的过程中顺便实现了回收。

1.4 元数据管理

由于 PRaid5 实现了写顺序化,因此需要维护数据块的物理地址(PBA)和逻辑地址(LBA)之间的映射关系,从而在捎带读时可确定读出的物理块的逻辑块号,并使应用读取数据时能根据映射表找到逻辑块所在的物理位置。本研究将图 5 中的数据和校验编号用条带号来替换得到 group0 的元数据表,如图 6 所示。假设应用需要读取 C42 中的数据,根据元数据表可知 C42 在 2 号 SWD,PBA 为 3。捎带回

收时,通过元数据表就可以确定需要读取的是哪个 stripe 中的数据和校验。如图 6 所示,每个带中的 tail 所指向的数据和校验属于 stripe4。

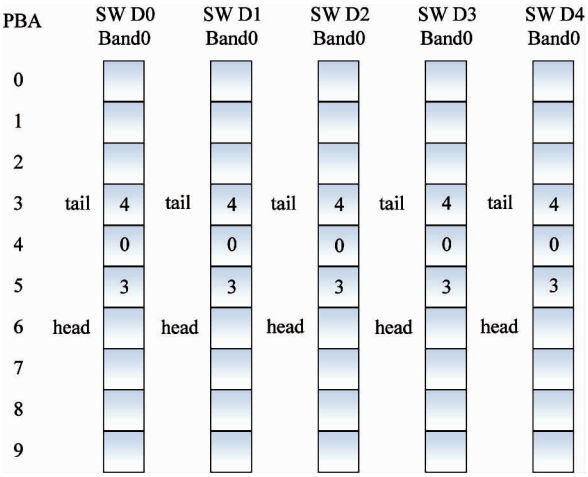

图 6　元数据表

对于元数据管理来说,需要考虑元数据量的大小和元数据的存储访问模式。

以单盘 8TB,5 块瓦记录磁盘组成的 PRaid5 为例,假设数据块的管理粒度为 4kB,以 4 个字节表示 PBA 和 LBA 的地址,则总的最大元数据量为 80GB,只占系统总容量的 0.2%。为保证系统元数据的可靠性,将元数据以副本的方式存放在 SWD 的随机区(T10 标准委员会将瓦记录磁盘划分为随机区和瓦记录区,随机区可以像 HDD 一样实现原地更新,不会产生写放大)中,可以实现元数据的原地更新,无需触发瓦记录磁盘的读改写操作。

如图7所示,MD0 ~ MD4是Disk0 ~ Disk4上的

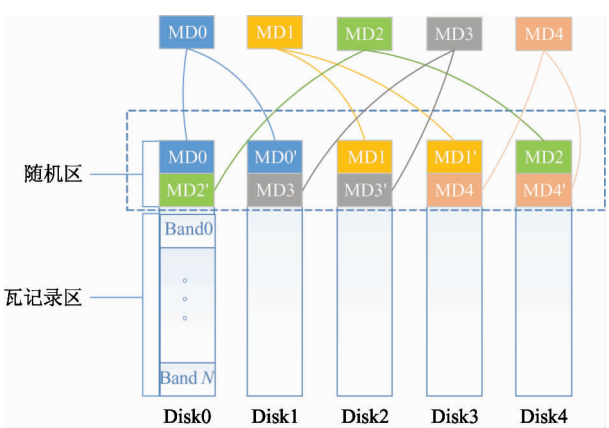

图 7　元数据布局

元数据,MD0′ ~ MD4′分别是 MD0 ~ MD4 的副本,元数据与其副本存放在不同磁盘的随机区中,一定程度上保证了元数据的可靠性。

1.5　重构

当发生磁盘损坏替换后,可由元数据信息确定恢复时需要读写的数据,不需要做全盘恢复。下面以 SWD2 的重构为例,如图 8 所示。

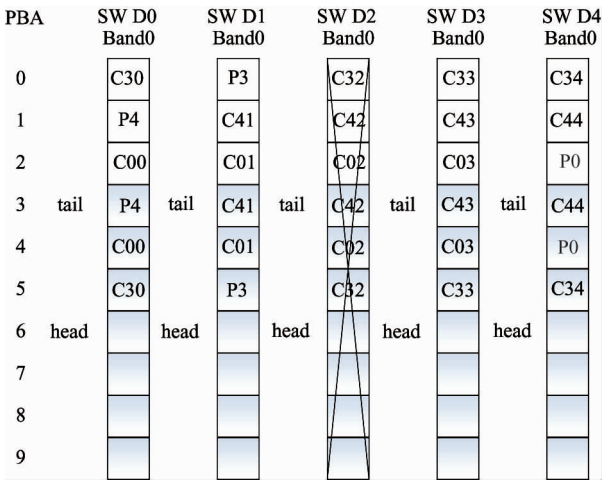

图 8　重构 SWD2

当 SWD2 需要被恢复时,通过地址映射表,就可以确定 stripe4 中 C41、C43、C44 和 P4 的位置,读出可计算并恢复 C42。同理 C02 和 C32 也可以被恢复。

2　测试评价

本文在 3.18 内核版本上设计并实现了 PRaid5 原型系统,系统实现为内核驱动模块,并采用希捷 4TB HDD 来模拟瓦记录磁盘。对比对象选取了 Bcache + 标准 Raid5(BRaid5)。Bcache 是 Linux 内核中的一个开源缓存模块,由于其条带感知的特性,即下刷时会优先下刷全条带,尽可能避免小写更新开销,因此对 Raid5 非顺序写有明显的性能改善。环境配置如表 1 所示。由于应用有一定局部性,因此假设标准Raid5在重构时只重构应用访问过的组,而不是全盘重构,使其与 PRaid5 的对比更公平一些。

表 1 环境配置

硬件环境：		
服务器	Dell R730	
CUP 类型	Intel(R)Xeon(R)CPU E5-2630 v3@2.40GHz	
CPU 数量	16	
内存容量	16GB	
HDD	4TB\SATA\7200 RPM\64MBcache\ write-caching = 0(off)	
带大小	256MB	
软件环境：		
操作系统	CentOS7.0	
内核版本	3.18.30	
Raid5 配置	磁盘数量	5
	Raid5 Stripe Cache	32
	Raid5 ChunkSize	512KB
	Queue Depth	磁盘数量 ×32

选取了 6 种真实场景下的应用作为测试用例，如表 2 所示，选择的用例来自于微软剑桥研究院的企业服务器。表 2 中，Trace 为应用名称，Tot. Reqs 表示应用总的请求数，W. percent 表示写比例，Tot. Size 表示总的 IO 数据量，W. Size 表示总的写入数据量，W. Update 表示写更新比例，Avg. Len 表示平均请求长度。

将缓存大小设置为应用 Footprint（Footprint 表示应用访问过的物理空间容量，重复访问只统计一次）的 4%，确保缓存与数据空间比例为一个合适的值。缓存设置如表 3 所示。

2.1 测试评价

本研究对比了 6 种场景下 Bcache + 标准 Raid5（BRaid5）和 PRaid5 的写带宽，如图 9 所示。从测试结果看，对于 mds0、rsrch0 和 usr0，PRaid5 写带宽相对于 BRaid5 分别提升了14.1%、24.7% 和29.4%。对于 stg0、ts0 和 hm0，PRaid5 的写带宽比 BRaid5 分别低15.3%、1.7% 和38.1%。

图 10 显示了 BRaid5 和 PRaid5 读开销。从图中可以看出，PRaid5 的读盘开销普遍高于 BRaid5。说明捎带回收引入了大量的读开销。特别对于 hm0 来说，PRaid5 的读开销是 BRaid5 的 1.5 倍。

由于捎带回收在写入要更新的数据和校验的同时还会写入回收读取的数据和校验。因此 PRaid5 的写开销也会大于 BRaid5。从图 11 中可以明显看出这一点。

表 2 应用特征

Trace	Tot. Reqs	W. percent	Tot. Size	W. Size	W. Update	Avg. Len
mds0	1211034	88.1%	10.6GB	7.4GB	95.5%	9417B
rsrch0	1433655	90.6%	12.2GB	10.8GB	97.3%	9144B
stg0	2030915	84.8%	22.4GB	15.1GB	97.4%	11853B
ts0	1801734	82.4%	15.5GB	11.3GB	95.3%	9220B
usr0	2237889	59.6%	48.4GB	13.1GB	95.1%	23212B
hm0	3993316	64.4%	30.4GB	20.5GB	92.0%	8185B

表 3 缓存设置

Trace	Footprint	Cache size	Percentage
mds0	0.3GB	12MB	4%
rsrch0	0.3GB	12MB	4%
stg0	0.4GB	16MB	4%
ts0	0.54GB	22MB	4%
usr0	0.65GB	26MB	4%
hm0	1.6GB	64MB	4%

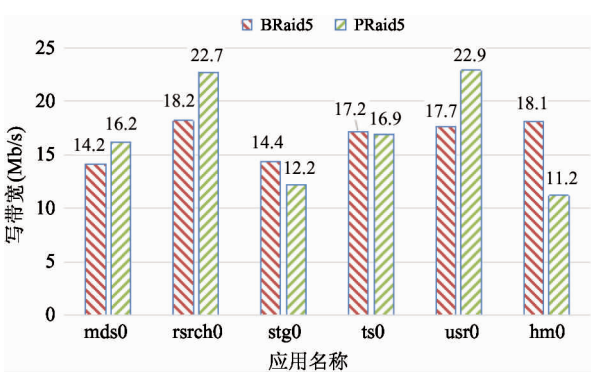

图 9 BRaid5 和 PRaid5 写带宽对比

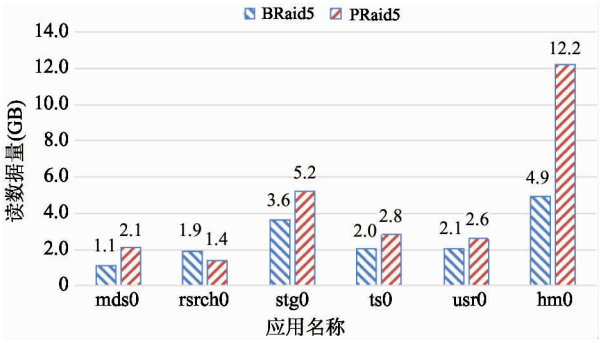

图 10　读开销对比

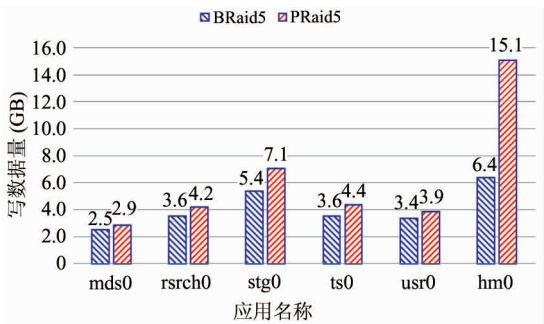

图 11　写开销对比

表 4 统计了触发回收最多的组的访问条带占比,即组内下刷访问过的条带数量(重复访问只统计一次)与组内总条带数量的比值。从统计数据可以看出,除 hm0 以外,其余应用的访问条带占比较低。访问条带占比低意味着下刷时更新写比例高,由于写顺序化的原因,更新写会产生洞,频繁更新写,使得组内每个带上的洞较多,洞的连续性就好,回收只需要读取较少的数据即可释放出连续物理空间,因此回收开销较小。而 hm0 的访问条带占比高达 91%,因此产生的洞较少,由于缓存聚集导致下刷时在带内为大块连续写 IO,当带中没有空间写入时需要回收较多的数据才能释放出连续物理空间容纳要下刷写入的数据,因此回收开销占比高,极大地影响了写性能。

表 4　参数统计

应用名称	mds0	rsrch0	stg0	ts0	usr0	hm0
访问条带占比	16%	8%	9%	9%	13%	91%

虽然 PRaid5 的读写开销普遍高于 BRaid5,但是由于写顺序化带来的磁盘磁头抖动程度的降低,使得 PRaid5 在 mds0、rsrch0 和 usr0 场景下的写带宽高于 BRaid5。而对于 stg0 和 ts0 来说,PRaid5 的写带宽也只比 BRaid5 有小幅降低,但是对于 hm0 来说由于应用的局部性较差,导致回收开销过高,因此 PRaid5 写带宽大幅低于 BRaid5。

图 12 是假设第 0 号盘损坏重构时统计的 PRaid5 总的读数据量与标准 Raid5 读数据量的比值。从统计结果可以看出,在 6 种应用场景下,PRaid5 重构时需要读取的数据量均小于标准 Raid5,其原因是 PRaid5 记录了数据块的地址映射表,在重构时无需重构组内的所有数据,只需重构访问过的条带,因此读取的数据量有所降低。

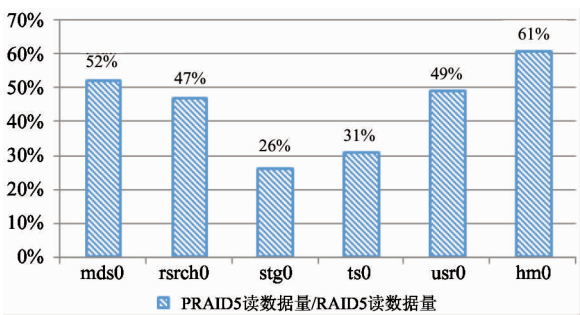

图 12　重构时 PRaid5 与 MD Raid5 读数据量比值

3　结　论

本文提出了一种基于捎带回收的瓦记录 Raid5 系统,通过小写读时进行捎带回收读和以循环 Log 来实现带内写顺序化一定程度上提升了瓦记录 Raid5 的非顺序写性能。测试显示对于局部性较好的应用,PRaid5 的回收开销较小、性能较好。由于系统记录了映射信息,使得重构时无需读取组内全部条带上的数据,减小了重构时的读数据量。捎带回收不仅适用于瓦记录 Raid5,对于传统磁盘组成的 Raid5 依然适用,并且很容易扩展到 Raid6 等更高可靠性的 Raid 系统中。

参考文献

[1] Wood R, Williams M, Kavcic A, et al. The feasibility of magnetic recording at 10 terabits per square inch on conventional media[J]. *IEEE Transactions on Magnetics*, 2009, 45: 917-923

[2] Richter H, Dobin A, Heinonen O, et al. Recording on bit-patterned media at densities of 1Tb/in 2 and beyound [J]. *IEEE Transactions on Magnetics*, 2006, 42(10): 2255-2260

[3] White R L, New R M H, Pease R F W. Patterned media: A viable route to 50 Gbit/in 2 and Up for magnetic recording? [J]. *IEEE Transactions on Magnetics*, 1997, 33(1): 990-995

[4] Kryder M H, Gage E C, McDaniel T W, et al. Heat assisted magnetic recording[J]. *Proceedings of the IEEE*, 2008,96(11):1810-1835

[5] Igarashi M, Suzuki Y, Miyamoto H, et al. Effective write field for microwave assisted magnetic recording[J]. *IEEE Transactions on Magnetics*, 2010,46(6):2507-2509

[6] Zhu J G, Zhu X C, Tang Y H. Microwave assisted magnetic recording[J]. *IEEE Transactions on Magnetics*, 2008,44(1):125-131

[7] Sohn J S, Lee D, Cho E, et al. Fabrication and MFM study of 60 nm track-pitch discrete track media[J]. *Nanotechnology*, 2011,22(3):353

[8] Amer A, Holliday J, Long D D E, et al. Data management and layout for shingled magnetic recording[J]. *IEEE Transactions on Magnetics*, 2011,47(10):3691-3697

[9] 肖文健. Hs-SWD:基于瓦记录磁盘的混合存储系统:[硕士学位论文][D]. 北京:中国科学院大学, 2016

[10] 罗旦. 瓦记录磁盘系统数据组织关键技术研究:[博士学位论文][D]. 武汉:华中科技大学计算机学院, 2016

[11] Greaves S, Kanai Y, Muraoka H. Shingled Recording for 2-3 Tbit/in2 [J]. *IEEE Transactions on Magnetics*, 2009, 45(10): 3823-3829

[12] Information technology-Zoned Block Commands (ZBC). Draft Standard T10/BSR INCITS 536[S]. American National Standards Institute, Inc., September 2014. http://www.t10.org/ftp/zbcr01.pdf

[13] Information technology-Zoned Device ATA Command Set (ZAC). Draft Standard T13/BSR INCITS 537[S]. American National Standards Institute, Inc., December 2015. http://www.t13.org/Documents/UploadedDocuments/docs2015/di537r05 - Zoned_Device_ATA_Command_Set_ZAC.pdf

[14] Amer A, Holliday J A, Long D D E, et al. Data management and layout for shingled magnetic recording[J]. *IEEE Transactions on Magnetics*, 2011, 47(10): 3691-3697

[15] Gibson G, Polte M. Directions for shingled-write and two-dimensional magnetic recording system architectures: synergies with solid-state disks, CMU-PDL-09-104[R]. Pittsburgh PA: Parallel Data Lab, Carnegie Mellon University, 2009

[16] Gibson G, Ganger G. Principles of Operation for Shingled Disk Devices. CMU-PDL-11-107[R]. Pittsburgh, PA: Parallel Data Lab, Carnegie Mellon University, 2011

[17] Amer A, Long D D E, Miller E L, et al. Design issues for a shingled write disk system[C]. In: Proceedings of the IEEE 26th Symposium on Mass Storage Systems and Technologies(MSST), Incline Village, USA, 2010. 1-12

[18] Feldman T, Gibson G. Shingled magnetic recording areal density increase requires new data management[EB/OL]. https://www.usenix.org/system/files/login/articles/05_feldman_022-030_final.pdf

[19] Pitchumani R, Hospodor A, Amer A, et al. Emulating a shingled write disk[C]. In: Proceedings of IEEE 20th International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), Washington, DC, USA, 2012. 339-346

[20] Luo D, Wan J G, Zhu Y F, et al. Design and implementation of a hybrid shingled write disk system[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2016, 27(4):1017-1029

[21] Cassuto Y, M. Sanvido A A, Guyot C, et al. Indirection systems for shingled-recording disk drives[C]. In: Proceedings of the IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), Lake Tahoe, USA, 2010. 1-14

[22] Lin C-I, Park D, He W, et al. H-swd: Incorporating hot data identification into shingled write disks[C]. In: Proceedings of IEEE 20th International Symposium on Modeling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS), Washington, DC, USA, 2012. 321-330

[23] Liu W, Feng D, Zeng L, et al. Understanding the SWD-based raid system[C]. In: Proceedings of International Conference on Cloud Computing and Big Data, Wuhan, China, 2014. 175-181

[24] Aghayev A, Shafaei M, Desnoyers P. Skylight-a window on shingled disk operation[J]. *ACM Transactions on Storage*, 2015, 11(4): 16

A method of SWD Raid5 write sequentialization based on piggyback GC

Zhang Qiang[*,**], Li Suling[***], Zhang Xiangyu[*]

([*] Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

([**] University of Chinese Academy of Sciences, Beijing 100049)

([***] All-China Federation of Trade Unions, Beijing 100085)

Abstract

Shingled magnetic recording (SMR) increases surface density by partly overlapping adjacent tracks, which can't support in-place update, because updating data on one track directly will destroy the data on successive tracks. So it needs to bring in additional overhead to avoid write overlapping. For SWD Raid5, in non-sequential write scene, final IOs sent to disk are also non-sequential, which affects the Raid5 performance greatly. A piggyback garbage collection Raid5 (PRaid5) is proposed to improve write performance of SWD Raid5 by modifying Raid5 RMW and write sequentialization. Since address mapping table is recorded, it needs not to read whole disk as recoverying. Test results show that in some non-sequential writing scenarios, PRaid5 has higher performance and lower read/write overhead than Linux MD Raid5.

Key words: shingled magnetic recording (SMR), SWD Raid5, garbage collection, write sequentialization