海量存储课程论文

虚拟守卫：基于磁道的叠瓦式转换层

院（系）名 称：计算机学院

专 业 名 称 ：计算机技术

学 生 姓 名 ：李翔

学 生 学 号 ：2017282110241

二〇一七年十二月

摘　要

虚拟守卫是基于磁道考虑为叠瓦式磁盘驱动所设计的静态映射转换层。在写入时通过将叠瓦方向上下个磁道的数据缓存到持久性缓存，实现对叠瓦式磁盘目标磁道上特定扇区的直接覆盖写入。这让虚拟守卫能够从磁道级别进行定位，从而减少了很多工作负载时所需要进行的清除操作。我们将虚拟守卫的性能与在以前研究中分析和建模过的驱动器管理型叠瓦式磁盘驱动器进行了比较。虚拟守卫将现实操作中 99.9％的延迟降低了 15 倍，对于叠瓦式磁盘固有弱项随机写入则将延迟最大降低了 32％。

关键词：虚拟守卫；叠瓦式磁盘；直接覆写

1. 介绍及相关工作

叠瓦式磁记录技术（SMR）在无需对磁盘设计和制造方面进行根本的改变的前提下为磁盘存储量的显着提高提供了新的方式。与传统的磁盘不同，叠瓦式磁记录(SMR)磁盘通过使磁盘之间像瓦片一样堆叠使磁道密度明显增加，从而在无需对磁道本身进行改变的同时大大提升了磁盘的存储容量。但叠瓦式磁记录技术在显著提高磁道密度的同时也产生了新的问题，其写入磁头会覆盖多个磁道。例如当在磁道 T 上进行写入时，被写入磁头覆盖的另一个磁道 T1 上的数据将会丢失。这种限制可以在主机上使用新的 SCSI 和 SATA 扩展来解决，但是现在市场上在售的许多 SMR 磁盘是驱动器管理式的，在不对现有文件操作进行改变的前提下通过磁盘内部的叠瓦转换层将传统磁盘操作转化为叠瓦式磁盘操作。

目前已经有了许多叠瓦式转换层机制[6,9,5,2,8]。 然而，通过传输驱动的测量，这些方法大多可以概括为特定区域叠瓦式转换层，在这些方法中通常将写入数据传入一个特定区域例如持久缓存，然后再数据组织后连续写入到叠瓦区域。在一个常见的改进[1]中，磁盘被划分成由未使用的磁道或保护磁道分开的磁带，从而允许磁带通过清除操作进行完全覆盖写入，并且将逻辑地址静态地映射到这些磁带中的”原始位置”。

迄今为止提出的叠瓦式转换层使用逻辑地址映射，将磁盘几何结构等细节留给磁盘固件等较低层处理。但有一个例外，与通常的叠瓦式转换层相反，H 和 Du[3，4]提出了两种叠瓦式转换层中的磁道映射机制，在将逻辑地址映射到磁道/头部/扇区位置之后再执行翻译。他们的第一步提出了一些磁道映射方式，在高编号磁道尚未写入的情况下，对低编号轨道进行高效率写入。最近提出的 SMaRT [4]更为直接，将磁道进行动态映射。SMaRT 对磁盘利用率高度敏感，由于对磁道进行动态映射而产生了一个庞大的映射表（64MB）需要 100 毫秒以上才能载入到磁盘中因而只能保存在内存中，使得利用率在 90%或以上时性能表现很差。最后，SMaRT 不处理由于磁道长度变化，自适应格式化[7]或滑动保留[11]而导致的轨道大小差异。

我们提出了虚拟守卫，一种新颖的磁道映射叠瓦式转换层。 在修改磁道 T 之前，虚拟守卫将 T+1 磁道中数据转移到缓存，允许对磁道 T 的多重就地修改操作。它将每个轨道映射到一个静态的磁带位置，消除了大多数轨道大小问题。持

久性缓存由数据量较大的外径磁道组成，允许大多数逻辑缓存磁道占用一整个物理磁道以及小的元数据头。磁盘的开销很小，由一个小的（<0.5%）持续性高速缓存和每个磁带中的单个保护磁道组成，RAM 开销几乎可以忽略不计。

我们介绍了虚拟守卫的设计以及仿真结果，展示了相对于传统的特定区域叠瓦式转换层的显著性能替身，同时拥有相同或更低的磁盘空间和内存开销。虚拟守卫看来利用了强大的空间局部性，在这些工作负载中除了合成痕迹之外，在所有情况下都避免了清除操作。

2. 虚拟守卫

虚拟守卫是基于磁道映射的叠瓦式驱动转换层，将逻辑地址映射到磁道号再转换为物理地址。选取盘面的外径磁道区域作为持久性缓存；盘面的主要区域划分为由保护磁道（空磁道）分割开来的磁带（bands），磁盘的逻辑地址将映射到这些叠瓦式区域，持久性缓存不对应逻辑地址，只起缓存作用。与市面上大多数叠瓦式转换层机制（特定区域叠瓦式转换层）将数据直接写入持久性缓存再组织后连续写入叠瓦式磁记录区域不同，虚拟守卫在写入将叠瓦方向上下一层，即被写入磁头干扰的下一磁道的数据复制到持久性缓存中。像一个不存在的守卫保护着即将被污染的叠瓦覆盖磁道。允许直接对当前磁盘进行原地写入等操作而不损坏其他磁道数据。

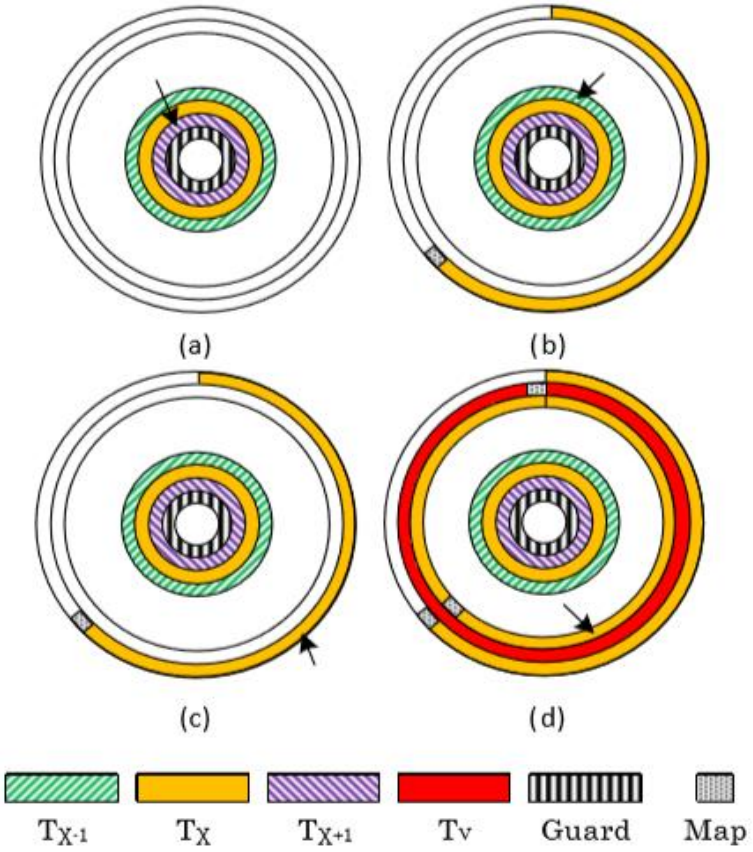
2.1 原地写入

当主机对逻辑地址 X 发出写指令时，虚拟守卫根据逻辑地址 X 计算出改地址对应的磁带号B_x和磁道号T_x，再检查B_x磁带上磁道T_x是否是最后一个磁道（如过是就不存写覆盖问题），如果不是再检测T_x的下一个磁道T_{x+1}是否已被缓存，如果未被缓存则将其数据复制到持久性缓存中。从而避免由于叠瓦式磁盘的写放大问题导致的临近磁盘数据损坏。

如果B_x磁带上磁道T_x既不是最后一个磁盘，磁道T_x在叠瓦方向上下一个磁道T_{x+1}也未被缓存，那么虚拟守卫将会读取磁道T_{x+1}上数据写入持久性缓存区域上连续的下一个空闲磁道T_{wf}，这样磁道T_x便可以进行原地写入操作而不用担心损坏磁道T_{x+1}上的数据。假如持久性缓存区域也是有叠瓦式磁道组成的话，虚拟守卫对磁道T_{x+1}进行同样操作将数据写入持久性缓存上的下一个空闲磁道，在叠瓦

式持久性缓存上进行连续写入操作，然后再对磁道T_x进行原地写入操作，显然虚拟守卫对持久性缓存区域具有很强的适应性，不像传统的叠瓦式转换层要求持久性缓存区域为传统磁道结构。

图一： 虚拟守卫在不同写入场景下的操作



(a)在磁道T_{x+1}上进行原地写入操作，T_{x+1}是当前磁带B_x上最后一个磁道，后面是守卫磁道（空磁道），不存在写放大问题。（b）在将磁道T_x缓存到持久性缓存区域后对磁道T_{x-1}进行原地写入操作。(c)在持久性缓存上紧随着日志头就地写入磁道T_x数据。(d)在磁道T_x之前的位置有空闲磁道并且磁道T_x的数据在持久性缓存区域的写出前沿时将其写回原来的位置

2.2 缓存映射

虚拟守卫保存一张异常表，保存那些磁道被缓存到持久性缓存中，以及这些

磁道的数据在持久性缓存中对应的位置。由于持久性缓存区域相对较小，基于磁道进行映射，虚拟守卫保存的异常表与 SMaRT [4]中动态映射产生的 60MB 表和商业领域中常见的特定区域叠瓦式转换层的大约 1MB 的表而言相当的小。特别地，如果N_{MTE}是映射表中的条目的数量，并且N_{tr}和N_{rtc}分别是以磁道为单位的驱动器和永久缓存的大小，则映射表大小将大致为:

$$MapSize = N_{MTE} \times \left(log(N_{tr}) + log(N_{trc})\right)$$

对于 5TB 的叠瓦式磁盘,据文章 Skylight[1],该磁盘约有 4000000 个磁道和 24GB 的持久性缓存，那么虚拟守卫的映射表将大致为 30KB。由于这个映射表足够小，允许以较小的开销将一个完整的副本附加到持久性缓存中的每个磁道上。

2.3 清除操作

清理过程将磁道从持久性缓存移回其原始位置。在出现下列情况时虚拟守卫将不会使用后台清理操作，而是直接触发清理。

1. 映射表中空闲磁道的数量降到阈值 a 以下，或者
2. 持久性缓存中最旧的轨道与写入前沿（包括任何无效的轨道）之间的距离超过第二阈值 b。

在一个单独的清理过程中，虚拟守卫将清理两个磁带，每个磁带都会重复以下过程：选择日志尾部中记录的前两个磁道，然后清除缓存中的所有在同一磁带的磁道。具体的清理步骤如下：

1. 在日志最后选择一个磁道，读出其对应的磁带号 B 以及在磁带中的位置
2. 寻找持久性缓存中所有在磁带 B 中磁道
3. 读取磁带 B
4. 将磁带 B 的数据读取内存中合并，再写入到暂存区即持久性缓存的保留部分；这样可以防止在磁带写入数据时电源发生故障时导致数据丢失。
5. 从持久性缓存的暂存区将磁带 B 写回

虚拟守卫还为清理过程添加了以下两项优化：

1. 如果前 N 个磁道不在持久性缓存中，则在清理过程中可以跳过它们（如

暂存区中数据）。这将最坏情况下的清理开销减少了 50%。这项优化已被用于市面上的驱动管理式叠瓦式磁盘[1]

2. 如果来自同一磁带的多个磁道被缓存，虚拟守卫可能会选择对磁带进行部分清理。清理过程开始时如果磁道T_x以被缓存，则复写在磁道T_{x-1}停止而不用一直复写到磁带的底部。

为了减少缓存大小和主机指令停止时的等待时间，一个磁带可以分多个阶段进行清理，每个阶段都会读取，写入持续性缓存，然后重新写入一个数据缓冲区。读取操作和部分写入操作在各阶段之间交错执行。虚拟守卫使用大约 15MB 的清理缓存大小和商用的驱动管理式叠瓦磁盘[12,1]大致相同。

尽管大多是基于特定区域的叠瓦式转换层都会倾向于执行更过的后台清理，但后台清理可能会对虚拟守卫带来不良影响。通过将即将被写覆盖的临近磁道写入持久性缓存同时在磁带之间引入虚拟守卫磁道（空磁道），虚拟守卫允许对热点位置的重复写入，而不需要复制数据；在热点位置可以再次写入前，清洁操作会通过额外的数据移动消除这些冗余空间。

3. 测试

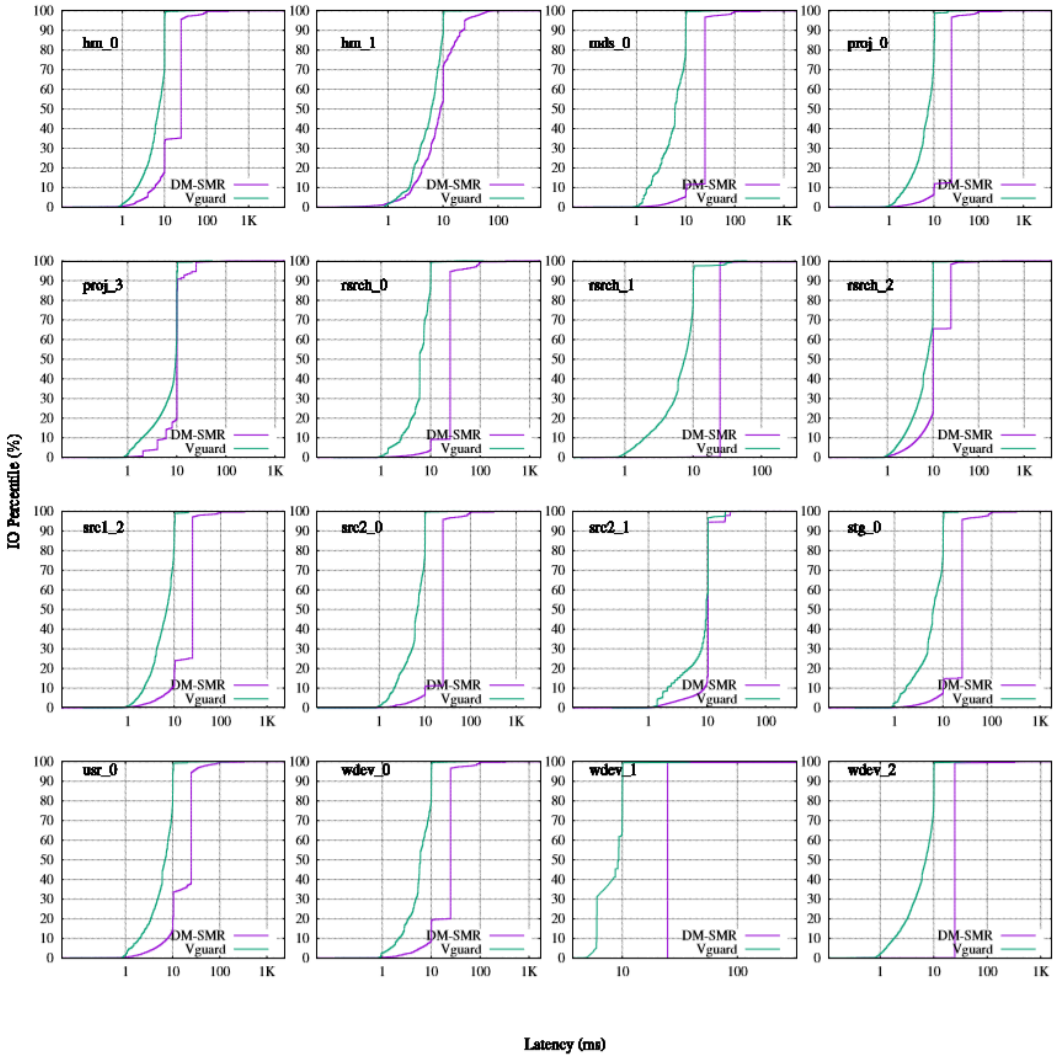
虚拟守卫通过对现在最精确的驱动管理式模拟器[12]进行扩展来实现（用 python 编写）。我们将虚拟守卫与模拟的希捷 5TB 叠瓦式磁盘（也是驱动管理式）进行性能比较，使用的磁盘参数和设置如下表所示，不对磁道的逻辑地址进行修改和缩放。追踪设置为平滑，在磁盘操作中无延迟。并且队列深度为一，写入缓存被禁止。

Parameter	Vguard	DM-SMR
Size	5TB	5TB
Form factor	3.5”	3.5”
RPM	5980	5980
Track lengths	1.8-0.9MB	1.8-0.9MB
Mapping type	Static	Static
Band size	20 tracks	20 tracks
Cleaning granularity	2 Bands	2 Bands
Cache location	Outer diameter	Outer diameter
Cache size	13.8K Tracks (24GB)	13.8K Tracks (24GB)
Mapping table size	~30KB	~1.3MB
α	9194	22986
β	9194	22986
Write cache	Disabled	Disabled
Read ahead	Disabled	Disabled

表一：模拟磁盘参数设置

3.1 MSR 测试

模拟了 Microsoft Research 测试集[10]中的十六种测试手段，代表了广泛的读写比率和大多数操作。图一展示了虚拟守卫和传统叠瓦式驱动的 CDFs 延迟大小，



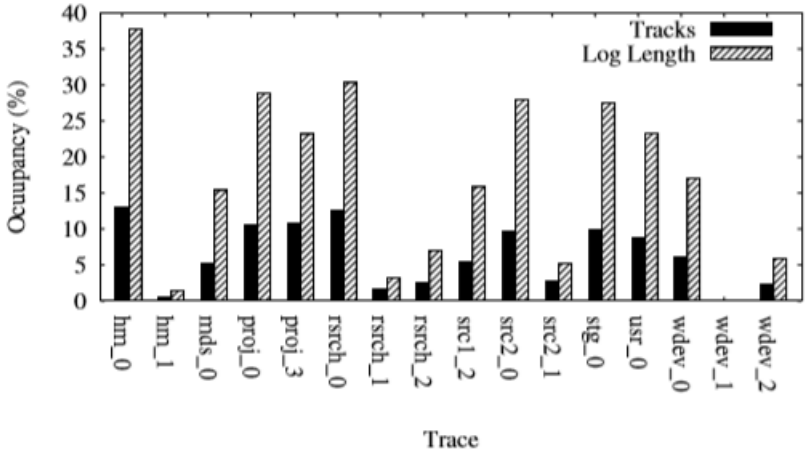
图一 CDFs 延迟

图中可见虚拟守卫的延迟减小的很快。在某些情况下，例如 src_2 和 mds_0，虚拟守卫的第 99 百分位延迟明显低于 DM-SMR 的第 10 百分位延迟。表二中可以看到其余延迟的详细信息。对于所有测试，虚拟守卫的 95%延迟大致是一次旋转（约 10ms），并且在很多情况下，最大延迟从几秒减少到约 50ms。

	95%		99%		99.5%		99.9%		Max	
Trace	Vguard	DM-SMR	Vguard	DM-SMR	Vguard	DM-SMR	Vguard	DM-SMR	Vguard	DM-SMR
hm_0	10.08	24.78	10.4	95.01	10.97	97.58	21.57	324.78	53.03	7,234.81
hm_1	10.1	24.78	10.4	56.55	10.4	64.06	20.13	74.62	50.60	592.04
mds_0	10.08	24.78	10.4	95.01	10.4	95.81	30.12	324.78	50.52	1,789.66
proj_0	10.4	24.78	18.7	95.01	20.43	99.78	21.56	324.78	52.29	4,341.17
proj_3	10.38	24.78	10.4	24.78	20.11	30.5	30.38	102.35	53.88	1,573.94
rsrch_0	10.08	28.79	10.31	96.24	10.68	105.24	30.92	324.78	117.81	1,657.22
rsrch_1	10.14	24.78	31.78	24.78	37	41.18	39.46	324.78	62.52	324.78
rsrch_2	10.08	24.78	10.08	31.59	10.1	81.64	30.79	324.78	47.83	4,082.31
src1_2	10.38	24.78	10.4	85.64	20.09	95.63	20.43	324.78	50.62	2,400.70
src2_0	10.08	24.78	10.18	95.09	10.4	98.77	30.21	324.78	60.59	3,428.12
src2_1	10.4	20.18	20.43	24.78	20.43	24.78	20.44	29.72	58.85	332.20
stg_0	10.11	24.78	10.40	95.08	14.89	97.47	30.12	324.78	123.92	1754.54
usr_0	10.16	26.96	10.4	85.3	20.18	95.36	24.85	324.78	158.74	2,396.69
wdev_0	10.08	24.78	10.23	95.01	10.49	97.19	30.14	324.78	56.56	1,710.40
wdev_1	10.08	24.78	10.08	24.78	10.38	24.78	38.04	324.78	39.25	324.78
wdev_2	10.1	24.78	10.19	24.78	11.51	95.01	33.21	324.78	49.32	1,628.12

表二：虚拟守卫和传统 STL 在 MSR 测试中的百分位延迟

由于测试的磁道中存在空间局部性，虚拟守卫磁道集合完全适合缓存，并且虚拟守卫能够在不进行清理的情况下就地执行任意数量的写入。相比之下，驱动管理式叠瓦转换层在每次写入时占用持久性缓存和映射中的空间，为所有长密集写入型输入指令进行清理。图三显示了各种 MSR 测试下虚拟守卫的高速缓存利用率，同时显示活动磁道的最大数量和使用的磁道总数，都用占总持久性缓存大小的百分比表示。我们看到，在任何情况下，这些测试都不会占用总 24GB 缓存的 13%，并且复制到持久性缓存的磁道总数不会超过触发清除操作的阈值 β 的 38%。



图三：高速缓存利用率

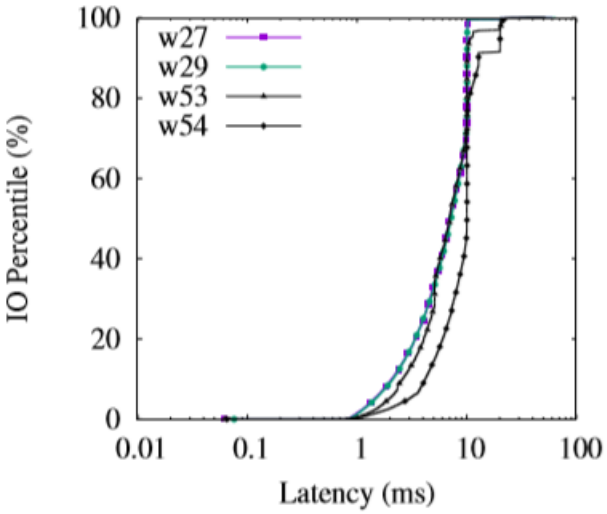
3.2 CloudPhysics 测试

另外一些实验是在 CloudPhysics[13]上一组更新的测试集上进行的，代表大量的实际操作。在表三，我们看到每个测试的写入总量和总容量，在图四中，我

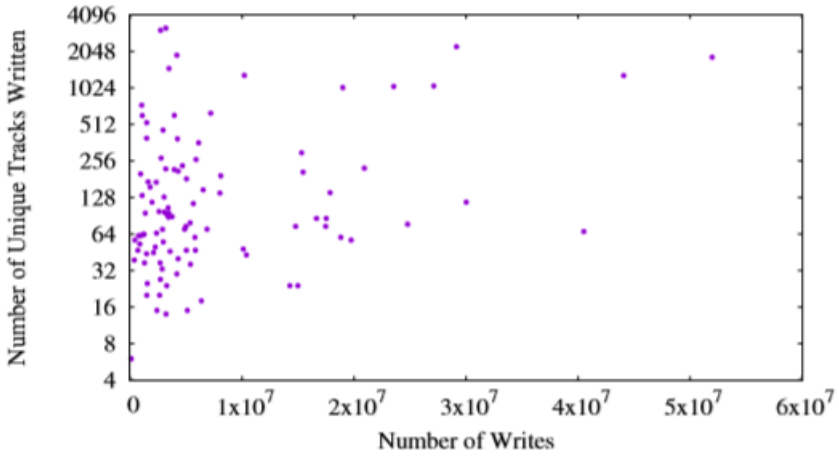
们看到了这些运行的结果;而且虚拟守卫能够处理整个测试而无需进入清除操作。事实上，对图五所示的所有 100 多个 CloudPhysics 测试中修改的多个特定磁道的分析表明，即使只拥有 24GB 的持久性缓存，没有一个测试有大到足以使虚拟守卫强制进行清除操作的写操作。

Trace	Number of writes	Drive Size
w27	3,182,636	1.95TB
w29	2,707,559	1.95TB
w53	4,162,497	1.5TB
w54	8,648,118	3.6TB

表三：写入大小与磁盘容量



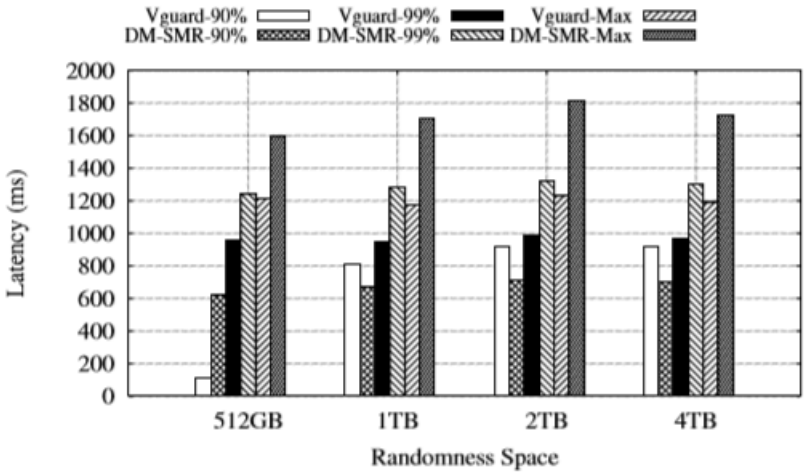
图四:虚拟守卫在 CloudPhysics 测试中的 CDFs 延迟



图五：在 CloudPhysics 测试中占用的磁道

3.3 随机写入

为了测试虚拟守卫的清除操作表现，我们编写了 4 个新测试，在 500GB，1TB，2TB 和 4TB 的地址空间上进行 100KB 大小的随机写入，同时在虚拟守卫和传统驱动管理式叠瓦磁盘上运行以作对比。这些实验的 CDFs 延迟如图六所示。虚拟守卫在最大等待时间中显示出约 30%的提升。在较小的磁盘占用时（小于 500GB）虚拟守卫的第 90 百分位延迟下降明显，但在其他情况下比传统驱动管理式叠瓦式磁盘较差。在写入较大的测试中可以预料到虚拟守卫和传统驱动管理转换层的延迟，因为叠瓦区域已被占满，每次写入都会触发强制清除操作。在这种情况虚拟守卫和传统转换层并无太大区别。



图六：虚拟守卫和传统 STL 在随机写入下的 CDFs 延迟

4. 总结

虚拟守卫提出一种新颖的叠瓦式转换层机制，在执行就地写入操作时将叠瓦方向上将被损坏的数据转入持久性缓存。因此不再需要消耗与写入数据量的大小的呈复杂函数对应的高速缓存空间，而是由写入地址决定的固定模式，并且不用考虑写入覆盖次数。在许多真实情况下，被保护的磁道完全可以很好保存在一个很小的持久性缓存（24GB）中。因为避免了叠瓦式磁盘固有的写放大问题，所有的写操作都可以在就地进行，可以提供接近传统磁盘的性能水平。但还需要做进一步的工作来比较虚拟守卫与传统磁盘的性能，以及真实物理设备的表现。

参考文献

[1] AGHAYEV, A., SHAFAEI, M., AND DESNOYERS, P. Skylight—a window on shingled disk operation. Trans. Storage 11, 4 (Oct. 2015), 16:1–16:28.

[2] CASSUTO, Y., SANVIDO, M. A., GUYOT, C., HALL, D. R., AND BANDIC, Z. Z. Indirection systems for shingled-recording disk drives. In Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on (2010), IEEE, pp. 1–14.

[3] HE, W., AND DU, D. H. Novel address mappings for shingled write disks. In Proceedings of the 6th USENIX conference on Hot Topics in Storage and File Systems (2014), USENIX Association, pp. 5–5.

[4] HE, W., AND DU, D. H. Smart: An approach to shingled magnetic recording translation. In 15th USENIX Conference on File and Storage Technologies (FAST 17) (Santa Clara, CA, 2017), USENIX Association, pp. 121–134.

[5] JONES, S. N., AMER, A., MILLER, E. L., LONG, D. D. E., PITCHUMANI, R., AND STRONG, C. R. Classifying data to reduce long-term data movement in shingled write disks. Trans. Storage 12, 1 (Feb. 2016), 2:1–2:17.

[6] KADEKODI, S., PIMPALE, S., AND GIBSON, G. A. Caveat-Scriptor: Write Anywhere Shingled Disks. In 7th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage 15) (Santa Clara, CA, 2015), USENIX Association.

[7] KREVAT, E., TUCEK, J., AND GANGER, G. R. Disks are like snowflakes: no two are alike. In Proceedings of the 13th USENIX conference on Hot topics in operating systems (2011), USENIX Association, pp. 14–14.

[8] LIN, C.-I., PARK, D., HE, W., AND DU, D. H. Hswd: Incorporating hot data identification into shingled write disks. In Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2012 IEEE 20th International Symposium on (2012), IEEE, pp. 321–330.

[9] MANZANARES, A., WATKINS, N., GUYOT, C.,LEMOAL, D., MALTZAHN, C., AND BANDIC, Z.Zea, a data management approach for smr. In 8th USENIX Workshop on Hot Topics in Storage and FileSystems (HotStorage 16) (Denver, CO, 2016), USENIX Association.

[10]NARAYANAN, D., DONNELLY, A., AND ROWSTRON, A. Write off-loading: practical power management for enterprise storage. In Proceedings of the 6th USENIX Conference on File and Storage Technologies (San Jose, California, 2008), USENIX Association, pp. 1–15.

[11]RUEMMLER, C., AND WILKES, J. An introduction to disk drive modeling. Computer 27, 3 (Mar. 1994), 17–28.

[12]SHAFAEI, M., HAJKAZEMI, M. H., DESNOYERS, P., AND AGHAYEV, A. Modeling smr drive performance. In Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science (New York, NY, USA, 2016), SIGMETRICS '16, ACM, pp. 389–390.

[13]WALDSPURGER, C. A., PARK, N., GARTHWAITE, A., AND AHMAD, I. Efficient MRC Construction with SHARDS. In Proceedings of the 13th USENIX Conference on File and Storage Technologies (2015), USENIX Association.