|  |  |
| --- | --- |
| 学号 | 2017282110242 |
| 密级 |  |

**海量存储课程作业**

|  |
| --- |
| **《Extending SSD Lifetimes with Disk-Based Write Caches》读书报告** |

|  |  |
| --- | --- |
| 院（系）名 称 ： | 计算机学院 |
| 专 业 名 称 ： | 计算机技术 |
| 学 生 姓 名 ： | 张宇光 |
| 指 导 教 师 ： | 何水兵 教授 |

二〇一七年十二月

目录

[1 论文信息 1](#_Toc500508601)

[1.1 论文题目 1](#_Toc500508602)

[1.2 论文链接 1](#_Toc500508603)

[1.3 作者信息 1](#_Toc500508604)

[2 论文技术背景 2](#_Toc500508605)

[3 Griffin概述 4](#_Toc500508606)

[3.1 其他的混合存储设计 4](#_Toc500508607)

[3.2 Griffin存储系统性能 5](#_Toc500508608)

[3.3 HDD作为写缓存的优点 6](#_Toc500508609)

[4 文章中原型设计与实现介绍 7](#_Toc500508610)

[5 关于整体系统的相关讨论 8](#_Toc500508611)

[6 总结及问题 10](#_Toc500508612)

# 论文信息

## 论文题目

Extending SSD Lifetimes with Disk-Based Write Caches

## 论文链接

<https://www.usenix.org/legacy/events/fast10/tech/full_papers/fast10proceedings.pdf#page=109>

## 作者信息

Gokul Soundararajan

联系方式：gokul@eecg.toronto.edu

作者Google信息：<https://xue.glgoo.net/citations?user=nw-7ErkAAAAJ&hl=zh-CN>

Vijayan Prabhakaran

联系方式：vijayanp@microsoft.com

Mahesh Balakrishnan

联系方式：maheshba@microsoft.com

Ted Wobber

联系方式：wobber@microsoft.com

# 论文技术背景

在经历了海量存储这门课程的学习之后，我学习到了很多相关知识，但是对于各个更加细分的领域还没有进行过仔细的探究。在查阅各种论文之后，我选择了一篇较为简单但是却很有意义的论文进行分析，看懂文章的内容然后加以整理解析，将其中的重要观点以及论述都一一阐明，这篇文章是关于Griffin混合存储系统的提出，基于SSD和HDD混合存储的使用。

SSD寿命：在以前作者做的几项研究中已经对SSD寿命进行了相应的评估。这些研究的共识是基于MLC的SSD的可靠性和性能随着时间的推移而降低。例如，误码率急剧增加，并且擦除次数增加（高达三倍），因为SSD寿命到期。这些趋势激发了我们工作的主要目标，即减少SSD擦除次数，从而延长其寿命。随着更少的磨损，SSD也可以提供更高的性能。

磁盘+SSD结构：为了结合旋转和固态介质的优点，已经提出了各种混合存储设备。以前的大多数工作是使用SSD作为硬盘顶部的缓存来提高读取性能。例如，Intel的Turbo Memory使用基于NAND的非易失性存储器作为HDD缓存。诸如Windows ReadyBoost之类的操作系统技术使用闪存（例如以USB驱动器的形式）来缓存通常被分页到HDD的数据。 Windows Ready-Drive适用于具有集成闪存的混合ATA驱动器，即使在硬盘驱动器停转时也可以进行读写操作。最近，研究人员已经考虑将HDD和SSD放在同一级别的存储层次上。例如，Combo Drive是一种异构存储设备，其中SSD和HDD的扇区被连接在一起形成一个连续的地址范围，其中数据是基于启发式放置的。由于存储地址空间分为两个设备，因此HDD中的故障可能导致整个文件系统无法使用。相比之下，Griffin使用HDD作为缓存，即使在HDD发生故障的情况下（尽管丢失了一些更新），它也可以提供一个可用的文件系统。同样，Koltsidas等。已经提出基于一组在线算法在两种媒体之间分割数据库存储。Sun的混合存储池由大量的SSD和HDD组成，以提高多核系统的数据访问性能。与上述作品相比，我们使用HDD作为写入缓存来延长SSD的使用寿命。虽然使用SSD作为读取缓存可能为笔记本电脑和台式机提供一些好处，但Narayanan等人已经证明它们在企业服务器环境中并不能证明有很大的好处。而且，任何迫使所有通过相对少量的闪存进行写操作的系统都将非常快速地穿过可用的擦除周期，极大地降低了这种方案的实用性。将HDD和SSD设置为同级的设置可以减少擦除周期并提供低延迟的读取访问，但是如果硬盘未被构造为日志，则会导致写入延迟。此外，硬盘故障可能导致数据丢失，因为它是一流的分区，而不是缓存。 SLC + MLC：最近推出了兼容SLC和MLC内存的混合SSD设备。例如，三星公司开发了一种混合存储器芯片，它包含SLC和MLC闪存存储器块。或者，可以将MLC闪存单元编程为单级或多级单元; FlexFS根据应用需求将存储动态分区到SLC和MLC区域。

其他架构使用SLC芯片作为缓存写入MLC的日志。这些研究强调SLC日志提供的性能收益，但不研究对系统寿命的影响。正如我们在第7节中所描述的那样，一个小的SLC写入缓存会比MLC设备更快磨损，而更大的缓存是昂贵的。

磁盘+磁盘：胡等人。提出了一种称为磁盘缓存磁盘（DCD）的体系结构，其中一个HDD用作日志随机写入转换为大型日志附件。在空闲时间，高速缓存的数据从日志到基础主磁盘为止。然而DCD的能力是提高性能，但我们的主要目标是增加SSD的使用寿命。

# Griffin概述

Griffin的设计非常简单：使用硬盘作为基于MLC的SSD的永久性写入缓存。所有写入都附加到存储在HDD上的日志中，并最终迁移到SSD，最好是在后续读取之前。将写缓存构建为日志允许Griffin以其快速顺序写入模式操作HDD。除了合并覆盖以外，写入缓存还增加了SSD观察到的工作负载的顺序性;如前一节所述，这会增加写入寿命。

由于成本是SSD部署的最大障碍，因此专注于便宜的基于MLC的SSD的写入缓存，因为低写入寿命是一个重要的限制因素。由于MLC设备的连续写入带宽通常等于商品HDD的70-80MB/s，所以它们是基于HDD写入缓存的优秀候选者[13]。

Griffin增加了基于MLC的SSD的写入寿命，而不会增加总成本;截至撰写本文时，350GB SATA HDD的成本大约为50美元，而128GB基于MLC的SSD大约为300美元。相比之下，128GB的基于SLC的SSD的写入寿命比MLC的高出约4至5倍。Griffin也提高了写入寿命，而不会显着改变MLC器件的可靠性特性。虽然HDD写入缓存代表了一个额外的故障点，但任何此类事件都会使文件系统在SSD上保持不动，只会导致最近丢失的数据。在5.3节讨论故障处理。

## 其他的混合存储设计

使用RAM，非易失性RAM和旋转媒体的各种组合的其他混合设计显然是可行的。由于对所有选项的深入比较分析超出了本文的范围，因此简要介绍一些其他设计，并将其与Griffin进行定性比较。

•NVRAM作为HDD的读取缓存：鉴于其出色的随机读取性能，NVRAM（例如SSD）可以在较大的硬盘驱动器之前作为读取缓存工作。然而，与RAM中的基于OS的文件缓存相比，较小的NVRAM可能仅提供增量性能优势，而较大的NVRAM缓存既昂贵又容易随着缓存内容的变化而磨损。任何使用旋转介质进行主存储的设计都将以比Griffin低的成本进行扩展。然而，随着闪存密度的增加，这种成本差异可能会减小。

•NVRAM作为SSD存储的写入缓存：Griffin设计可以将NVRAM作为写入缓存来代替HDD。使用NVRAM的有效性取决于两个因素：1）是否使用SLC或MLC闪存;2）命中写入缓存的读取比率，从而破坏顺序记录。NVRAM的使用也能够做到更好的节能。但是，所有这些好处的成本比配置HDD缓存的Griffin更高，特别是如果SLC闪存用于写入缓存。稍后，用SLC和MLC写入缓存（6.4节）评估Griffin的性能，并探索所需的最小写入缓存大小（第7节）。

•RAM作为SSD存储器的写入缓存：RAM可以实现快速有效的写入缓存，然而RAM的首要问题在于它不是持久的（缺少某些电源连续性配置）。增加RAM大小或周期性定时器间隔时间可能会减少写入存储的次数，但这只是以更大的漏洞窗口为代价，在此期间，电源故障或崩溃可能会导致更新数据丢失。此外，基于RAM的写入缓存可能不适用于所有工作负载;例如，稍后会发现，对于某些工作负载（第6.1.2节），需要超过1小时的缓存才能获得更好的写入节省;易失性缓存并不适合这么长的持续时间。

## Griffin存储系统性能

Griffin面临的主要挑战是增加SSD的写入寿命，同时保持读取性能。写缓存是一种众所周知的技术，用于缓冲对一组块的重复写入。然而，Griffin与传统的高速缓存设计相差甚远，它们通常使用小型，快速和昂贵的介质（如易失性RAM或非易失性电池备份RAM）来缓存针对较大和较慢后备存储的写入。Griffin的HDD写入缓存既便宜又持久，实际上可能比后备SSD大;因此，从写入缓存到SSD的脏数据的翻转不受容量和同步写入的驱动限制。

但是，Griffin的HDD写入缓存比读取操作的后备SSD要慢，这将转化为HDD日志中的高延迟随机I/O。另外，读取操作可能会破坏HDD接收的顺序写入流，从而将记录带宽降低一个数量级。因此，为了避免从硬盘昂贵的读取，脏数据必须在SSD再次读取之前转移到SSD。

因此，Griffin的性能决定于竞争性要求——数据必须保存在HDD中才能缓存重写，数据必须从HDD中流出以防止昂贵的读取。用以下两个指标量化这些数据：

•写入节省：这是防止到达SSD的总写入的百分比。例如，如果混合设备接收到60M写入，而SSD接收到45M，写入节省25％。理想情况下，希望写入节省尽可能高。

•读取损失：这是由HDD写入缓存服务的总读取的百分比。例如，如果混合设备接收到50M的读取数据，而HDD接收到这些读取的1M数据，则读取损失为2％。理想情况下，希望读取的损失越低越好。

如果甲骨文事先告诉Griffin的数据读取将不会被阅读的损失；在即将进行读取之前，所有这些块都可以转移到SSD。没有读取损失，最大写入节省量是依赖于工作负载的，并且基本上是没有插入读取的连续重写频率的度量。在最坏的情况下，如果没有重写，就不会有写入节省，即没有中断读取连续写入块。理想的HDD写入缓存实现了最大的写入节省，并且没有任何工作负载的读取损失。

要理解理想化HDD写入缓存的性能，请考虑以下顺序的写入和读取特定块：写入→写入→写入→读取→写入→写入。如果没有写入缓存，这个序列将导致一次读取和五次写入SSD。一个理想化的HDD写入缓存将在每次读取之前立即合并连续的写入和闪存数据到SSD，从而导致SSD的操作序列包含两个写入和一个读取：写入→读取→写入。因此，在这个简单的例子中最大的写入节省是3/5或60％。

Griffin尝试通过从两个方面控制策略来实现理想化的HDD写缓存的性能：要缓存什么数据，缓存多长时间。每种情况下的政策选择都由实际工作量的特征来决定，将在下一节中讨论。使用这些不同的策略，Griffin能够在读取损失和写入节省之间的权衡曲线中实现不同的点。

## HDD作为写缓存的优点

通过分析桌面和服务环境的迹线，探讨基于HDD的写入缓存的好处的分析有两个方面。首先，展示了一个理想化的基于HDD的写入缓存可以为这些记录提供显着的写入节省;换句话说，覆盖通常出现在现实的工作负载中。其次，文章在这些重写中寻找空间和时间模式，这可以帮助确定Griffin的缓存策略。

我们发现，实际上桌面和服务工作负载中经常出现块重写现象，验证了Griffin背后的核心思想。此外，块重写显示空间和时间的局部性，提供有用的实用缓存策略，可最大限度地节省写入成本，避免大量的读取损失。

# 论文原型设计与实现介绍

到目前为止，文章中已经从抽象的角度讨论了基于HDD的写入缓存，目的是确定哪些策略能够指示在HDD中缓存什么数据以及何时将其迁移到SSD中。唯一指标是写入节省和读取损失。

然而，Griffin的政策选择和实施也受到其他因素的严重影响。一个重要的考虑是迁移开销，直接（总字节）和间接（HDD顺序的损失）开销。例如，从写入节省和读取损失的角度来看，由假想oracle提供的迁移时间表可能是最佳的，但是可能需要以小的增量不断迁移数据，破坏HDD的访问模式的顺序性。

另一个主要问题是容错性；Griffin中的HDD代表一个额外的隐患，某些策略可能会使混合系统比未修改的SSD更不可靠。例如，将数据推送到SSD而将相关文件系统元数据留在HDD上的迁移计划将非常容易丢失数据。考虑到迁移开销和容错的双重担忧，Griffin使用两种机制来支持关于要缓存什么数据以及缓存多长时间的策略：重写率和迁移触发器。

文章中实现了一个跟踪驱动运行的模拟器和用于评估Griffin的用户级实现。模拟器用于测量写入节省、HDD读取损失和迁移开销，而用户级别实现则用于通过使用原始设备接口将跟踪中的I/O传输到实际的HDD/SSD组合来获得实际的延迟测量。

在写入块时，Griffin将I/O重定向到HDD日志的尾部，并将其新位置记录在内部内存映射中。内存映射的最近内容会周期性地转移到硬盘上以用于故障恢复。在读取块时，Griffin从相应的设备读取块的最新副本。

每当选定的迁移触发解除，缓存的数据将从HDD迁移到SSD。为了识别日志写入和逻辑SSD块之间的映射，Griffin从HDD读取块映像（如果它不在内存中）并重建映射。在迁移时，Griffin会尽可能按顺序读取日志内容，只跳过较旧版本的数据块，根据逻辑地址对记录的数据进行排序，然后将其写回到SSD中。正如我们稍后所示，此迁移改善了向SSD写入数据的顺序性。

即使写入是按顺序记录的，HDD也可能产生旋转延迟。这种旋转等待时间可以通过使用小缓冲区（例如，128KB）在将数据写入HDD之前缓存或通过使用像控制范围写入的新机制来使它最小化。

# 关于整体系统的相关讨论

基于文件系统的设计：可以在文件系统级而不是块设备级执行Griffin。这种方法有三个好处：首先，文件系统可以利用块之间语义关系的知识来更好地利用原文中描述的空间局部性。其次，可以通过修改现有的日志文件系统来将更新日志存储在HDD上以及SSD上的实际数据上，从而很容易实现Griffin尽管目前的日志文件系统通常被设计为仅存储日志中的元数据更新，我们想要缓冲的许多重写发生在用户数据中。

文件系统设计的第三个优点是可以获得更好的信息，使其能够接近理想化的HDD写缓存的性能。回想一下，理想化的缓存需要一个oracle来通知它即将发生的块读取，所以脏数据可以及时迁移，以避免从硬盘读取。在分组层面上，这样的假设不存在，我们不得不采用启发式的迁移策略。但是，在文件系统级别，缓冲区缓存中块的逐出可以用来表示即将来临的读取。只要文件系统在其缓冲区缓存中存储了一个块，它就不会将该块的读取内容发送给存储设备;一旦它清除了该块，任何后续的读取都必须从设备中提供服务。因此，从缓冲区高速缓存中逐出HDD的块到SSD的策略将导致最大的写入节省，而不会造成读取损失。但是，块设备具有显着的优点，即不需要修改软件堆栈，可以与任何操作系统或体系结构配合使用。此外，我们的评估显示，我们使用的简单设备级迁移策略对于近似理想缓存的性能非常有效。

以闪存作为写入缓存：虽然Griffin使用HDD作为写入缓存，但它也可以使用较小的SSD，并且性能更好。由于SLC闪存价格昂贵，因此写入缓存的大小很小是至关重要的。但是，写缓存还必须至少支持与基于MLC的后备SSD一样多的擦除，这需要一定的最小尺寸。

由于每个SLC块可以忍受10倍的MLC块的擦除，所以经历与MLC设备相同数量的写入的SLC设备将需要是MLC的十分之一大，以持续长久。如果SLC的写入次数是MLC的两倍，那么就需要这么大。因此，缓存设置实现了50％的写入节省导致向SLC发送的写入次数比MLC多两倍。这需要SLC缓存至少是MLC的五分之一。例如，如果MLC设备是80 GB，那么我们需要至少16GB的SLC缓存。在这个分析中，我们假设了一个理想的FTL，它执行页面级映射，一个完美的顺序写入流，以及MLC和SLC设备。如果MLC的块大小是SLC块大小的两倍，就像当前设备的情况一样，所需的SLC大小在一个完美的顺序工作负载下保持不变，但会降低更多的随机工作负载；为了简洁，本文省略了块大小分析的细节。认为16GB SLC写入缓存（对于80GB MLC主存储）开销很大，足以证明Griffin选择缓存介质的合理性。

功耗：Griffin设计中可能出现的一个主要问题是功耗。由于HDD比SSD消耗的功率更大，因此Griffin的功耗预算要高于普通SSD。缓解这个问题的一个方法是使用一个更小，功率更高的HDD，例如1.8英寸的驱动器，这个驱动器可以提供稍低的带宽。例如，东芝的1.8英寸HDD[28]消耗大约1.1瓦的功率，读取或写入的功耗大约为1.0瓦，这与Mircon SSD的功耗相当，从而在功耗，性能和使用寿命之间进行权衡。此外，桌面工作负载可能会有空闲的时间间隔，在这段时间内可以降低HDD缓存的速度以节省电力。

最后，我们可以使用单个HDD作为多个SSD的写入缓存，从而降低每个SSD的功率溢价（以及硬件成本）。按照Intel X25-M的规格，单个SSD支持3.3K随机写入IOPS，或者大约13MB/s，而HDD可以支持70到80MB/s的连续写入。因此，如果单个HDD全部运行在完全随机的工作负载上，则可以跟上多个SSD，但是当组合的工作负载的数据速率超过HDD速度时，需要进行不重要的工程来禁用高速缓存。

# 总结及问题

随着新技术的诞生，旧技术在系统演化过程中可能会发挥新的作用。文章展示了被广泛用作主存储的硬盘驱动器可以用作基于MLC的SSD的缓存。Griffin的设计是由工作量和硬件特性驱使。经过对Griffin政策和性能的仔细评估，发现Griffin具有显着改善SSD使用寿命的潜力，而不牺牲性能。

文章中提供了一种基于SSD和HDD的混合存储系统，原文中的缓存策略和超时迁移策略在最后的探讨中，原文作者只利用了全缓存策略，虽然作者说明了原因，但是我认为对于选择性缓存是不是可以在特定的情况下进行考虑，在实际应用中，选择性缓存策略会不会给SSD的写寿命带来更大的提升，这样对于减少读取损失和写入节省会不会有更大的好处，结合实际情况可能会更好一点？我有个大胆的想法，能不能够利用机器学习的思路来解决这样的问题？

感谢老师几个月来的教育，老师的上课风格我很喜欢，曾经完全不了解海量存储系统的我也在课堂上了解了很多关于这方面的知识。老师在讲完自己的课堂知识后还让我们对于海量存储方面的论文进行进一步了解，在制作小组PPT时，对于海量存储中的冗余并不等同于容错有了更加深入的理解，曾经我以为只要添加了冗余，即增加了相应的备份就不用担心系统出现故障，但是在经过学习之后，制作PPT的过程中对于以前的想法彻底清除了，容错并不能通过简单的冗余来实现，对于这方面的工作，世界上也没有谁能够完全解决这个问题，还需要后来的人对文章提出的问题进行相应的研究，我相信不久的将来一定能在很大程度上解决这个问题。

对于老师布置的最终的作业，我也很感谢，让我对于Griffin这样的一个存储系统有了一个较为概括的理解，虽然很多底层的更加深入的研究我没有研究，但是就对于Griffin，我学到了很多。Griffin利用HDD和SSD各自的优点组成一套复合的存储系统，期间还研究了各种方案的优缺点这里就不再讲了，文章中我都详细地进行了阐述。

再次感谢老师对我的教导！