ZNS SSD 性能分析及针对性改进

王梓尧[1)]

[1)](华中科技大学计算机科学与技术学院，武汉 中国 430074)

摘 要 NVMe分区命名空间(ZNS)作为一种新的存储接口，其中逻辑地址空间被划分为固定大小的区域，每个区域必须按顺序写入，以便实现对闪存友好的访问。由于 ZNS 使用顺序写方式，因此需要 LFS(日志结构化文件系统)访问 ZNS SSD。这种实现方式扩大了 SSD 容量，同时减少了写放大，并通过区域向主机开放了数据放置和垃圾收集。虽然 SSD 可以在当前的 ZNS 接口下进行简化，但其对应的 LFS 必须承担段压缩开销。在本文中，讨论了 ZNS SSD 在不同工作负载下的性能。然后，从并行性、孤立性和可预测性方面考察了 ZNS SSD 的相应特征。同时，讨论了现阶段针对 ZNS SSD 的进一步性能优化的实现方案，将其与传统 ZNS 文件系统的性能进行比较，结果显示本文提出的 ZNS+存储系统的文件系统性能是普通基于 ZNS 的存储系统的 1.33 ~ 2.91 倍。

关键词 日志结构文件系统；写放大；并行性；可预测性；回写；内部插入

ZNS SSD performance analysis and improvement

Wang Ziyao[1)]

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

Abstract NVMe Zoned Namespace (ZNS) serves as a new storage interface in which the logical address space is divided into fixed-size regions, each of which must be written sequentially for flash-friendly access. Since ZNS uses sequential writing, LFS (Log Structured File System) is required to access ZNS SSD. This implementation expands SSD capacity while reducing write amplification and opens up data placement and garbage collection to the host through zones. Although SSD can be simplified under the current ZNS interface, its corresponding LFS must bear segment compression overhead. In this paper, the performance of ZNS SSD under different workloads is discussed. Then, the corresponding characteristics of ZNS SSD are examined in terms of parallelism, isolation and predictability. At the same time, the further performance optimization implementation scheme for ZNS SSD is discussed at this stage, and compared with the performance of the traditional ZNS file system, the results show that the file system performance of the ZNS+ storage system proposed in this paper is 1.33 times that of the ordinary ZNS-based storage system 1.33~ 2.91 times.

Key words LFS; overwrite; parallelism; predictability; copyback; internal plugging

1 概述

如今，许多研究都提出了揭示 SSD 内部结构以提高性能和 I/O 稳定性。在 NVM Express 的标准化下，ZNS SSD 就是其中之一，它使用区域的概念公开其地址空间。通过将不同的工作负载分配到不同的区域，它们有机会降低 WAF（写放大因子），最终提高性能和寿命。ZNS SSD 的另一个特点是闪存管理，如映射和垃圾收集是在主机级进行的。传统的 SSD 在设备级使用 FTL（Flash 转换层）处理闪存特性，如写入前擦除要求和有限的续航时间。然而，ZNS SSD 将大多数 FTL 功能转移到主机，这使得减少 DRAM 的使用和 SSD 中的超额配置区域成为可能。目前许多供应商积极宣布推出他们的 ZNS SSD 解决方案。

然而，ZNS SSD 却产生了一些新问题。一个问题是如何在主机级别上管理区域，包括区域重置和垃圾收集。此外，ZNS SSD 还有一个约束，称为顺序写约束，即数据必须按顺序写入一个区域，这使得如果不对其进

行针对性优化，其随机性能将不甚理想。为了针对 ZNS SSD 进行优化，我们需要了解 ZNS SSD 的性能特征。在接下来的部分，我们会对 ZNS SSD 与传统 SSD 进行多维度比较，使得更加直观的展示与传统 SSD 相比的性能增益。同时，提出一种改进 ZNS SSD 的方案，使得性能在原来基础上更进一步。最后，得出结论。

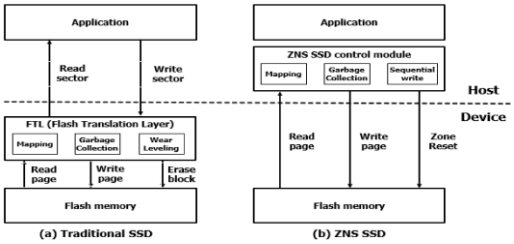


图 1 传统 SSD 和 ZNS SSD 的比较

2 背景

2.1 ZNS SSD 的特性

ZNS SSD 和传统 SSD 之间的区别如图 1 所示。固态硬盘是由闪存（Flash）组成的，它有一些特点，如覆盖限制和擦写次数有限。为了克服这些限制，开发了各种功能，如外接更新和映射、垃圾收集、磨损均衡和坏块处理。现在，问题是这些功能部署在哪里。从概念上讲，计算机系统可以分为两个层次，主机和设备。在传统 SSD 中，这些功能部署在设备级，通常称为 FTL （Flash 转换层）。这种方法的好处是它隐藏了闪存的特性，并将 SSD 抽象为可以通过扇区接口访问的磁盘那样的一般块设备。然而，它的缺点是可能会导致语义鸿沟，因为两个级别之间没有意识和冗余的软件模块。ZNS SSD 采用相反的方法，在主机级部署功能。它们是一种 OCSSD（Open Channel SSD），公开 SSD 内部，并允许主机级软件直接控制闪存。与 OCSSD 相比，ZNS SSD 有一些区别：1)将其地址空间划分为多个区域；2)需要主机级的区域管理，如区域重置和垃圾收集；3)数据必须按顺序写入一个区域。

ZNS 固态硬盘有几个优点。首先，它们可以通过将不同的工作负载分配到不同的区域来减少 WAF，最终有助于提高性能和寿命。此外，它们还通过将不同的用户分配到不同的区域来提高 I/O 确定性。此外，他们通过将大部分 FTL 功能转移到主机中来减少 DRAM 的使用和介质过度配置。通常，映射和垃圾收集被移动到主机中，而特定于设备的功能（如 ECC（错误纠正码）和坏块处理）仍然保留在设备级别。

2.2 ZNS SSD 的短板

关于 ZNS 的堆栈：通常，新的存储接口需要修改软件堆栈。对于 ZNS，我们需要修改两个主要的 IO 堆栈组件，文件系统和 IO 调度器。首先，必须用附加日志记录文件系统(如日志结构文件系统(LFS))替换现有的更新文件系统(如 EXT4)，以消除随机更新。因为 LFS 的一个段是通过追加日志按顺序写入的，所以每个段可以映射到一个或多个区域。其次，IO 调度器必须保证按顺序提交区域的写请求。例如，可以使用每个区域的有序队列，调度器只需要确定不同区域之间的服务顺序。

主机开销的增加：在 LFS 的追加日志方案下，脏段的废弃块必须通过段压缩(也称为段清理或垃圾收集)回收，将段中的所有有效数据移动到其他段，使段干净。压缩会调用大量的复制操作，特别是在文件系统利用率较高的情况下。必须执行主机端 GC 以换取使用不进行垃圾回收的 ZNS SSD，尽管可以避免重复 GC。主机端 GC 的开销比设备端 GC 高，因为主机级块复制需要 IO 请求处理、主机到设备的数据传输以及读取数据的页面分配。

3 性能评估

3.1 性能基准测试简介

为了定量评估 ZNS SSD 的性能特征，设计了一个分析工具，如图 2 所示。它由三个关键软件组件组成，即工作负载生成器、性能监视器和通用 SSD 管理器。

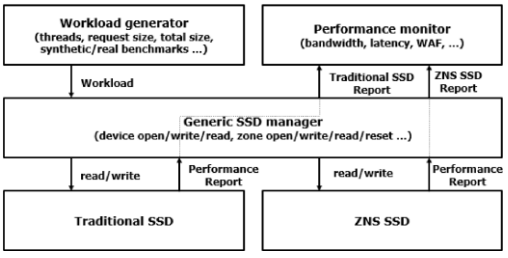


图 2

工作负载生成器使用合成基准和真实

基准(例如 fio 基准)创建具有不同模式的 I/O 请求，例如顺序和随机。它还允许配置一些控制参数，包括线程数、总 I/O 大小、每个请求大小和 LBAs。性能监控器在执行工作负载时测量延迟、带宽和 WAF 等指标。此外，它还支持跟踪功能，显示如何在时间和空间维度上处理每个 I/O 请求。

通用 SSD 管理器直接操作 ZNS SSD，并支持区域重置/打开/关闭和页面读/写接口。本文中使用了一个由 ZNS SSD 供应商提供的真实 ZNS SSD 原型。该原型机容量为 1TB，划分为 1024 个分区，每个分区大小为 1GB，如表 1 所示。

Item	Specification
SSD Capacity	1TB
Size of a Zone	1GB
Number of Zones	1024
Interface	PCIe Gen3
Protocol	NVMe 1.2.1

表 1

3.2 性能分析

3.2.1 并行性

图 3 显示了当我们将请求大小从 4KB 更改为 128KB 时的写和读延迟。由于本文使用的 ZNS SSD 设备与传统 SSD 设备规格不同，使用实际测量值进行比较有失公允。此外，本研究的目的是分析 ZNS SSD 的性能特征，而不是直接与传统 SSD 进行比较，所以我们用相对值来画这个图，以便更清楚地揭示特征。从图 3 中，我们可以观察到请求大小对 ZNS SSD 的写延迟有很大的影响。例如，当我们将请求大小从 4KB 更改为 8KB 时，ZNS SSD 可以减少 50%的延迟，而传统 SSD 减少 16%。对于读取情况，ZNS SSD 和传统 SSD 显示出类似的趋势。对于 ZNS SSD 开发人员来说，他们需要设计一种以更小的单位来摊销写请求的机制。使用 DRAM 写缓冲区不是一个好的选择，因为它与 ZNS SSD 所追求的方向相反

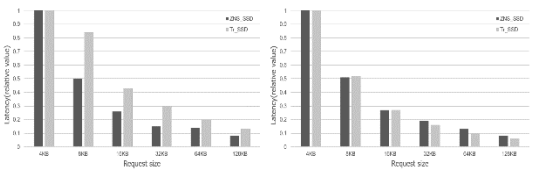


图 3

3.2.2 隔离性

为了评估 ZNS SSD 的隔离能力，我们同时与多个其他线程一起执行工作线程。具体来说，我们在不同的区域上运行所有线程，并测量其 IOPS，结果如图 4 所示。

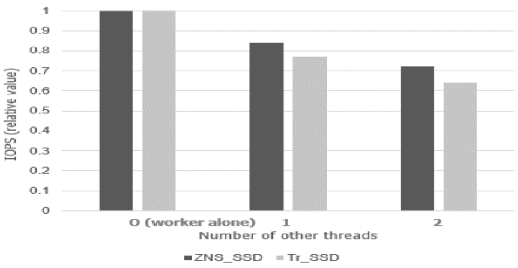
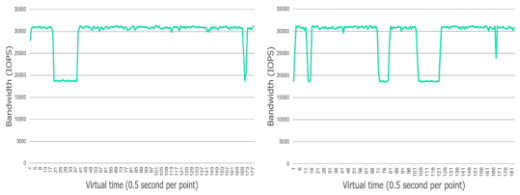


图 4

图 4 显示 ZNS SSD 比传统 SSD 具有更好的隔离性能。例如，当我们同时运行两个其他线程时，工作线程的性能在 ZNS SSD 中下降了 28%，而在传统 SSD 中下降了 36%。这是因为传统 SSD 具有更多的共享资源，如 FTL 和重叠的地址空间。但是，隔离能力比预期的要弱得多。首先，我们预期工作线程不会受到其他线程太多的干扰，因为它们访问不同的区域。但我们的观察发现，区域共享 ZNS SSD 中的通道和芯片等资源。

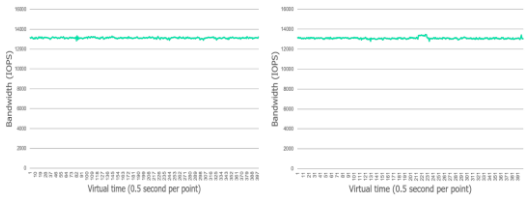
3.2.3 可预测性

传统 SSD 的一个显著的缺点是由于在 SSD 中由 FTL 触发的垃圾收集而导致的意外性能下降。为了检查这种现象，我们进行了一个实验，在传统 SSD 和 ZNS SSD 的初始利用率分别为 60%和 90%的情况下写入 10GB 文件，分别如图 5 和图 6 所示。图中的每个点都是每 0.5 秒测量一次的带宽值。



(a) 60% utilization (b) 90% utilization

图 5



(a) 60% utilization (b) 90% utilization

图 6

如图所示，在传统 SSD 中确实会出现

意想不到的性能下降，这种情况在利用率较高的情况下更为频繁，如图 5 所示。相反，在 ZNS SSD 中没有监测到这种退化，如图 6 所示。这一观察揭示了 ZNS SSD 是增强可预测性的良好基础。但这个结果并不意味着 ZNS SSD 中没有垃圾收集开销。

4 ZNS SSD 的改进

基于第 2 节中对于 ZNS SSD 的不足方面的叙述，虽然 ZNS SSD 相较传统 SSD 已经有了非常明显的性能提升，但仍然需要对其进行进一步的改进，以使其性能得到进一步的提高。

4.1 ZNS+文件系统

支持 LFS 的 ZNS：我们需要一些设备级支持来减轻 LFS 的段压缩开销。可以考虑两种方法：压实加速和压实避免。我们提出了一种新的支持 LFS 的 ZNS 接口，称为 ZNS+，并通过两个新命令 zone_compaction 和 TL_open 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：受害者段选择、目标块分配、有效数据复制和元数据更新。尽管所有其他操作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复制更快。对于压缩加速，ZNS+使主机能够通过 zone_compaction 将数据复制任务卸载到 SSD。ZNS+的稀疏顺序重写接口是 ZNS 中密集顺序追加写入约束的宽松版本。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 将稀疏的顺序写入请求转换为密集的顺序请求，方法是用同一段中未触及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闪存块。与原始 ZNS 相比，ZNS+不但有显著的扩展，同时 ZNS+ SSD 可以提供与原始 ZNS SSD 相同的优点，如映射表小、无重复 GC、无过度调配空间以及性能隔离/可预测性。

支持 ZNS+的文件系统：文件系统还需要进行调整，以利用 ZNS+的新功能。首先，SSD 内部数据复制操作将根据源和目标逻辑块地址（LBA）使用不同的复制路径。例如，当两个 LBA 被映射到同一闪存芯片时，可以利用闪存的回写操作，这在闪存芯片内移动数据而不进行芯片外的数据传输，从而减少数据迁移延迟。复制操作目前在标准 NAND 接口中，其在 SSD 内部垃圾收集中的可行性已被许多研究证明。为了充分利用回写操作，我们提出了用于段压缩的回写感知块分配，它尝试分配数据拷贝的目标 LBA，使得目标数据的源 LBA 和目标 LBA 都映射到同一闪存芯片。该技术可以扩展到 SSD 的其他快速复制路径。其次，由于 ZNS+同时支持段压缩加速和线程日志记录，因此主机文件系统需要选择其中一种段回收策略。基于此，本文提出了 ZNS+的混合段回收技术，该技术根据回收成本和收益选择线程日志记录或段压缩。

4.2 性能评估

在实验中，使用了以下两种不同版本的 ZNS+：IZC 和 ZNS+。在 IZC 中禁用线程日志记录时，在 ZNS+中启用线程日志记录并使用混合段循环。在实验中默认使用 PPA 顺序封堵。将 ZNS+方案与使用原始 F2FS（ZNS 补丁版本）和 ZNS SSD 的 ZNS 进行了比较。ZNS 使用主机级拷贝执行段压缩，不使用线程日志记录。因为每个工作负载都会生成没有空闲间隔的写请求，所以 F2FS 不会调用后台压缩。

4.2.1 段压缩性能表现

图 7 显示了 ZNS 和 IZC 各种基准的平均段压缩延迟。对于 ZNS 和 IZC，压缩时间分别分为四个阶段（初始化、读取、写入和检查点）和三个阶段（初始、IZC 和检查点）。初始化阶段读取与受害段相关的所有文件的几个元数据块。因为这些元数据通常被缓存在页面缓存中，所以初始化阶段很短。

与 ZNS 相比，IZC 通过消除主机级复制开销和利用回写操作，将区域压缩时间缩短了 28.2–51.7%。在 fileserver、varmail、tpcc、YCSB-a 和 YCSB-f 的工作负载期间，所有存储内复制操作中的复制操作比率分别为 87%、74%、81%、83%和 83%。尽管 filebench 工作负载的 IO 流量很大，但 OLTP 工作负载小。因此，在 filebench 工作负载期间，IZC 的性能改进更为显著，因为存储内复制操作减轻了对用户 IO 请求的干扰，从而使用主

机资源和主机到设备 DMA 总线。

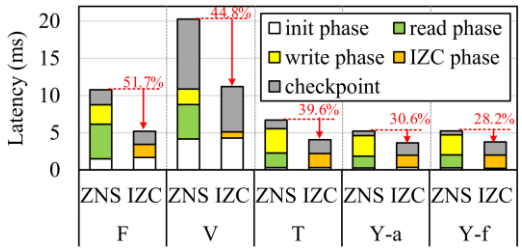


图 7

表 2 比较了不同闪存介质中不同复制方案的带宽。IZC 的性能增益对于速度更快的闪存介质更为显著，因为主机 IO 堆栈为速度更快的快闪介质提供了更大的 IO 延迟。IZC-H 和 IZC-D 分别将 89.6%和 99.4%的块复制操作卸载到 ZNS+ SSD。这表明大约 10.4%的待复制块缓存在主机 DRAM 中（干净：9.8%，脏：0.6%）。使用 TLC 闪存时，IZC-H 优于 IZC-D，因为 TLC 闪存读取延迟比主机级写入请求处理开销更长。然而，当使用 MLC 闪存时，IZC-H 和 IZC-D 之间的性能差异减小。如果通过使用 ZNAND 进一步缩短闪存访问时间，IZC-D 将提供更好的性能（即，无论目标块是否已缓存，都将所有拷贝请求卸载到存储器，从而获得更好的性能）。

	TLC	MLC	ZNAND
ZNS	79.5 (1.00x)	84.5 (1.00x)	104.0 (1.00x)
IZC-H	113.4 (1.43x)	154.6 (1.83x)	218.9 (2.10x)
IZC-D	96.5 (1.12x)	148.0 (1.75x)	242.4 (2.33x)

表 2

4.2.2 线程日志记录性能表现

本文比较了不同文件系统利用率时 ZNS 方案的性能。通过更改目标工作负载的文件集大小，我们控制了文件系统的利用率。图 8 显示了文件服务器工作负载的每种技术的工作负载吞吐量和写入放大因子（WAF）。WAF 是文件系统调用的总写入流量（包括数据块写入、节点块写入、元数据更新、段压缩和内部插入）除以用户工作负载生成的写入流量。IZC 和 ZNS 的 WAF 值相似。随着文件系统利用率的增加，WAF 会增加，因为段压缩必须复制更多的有效块，并且段压缩的调用频率会更高。由于线程日志记录减少了节点和元数据更新的数量，因此 ZNS+显示的 W AF 值低于 IZC。IZC 或 ZNS+比 ZNS 的性能增益随着文件系统利用率的增加而增加，因为在文件系统利用更高的情况下，段回收成本更为显著。

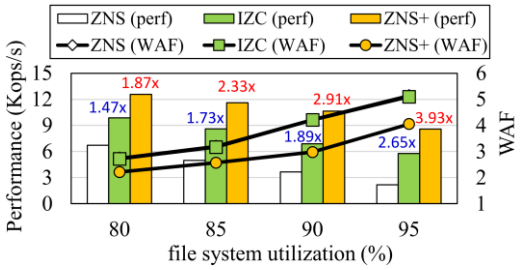


图 8

5 结论

ZNS SSD 相较于传统的 SSD，可以通过将不同的工作负载分配到不同的区域来减少 WAF，最终有助于提高性能和寿命。此外，它们还通过将不同的用户分配到不同的区域来提高 I/O 确定性。但 ZNS SSD 也有随机性能相对不足的缺点，通过 ZNS+文件系统的改进，可以使得 ZNS+ SSD 在拥有 ZNS SSD 原本拥有优点的基础上，进一步提升其随机性能，从而为其未来可能的广泛应用奠定了基础。

参考文献

[1] 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.

[2] 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.

[3] Jin P, Zhuang X, Luo Y, et al. Exploring Index Structures for Zoned Namespaces SSDs[C]//2021 IEEE International Conference on Big Data (Big Data). IEEE, 2021: 5919-5922.

[4] Bjørling M, Aghayev A, Holmberg H, et al. {ZNS}: Avoiding the Block Interface Tax for Flash-based {SSDs}[C]//2021 US

ENIX Annual Technical Conference (USENIX ATC 21). 2021: 689-703.

[5] Purandare D R, Wilcox P, Litz H, et al. Append is Near: Log-based Data Management on ZNS SSDs[C]//12th Annual Conference on Innovative Data Systems Research (CIDR'22). 2022.