Practical Erase Suspension for Modern Low-latency SSDs

# 一 论文背景

与硬盘相比，基于NAND闪存的SSD具有更高的吞吐量和平均延迟，因此已成为存储设备的实际标准。随着NAND闪存技术的不断扩展，基于闪存的SSD已成为数据中心服务器中的关键组件。数据中心SSD的主要设计目标之一是降低读取尾部延迟，这对于交互式在线服务至关重要，因为单个查询可以生成数千个磁盘访问，单个长等待时间的磁盘访问会导致总体查询响应时间增加。为了实现这个目标，许多先前的工作集中在最小化垃圾收集对读取尾部延迟的影响上。[[[1]](#endnote-0)][[[2]](#endnote-1)][[[3]](#endnote-2)]这样的进步使得长读尾的其他的来源探究的很少，所以块擦除操作变得更加重要。

为了控制擦除对尾部等待时间的影响，提出了一种擦除暂停技术，当向同一个闪存芯片发出读取请求时，该技术暂停正在进行的擦除（并验证）脉冲。[[[4]](#endnote-3)]处理完读取请求后，擦除脉冲将从暂停的确切点恢复。但是，众所周知，NAND闪存业务对成本极为敏感，因此该技术可能会增加NAND外设的成本，以产生任意长度的擦除脉冲并跟踪每次擦除的确切状态。此外，这可能导致严重的NAND可靠性问题并导致写入不足。

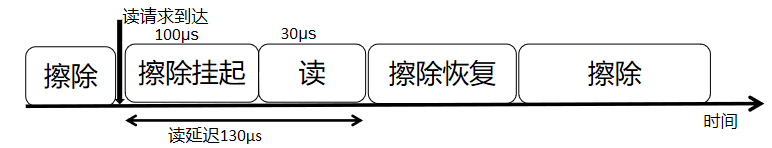


图1：擦除暂停的过程

为了解决这些限制，我们提出了适用于现代低延迟SSD的实用擦除暂停方案。我们的工作不是在任意点暂停/恢复擦除脉冲，而是着重于通过以下两种方法之一：在i）立即中止正在进行的擦除操作并从最后一个安全点恢复运行，在良好对齐的安全点处暂停/恢复擦除操作，或者ii）推迟暂停擦除操作，直到下一个安全点。利用它们之间的优缺点，我们还在两种机制之间引入了基于超时的切换机制，以动态地适应工作负载的变化。该方案使现代低延迟SSD可以在各种工作负载上提供极低的读取尾部延迟，而不会引起任何NAND可靠性问题或写入不足。

# 二 提出方法

## 1.立即擦除挂起

在擦除脉冲期间满足传入读取请求的一种方法是立即终止正在进行的擦除步骤并放弃进度。然后，在满足读取请求之后，擦除操作可以从当前擦除步骤的起点恢复。我们将此方案称为立即擦除暂停（I-ES），因为它立即取消了正在进行的擦除步骤。



图2：立即擦除暂停示意图

实际上，I-ES是原始擦除暂停的一种实用变体，但有以下两个更改。为了提高NAND的可靠性，利用增量步进脉冲擦除方案，在恢复暂停的擦除脉冲之前施加一个验证脉冲。由于NAND块中的每个NAND单元具有不同的擦除时序，因此某些NAND单元比其他NAND单元的擦除速度更快。因此，在恢复擦除脉冲时向所有NAND单元施加相同的脉冲会导致某些单元的过度擦除，这对已经擦除的单元施加了不必要的压力，并损害了NAND的可靠性。在恢复擦除脉冲之前，验证脉冲检测已擦除的单元，以不对NAND块中的这些单元施加不必要的压力。此外，为了使NAND外设的成本保持较低，在每次恢复时都将确定整个步进脉冲，而不是剩余的步进脉冲时间。为了产生任意长度的擦除脉冲，必须在NAND器件上放置具有细粒度控制的可变长度脉冲发生器。但是，当今的商品NAND不提供这种机制。相反，I-ES不需要进行此类更改即可使NAND的成本保持较低。的优点和缺点。由于擦除步骤会立即取消，因此始终可以确保读取命令具有最高优先级。读取命令没有经历由擦除操作引起的任何其他延迟，除了固定的等待时间（例如，约100μs）试图消除正在进行的擦除脉冲。

但是，该方案存在问题。对于连续的传入读请求，可以重复取消擦除步骤，从而阻止写请求（取决于擦除请求）完成。为了确认这种行为，我们修改了实际的生产级低延迟SSD的固件以实现此方案。一些写入操作长时间无法完成，因为先前的擦除操作被传入的读取请求连续取消。结果，它的尾部等待时间一直增加，直到工作负载结束。

## 2.延迟擦除挂起

在擦除脉冲期间为传入的读取请求提供服务的另一种方法是让读取请求等待，直到当前的擦除步骤完成为止。之后，代替进行下一个擦除步骤，服务读取请求。然后，擦除操作恢复并继续到下一擦除步骤。我们将其称为延迟擦除暂停（D-ES），因为它可以将擦除暂停推迟到当前擦除步骤结束。的优点和缺点。这种机制可以使擦除操作完成而不会引起写饥饿问题。通过保证一旦开始擦除步骤就执行单个擦除步骤，该机制保证了擦除操作的向前进行。尽管此机制不会导致擦除（和写入）饥饿，但它会通过使读取请求一直等到当前擦除步骤的结束（即我们的低延迟NAND中的1毫秒）来损害读取尾部延迟。由于D-ES为读取增加了额外的延迟，因此该方案还可能显示突发读取的较差的读取延迟。但是，可以通过在先提供突发读取后分批积压的擦除操作来避免此类情况。 T-ES解决了这个问题，它可以在I-ES和D-ES之间自适应切换。

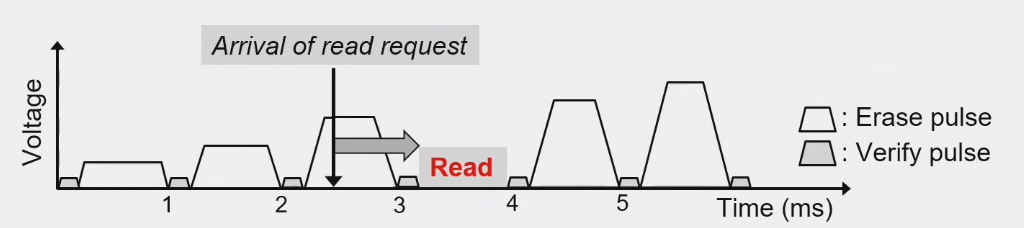


图3：延迟擦除暂停示意图

## 3.基于超时的擦除暂停

基于超时的擦除暂停策略。如果可以事先了解应用程序的请求模式，则当应用程序预期具有稀疏读取请求的阶段时，应该使用I-ES，因为在这种情况下，擦除不会发生饥饿（和写入）在即将到来的稀疏阅读期间可能会取得进展。另一方面，如果应用程序具有稳定的传入读取请求流，则用户应在这种情况下使用D-ES，采用I-ES会导致写尾延迟（即，写入不足）呈指数增长。

不幸的是，实际上，不进行性能分析就很难预测将来的I/O访问模式。因此，我们提出了一种基于超时的擦除暂停（T-ES）方案，该方案执行I-ES直到擦除操作被延迟Nms。如果擦除操作延迟了Nms（即发生超时），则此方案将切换到D-ES模式以避免潜在的擦除（和写入）不足。希望该方案找到一个执行擦除操作的周期，而不会被读取操作中断。

选择擦除超时延迟。 T-ES涵盖I-ES和D-ES之间的一系列策略，由N的值控制。如果将其设置为0，则等效于基本D-ES方案。相反，如果将其设置为无穷大，则等效于易于擦除（和写入）饥饿问题的I-ES方案。通常，选择较高的值会通过延迟擦除操作提供更多机会提供更小的读取尾部延迟，但是这种选择会导致最大写尾延迟增加很多。另一方面，选择较小的N使其表现得更像D-ES，从而提供较小的最大写尾延迟，但读请求等待擦除而产生延迟。 T-ES为用户提供了一个旋钮，使用户可以根据自己的意愿来考虑N值，以平衡最大写尾延迟，从而有可能改善读尾延迟。

# 三 验证与评估

## 1.测试方法

评估框架。尽管我们使用了带有修改后固件的真实，低延迟的原型SSD来执行一些实验，但无法利用真实的设备来评估我们提出的方案，例如D-ES和T -ES，因为此类策略的实现要求扩展SSD控制器和NAND闪存芯片之间的接口（例如，新命令）。因此，我们将MQSim用于我们的实验，并对其进行了扩展，以便可以准确地对低延迟NAND闪存芯片进行建模。特别是，我们（i）允许数据缓存管理器和FTL使用严格的8平面程序，以便可以实现更高的写入吞吐量，（ii）使I / O调度程序将用户读取请求作为最高优先级处理，并且（iii）修改NAND。闪存控制器和存储器逻辑可为我们的实用擦除暂停机制建模。表1总结了用于我们的MQSim的参数，通过仿真结果与实际低延迟SSD的仿真结果进行了比较，以验证仿真器。与真实设备的测量值相比，仿真器的平均读写延迟和吞吐量仅显示平均6％的错误（最坏情况下为13％）。

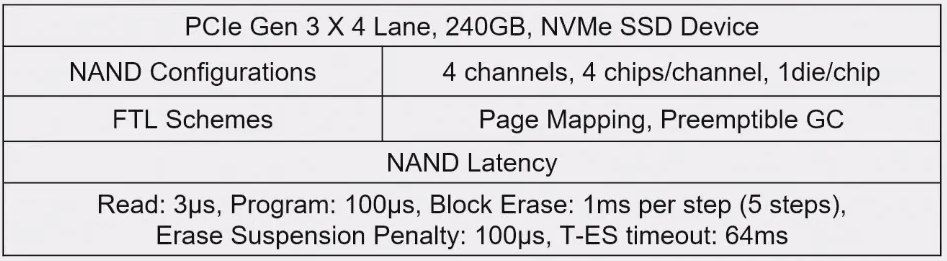


表1：MQSim的参数

评估配置。在本节中，我们将介绍不同基准测试以下配置的评估结果（即随机4KB访问，ACT和TPC-C）。此外，我们在所有实验中都使用稳态前提条件，其中SSD的空间（包括预留空间[OP]）已满；此前提条件有助于评估在最坏情况下的读写请求条件下可能发生的延迟行为。第2节讨论了三种实用的擦除暂停机制：I-ES，D-ES和T-ES。我们将以下三种配置添加到我们的评估中以进行比较：

•基准：传入的读取请求不会抢占擦除操作。

•擦除暂停（ES）：该方案可以从任意点暂停和恢复擦除脉冲。

•理想擦除暂停（Ideal-ES）：此方案可以从任何任意点暂停并恢复擦除操作，擦除暂停惩罚为零。

## 随机访问基准

工作量。我们首先使用微基准评估擦除挂起策略，该基准在队列深度16处产生4KB随机读取（70％）和写入（30％）的混合。我们利用Flexible I / O Tester（FIO）生成此类磁盘访问模式。该工作负载被广泛用于评估SSD的延迟性能。图显示了该实验的结果。

阅读尾部延迟。在基线中，当发生擦除时，后续的读取请求仅会延迟剩余的擦除时间（最长5ms）。结果，基线策略显示大约5毫秒的读取尾部延迟。对于ES和I-ES，此类擦除操作会因传入的读取请求而反复中止。结果，除了正在进行的擦除操作外，传入的读取请求不会有额外的延迟，除了擦除暂停/恢复所需的100µs延迟。因此，ES和I-ES策略都具有非常低的读取尾部延迟。请注意，如第2节所述，它们很容易出现擦除（和写入）饥饿现象。但是，此处不会发生这种情况，因为此工作负载具有一段时间，一段时间内未发出读取请求（即，队列中的所有16个未完成请求都是等待擦除进行的写入请求）。在此期间，擦除操作成功完成。 D-ES的读取尾部延迟大约为1ms，因为它总是使正在进行的擦除步骤（而不是整个擦除）完成，从而导致读取请求等待。在此工作负载中，T-ES的行为类似于I-ES，因为很少触发超时。

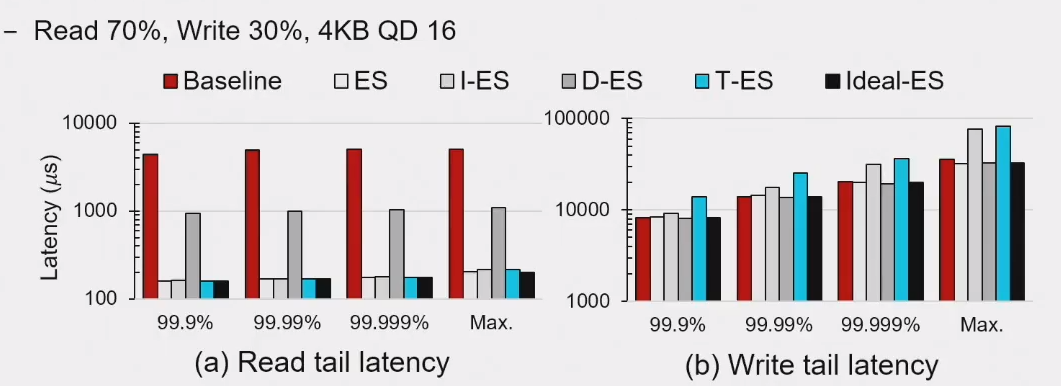


图4：随机访问基准下各模式读写尾延迟

写尾延迟。基线，ES，D-ES和Ideal-ES具有较低的写尾延迟，因为它们的擦除操作在相对较短的时间内完成，而不会取消现有的擦除脉冲。值得注意的是，基准策略的写尾延迟是没有擦除暂停的模型的最大GC延迟（35毫秒）。另一方面，I-ES和T-ES都多次中止擦除操作，因此大大延迟了写入操作。结果，它们倾向于具有更长的写尾延迟。

## 数据库基准

工作量。 ACT为Aerospike数据库服务器的实时I / O访问模式建模。本质上，ACT由三个线程组成：一个线程每秒发出2K小（1.5KB）读取请求，另一个线程每秒发出24个大（128KB）读取请求，第三个线程每秒发出24个大（128KB）写入请求。 ACT逐渐以整数倍提高此速率（例如，ACT 4倍的工作量意味着每秒8K个小读取请求，96个大读取请求，96个大写入请求），并认为该设备已通过性能测试（如果满足）在长时间（例如24小时）内满足以下条件：i）95％的事务在1毫秒内完成，ii）99％的事务8ms内完成，iii）99.9％的事务在64ms内完成，并且iv）每种请求的平均事务时间小于ACT工作负载的I / O请求时间。如果某个设备仅满足第四个条件，但不满足前三个条件之一，则称该设备通过了压力测试，但未通过性能测试。 SSD可以满足的最大ACT乘数是该特定SSD的性能等级。

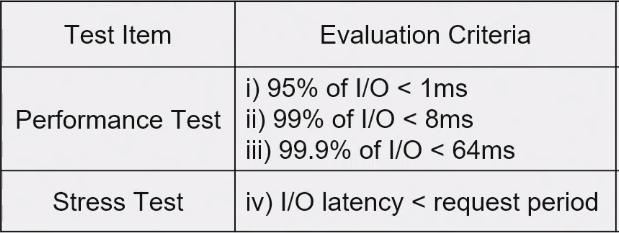


表2：压力测试和性能测试要求

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **基线** | **ES** | **I-ES** | **D-ES** | **T-ES** | **Ideal-ES** |
| 压力 | 32× | 22× | 22× | 30× | 30× | 32× |
| 性能 | 14× | 22× | 22× | 30× | 30× | 32× |

表3：使用ACT各模式的压力和性能测试结果

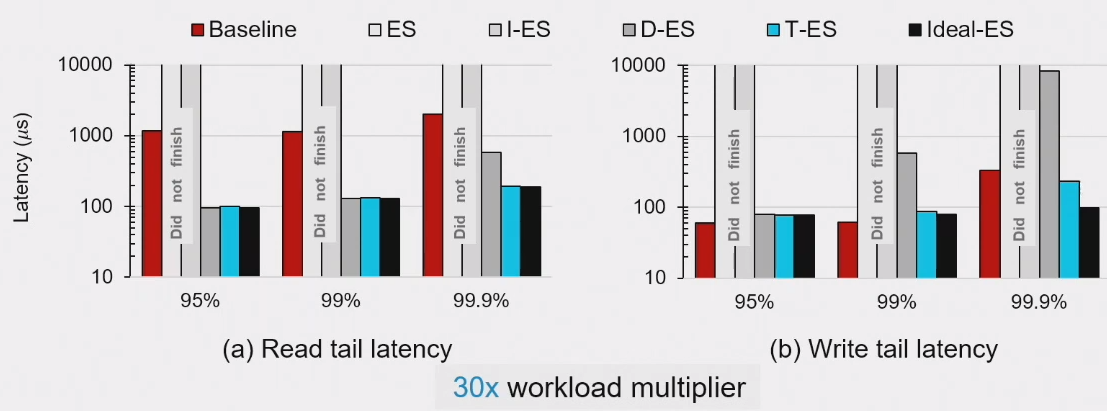


图5：数据库基准下各模式读写尾延迟

ACT结果。表3显示了每个配置通过性能和压力测试的最大乘数。图显示在30倍的工作负载倍增器下，访问的尾部延迟为95％，99％和99.9％。如表3所示，基线具有良好的平均响应时间，因此可以使用32倍乘数成功运行ACT。另一方面，它的读取尾部延迟最差，因为部分擦除延迟（最长5毫秒）暴露于读取请求。这导致基线的性能测试结果相对较差。 ES和I-ES具有良好的读取尾部延迟行为。但是，连续的读取请求会导致擦除（和写入）饥饿，这会导致在值大于22倍的工作负载乘数上的压力测试（第四种情况）失败。另一方面，D-ES和T-ES都在压力和性能测试中均表现出了出色的结果。 D-ES和T-ES都可以保持较低的读取尾部等待时间，同时避免擦除（和写入）不足。

## 3.4Transaction Processing Benchmark

TPC-C是在线交易处理框架的流行基准。我们利用已发布的SNIA 中运行TPC-C的系统的磁盘跟踪信息，来评估我们的擦除暂停机制。

TPC-C结果。图6显示了该实验的结果。基线策略通常会比其他方案观察到明显更高的读取尾部延迟，因为读请求在擦除操作之后，因此有延迟。 ES或I-ES都无法正常运行，因为它们遭受严重的擦除（和写入）饥饿，并在长时间不能满足许多写请求导致模拟器过早退出。与ACT工作负载一样，这是因为连续的读取请求阻止了擦除操作的完成。T-ES和D-ES均比基线降低了尾部等待时间。特别是，两者均达到约1毫秒的最大尾部延迟，这表明读取请求最多仅有约一个擦除步长的延迟。

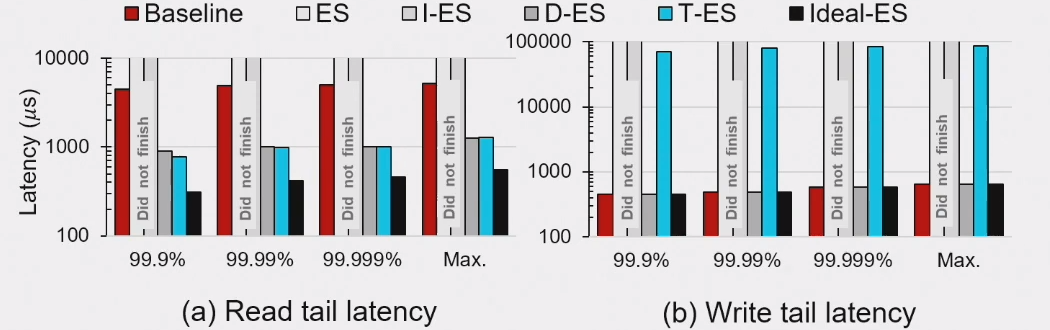


图6：Transaction Processing Benchmark下各模式读写尾延迟

基线，D-ES和Ideal-ES的写尾延迟彼此相似。相反，ES和I-ES遭受写饥饿，并且T-ES记录了最长的写尾延迟。 T-ES方案将擦除操作（和后续写入操作）延迟到超时值，直至达到一个超时值，以期找到一个可以执行擦除操作而不会阻塞读取的周期。在这种情况下，T-ES找不到这样的时间段，因此在延迟写请求后最终触发了D-ES机制。如预期的那样，最大写入延迟会在90毫秒内收敛（即，GC延迟（24毫秒）和T-ES超时值（64毫秒）之和）。

## 3.5对T-ES超时阈值的敏感性（N）

如果擦除操作延迟了Nms，则T-ES从I-ES切换到D-ES，以避免擦除（从而避免写入）不足。我们使用N可变的TPC-C进行敏感性研究，以了解选择此参数时的大小。

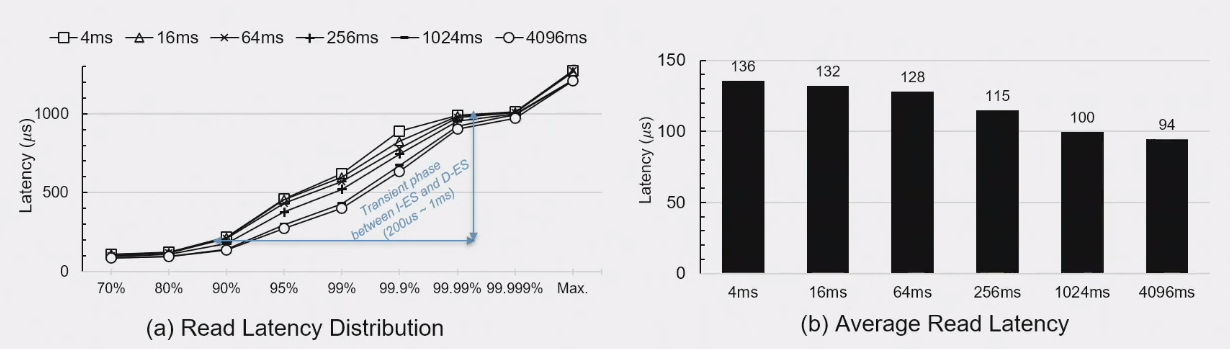


图7：TPC-C下超时限读延迟和平均延迟

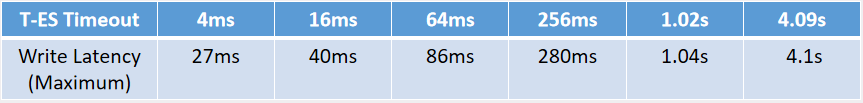


表4：TPC-C下最大写延迟

图7（a）显示了不同的N值（从4ms到4096ms）的读取等待时间分布。我们开始观察到从更多的I-ES（大约200µs）到D-ES（大约1ms）的逐渐过渡。通常，增加N i）降低高延迟读取的频率（即超过200µs），并且ii）平均读取延迟，但是iii）增加最大写入延迟。随着N的增加，T-ES更有可能在I-ES模式下运行，以降低读取请求经历1毫秒延迟以完成正在进行的擦除脉冲的机会。如图7（b）所示，较少的长等待时间读取导致较低的平均读取延时。

但是，增加N会对最大写入延迟产生负面影响。表3总结了变化N时的最大写延迟。如第2节所述，T-ES的最大写延迟是GC延迟和T-ES超时值的总和。这是因为在I-ES模式下运行时，GC操作（需要擦除操作以生成用于用户数据写入的空闲块）可能会受到干扰。触发T-ES超时后，它将切换到D-ES，以允许GC操作产生空闲块来写入用户数据。 TPC-C工作负载（具有稳态前提条件）的最大GC延迟为24ms。因此，测得的最大写等待时间与估计值相差不远（即，GC等待时间加N）。

# 四 总结

## 1.总结

·是第一个在现有的擦除挂起方案中识别NAND可靠性问题并出现饥饿现象的公司，并在实际的生产级SSD上演示了后者。

·提出了两种实用的擦除暂停机制，即立即擦除暂停和递延擦除暂停。我们还分析了这两种机制之间的权衡，并在两者之间引入了基于超时的切换策略，以便在工作负载变化时充分利用两者。

·通过提议的擦除暂停机制在包括Aerospike认证工具（ACT）和TPC-C基准测试工作负载在内的各种工作负载下，显着降低了读取尾部延迟。

## 可能需要的优化

· I-ES和T-ES在完成读请求后恢复脉冲对块擦除的重复擦除现在不可避免，对硬盘的损伤同样存在。

·可以进行更多试验测试对不同环境和负载的条件下I-ES的适宜值进行探索。

1. [1] Wonil Choi, Myoungsoo Jung, Mahmut Kandemir, and Chita Das. Parallelizing garbage collection with I/O to improve flash resource utilization. In Proceedings of the 27th International Symposium on High-Performance Parallel and Distributed Computing, HPDC ’18, pages 243–254. ACM, 2018. [↑](#endnote-ref-0)
2. [2] Wonkyung Kang, Dongkun Shin, and Sungjoo Yoo. Reinforcement learning-assisted garbage collection to mitigate long-tail latency in SSD. ACM Transactions on Embedded Computing Systems, 16(5s):ACM, 134:1– 134:20, 2017. [↑](#endnote-ref-1)
3. [3]Shiqin Yan, Huaicheng Li, Mingzhe Hao, Michael Hao Tong, Swaminatahan Sundararaman, Andrew A. Chien, and Haryadi S. Gunawi. Tiny-tail flash: Near-perfect elimination of garbage collection tail latencies in NAND

   SSDs. In Proceedings of the 15th USENIX Conference on File and Storage Technologies, FAST’17, pages 15– 28. USENIX Association, 2017. [↑](#endnote-ref-2)
4. [4]Guanying Wu and Xubin He. Reducing SSD read latency via NAND flash program and erase suspension. In Proceedings of the 10th USENIX Conference on File and Storage Technologies, FAST’12, pages 117–123. USENIX Association, 2012. [↑](#endnote-ref-3)