

写入模式感知与热度分级的固态硬盘垃圾回收方法^{*}

王婷煜¹, 杨孟飞², 刘洪标³, 姜菁菁³, 李少峰¹, 刘 波³

¹(西安电子科技大学 计算机科学与技术学院, 陕西 西安 710071)

²(中国空间技术研究院, 北京 100094)

³(北京控制工程研究所, 北京 100190)

通信作者: 杨孟飞, E-mail: yangmf@bice.org.cn



摘 要: 固态硬盘 (solid-state drive, SSD) 凭借其高速读写性能与低功耗等特点, 已广泛应用于嵌入式设备和数据中心等多种场景. 其中, 垃圾回收 (garbage collection, GC) 机制对空间回收效率与设备寿命具有重要影响. 然而, 现有 GC 方法在触发灵活性、数据迁移效率及磨损均衡性方面仍存在不足. 为提升 GC 机制在不同负载下的适应性与可靠性, 提出一种写入模式感知与热度分级的垃圾回收 (write pattern-aware and heat-tiered garbage collection, WHGC) 方法. 该方法首先构建写入模式感知指标, 设计多阈值触发机制, 实现 GC 在系统空闲与忙碌状态下的动态调度; 然后, 提出一种融合空间使用状态与磨损程度的块评分策略, 用于指导回收块选择; 最后, 在有效页迁移阶段, 采用指数衰减模型识别逻辑页热度, 并结合块擦除次数的等级划分, 构建热度与磨损的映射关系, 引导有效页迁移至对应等级的目标块, 从而实现数据分布优化与块间磨损均衡. 基于 SSDsim 平台的实验结果表明, WHGC 在所有数据集下均实现了相较其他典型 GC 方法更长的设备寿命, 并在写放大系数、有效页迁移量与擦除次数标准差等指标上保持了较优的综合性能.

关键词: 固态硬盘; 垃圾回收; 动态阈值; 数据热度识别; 磨损均衡

中图法分类号: TP306

中文引用格式: 王婷煜, 杨孟飞, 刘洪标, 姜菁菁, 李少峰, 刘波. 写入模式感知与热度分级的固态硬盘垃圾回收方法. 软件学报. <http://www.jos.org.cn/1000-9825/7679.htm>

英文引用格式: Wang TY, Yang MF, Liu HB, Jiang JJ, Li SF, Liu B. Write Pattern-aware and Heat-tiered Garbage Collection Method for SSDs. Ruan Jian Xue Bao/Journal of Software (in Chinese). <http://www.jos.org.cn/1000-9825/7679.htm>

Write Pattern-aware and Heat-tiered Garbage Collection Method for SSDs

WANG Ting-Yu¹, YANG Meng-Fei², LIU Hong-Biao³, JIANG Jing-Jing³, LI Shao-Feng¹, LIU Bo³

¹(School of Computer Science and Technology, Xidian University, Xi'an 710071, China)

²(China Academy of Space Technology, Beijing 100094, China)

³(Beijing Institute of Control Engineering, Beijing 100190, China)

Abstract: Solid-state drives (SSDs), owing to their high read/write performance and low power consumption, have been widely used in diverse scenarios such as embedded systems and data centers. Among them, the garbage collection (GC) mechanism plays a critical role in space reclamation efficiency and device lifespan. However, existing GC methods still suffer from limitations in triggering flexibility, data migration efficiency, and wear leveling effectiveness. To enhance the adaptability and reliability of GC under varying workloads, this study proposes a write pattern-aware and heat-tiered garbage collection (WHGC) method. WHGC first constructs a write pattern-aware indicator and designs a multi-threshold triggering mechanism to enable dynamic GC scheduling under both idle and busy system states. It then introduces a block scoring strategy that integrates space utilization and wear level to guide victim block selection. During the valid page migration phase, WHGC employs an exponential decay model to identify the heat of logical pages, and combines it with block erase count

* 基金项目: 国家自然科学基金 (62192730)

收稿时间: 2025-08-19; 修改时间: 2026-02-26; 采用时间: 2026-03-24; jos 在线出版时间: 2026-07-15

classification to establish a mapping between heat levels and wear levels, thus guiding valid pages to migrate to blocks of corresponding levels. Experimental results based on the SSDsim platform demonstrate that WHGC achieves a longer device lifespan than other typical GC methods across all datasets, while maintaining superior overall performance in terms of write amplification factor, the number of migrated valid pages, and the standard deviation of erase counts.

Key words: solid-state drive (SSD); garbage collection; dynamic threshold; data heat recognition; wear leveling

固态硬盘 (solid-state drive, SSD) 凭借高速读写性能、抗震耐用、低功耗等优势, 已广泛应用于嵌入式设备、数据中心及边缘计算等多种场景^[1]. 然而, 其核心存储介质 NAND 闪存存在读写与擦除粒度不一致、擦除次数有限等固有特性^[2], 限制了 SSD 的整体性能与使用寿命. 为避免每次数据更新均伴随高成本的擦除操作, SSD 通常采用异地更新机制^[3], 将新数据写入空闲位置, 并将旧数据标记为无效, 导致无效页持续累积. 为维持系统写入性能, 控制器需依赖垃圾回收 (garbage collection, GC) 机制及时回收无效页并释放空间. 作为 SSD 控制器的核心功能之一, GC 策略设计直接影响系统响应延迟、写放大效应与块间磨损均衡水平, 是当前 SSD 架构优化的重要研究方向^[4].

在过去的研究中, 学者们提出了多种 GC 方法以提升空间回收效率与磨损控制能力. 早期方法如 Greedy^[5]通过优先选择无效页最多的块进行擦除, 以提升空间回收效率; CB^[6]结合空间利用率与页失效时间等信息进行回收块选择; CAT^[7]进一步考虑块的擦除次数, 同时实现冷热数据的分类迁移, 兼顾磨损均衡; FaGC^[8]根据逻辑页的更新频率划分数据热度, 并在有效页迁移过程中以此引导目标块选择, 从而实现冷热数据的分类管理. 近年来, 为进一步优化 GC 性能与系统寿命, 研究者在方法设计中引入了数据热度感知、动态调度与主机协同调控等机制. TPGC^[9]利用写入序号与更新时间区分数据热度, 实现数据分类迁移; DA-GC^[10]采用动态阈值与回收块评分机制, 综合空间利用与擦除状态调度 GC; USIGC^[11]引入可中断 GC 与空闲预测机制, 实现 GC 与主机 I/O 请求的协同调度; Lifespan-based GC^[12]通过系统运行前的访问模式分析, 预测数据生命周期以优化 GC 策略.

然而, 随着主机 I/O 负载日益多样化、写入请求频繁波动以及对设备寿命要求不断提高, 现有方法虽在提升 GC 效率与延长设备寿命方面取得了一定进展, 但仍面临以下挑战: (1) 缺乏对系统 I/O 负载状态的动态感知机制, 无法根据主机写入行为动态调整 GC 触发阈值, 易导致高负载下资源抢占, 低负载下空间浪费, 降低 GC 效率; (2) 数据热度识别依赖时间戳或更新次数, 受时间精度与历史记录影响, 容易导致冷热数据误判, 导致不必要的数据迁移; (3) 有效页迁移策略在数据热度与物理块磨损程度之间的匹配上仍显粗糙, 可能出现热数据误迁至高磨损块或冷数据写入低磨损块的情况, 从而加剧磨损不均, 缩短 SSD 使用寿命.

针对上述问题, 本文提出一种写入模式感知与热度分级的垃圾回收方法 WHGC (write pattern-aware and heat-tiered garbage collection), 以实现 GC 性能与设备寿命的协同优化. 该方法的核心机制包括: 基于主机写入行为动态感知系统负载, 自适应调整 GC 触发阈值, 实现灵活的 GC 调度策略; 结合空间利用率与块擦除状态构建评分模型, 以指导回收块选择; 构建逻辑页热度等级与块擦除等级的映射机制, 实现热度感知的数据迁移与磨损均衡调控.

综上, 本文的主要贡献如下.

(1) 提出一种写入模式感知的 GC 触发机制, 结合主机写入行为动态调整触发阈值, 提升系统对负载状态变化的自适应能力.

(2) 提出一种基于逻辑页热度等级与块擦除等级的映射迁移机制, 实现逻辑页活跃度与物理块磨损状态之间的协调匹配, 提升数据迁移的合理性.

(3) 在多个典型工作负载下, 对所提方法与主流 GC 策略进行对比评估, 实验结果验证了 WHGC 在写放大控制、有效页迁移开销与延长设备使用寿命能力等方面具有综合优势.

本文第 1 节介绍垃圾回收的研究背景与相关工作. 第 2 节介绍 WHGC 方法的设计原理与实现机制. 第 3 节通过对比实验验证本文所提方法的有效性. 第 4 节总结全文.

1 相关工作

1.1 SSD 结构

SSD 主要由主机接口、控制器和 NAND 闪存芯片组成. 图 1 展示了典型的基于 NAND 闪存的 SSD 内部结

构^[13]. 其中, 主机接口负责 SSD 与主机系统之间的通信; SSD 控制器负责接收主机读写请求, 协调各通道的数据传输与管理操作; 闪存控制器负责将逻辑命令转换为底层 NAND 操作, 其连接多个通道 (channel), 每个通道可并行访问多个闪存芯片 (chip); 每个芯片内部包含多个晶圆 (die), 每个晶圆划分为多个分组 (plane), 分组中包含多个块 (block), 每个块由多个页 (page) 组成. 其中, 块是闪存擦除的最小单位, 页是读写的最小单位^[14]. 在 SSD 控制器中, 垃圾回收是闪存转换层 (flash translation layer, FTL) 的关键模块之一, 用于回收无效页并释放可用块空间, 从而提高空间利用率, 保障系统写入性能^[15].

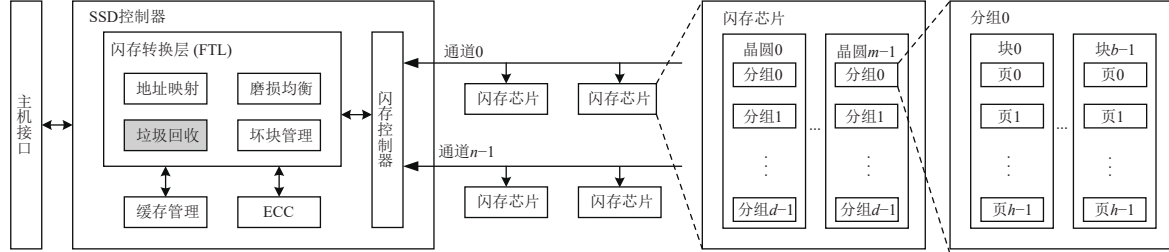


图1 SSD 结构图

1.2 垃圾回收方法研究现状

一个完整的 GC 方法通常包含 4 个关键步骤: (1) 触发 GC 操作; (2) 选择回收块; (3) 迁移有效页; (4) 擦除回收块. 围绕这些核心模块, 研究者主要从 GC 触发时机、回收块选择策略、有效页冷热识别方法以及迁移策略等方面开展了深入研究.

GC 触发时机的设计决定了回收操作介入的时机与频率, 进而影响系统延迟、写放大及空闲空间利用效率. 现有多数方法采用固定阈值策略作为 GC 触发条件^[5-7,16-21], 即当 SSD 整体或局部单元 (如通道^[11]和分组^[10,22]) 中的空闲页比例低于预设值时立即执行 GC 操作. 为提升 GC 触发机制的灵活性, 一些研究引入了动态触发策略^[8,10,22], 通过结合系统空间使用状态实时调整触发阈值. 部分方法进一步引入系统运行状态感知机制^[10,11,22], 根据 SSD 当前处于空闲或忙碌阶段, 分别执行可中断 GC (interruptible GC, IGC) 或不可中断 GC (non-interruptible GC, NGC), 以降低 GC 对主机 I/O 请求的干扰. 还有部分研究从 GC 的执行方式出发, 优化其调度过程以进一步提升系统性能. Lee 等人^[23]提出的 semi-preemptive GC 机制支持在 GC 执行期间根据主机 I/O 请求状态动态暂停回收操作, 从而优先处理主机请求, 避免 GC 阻塞造成的响应延迟. Shahidi 等人^[20]将传统串行 GC 操作转化为多分组并行调度方式, 在多个通道中同时进行回收任务, 提升空间回收效率与执行并发度. 因此, 现有 GC 触发机制虽然在固定阈值、动态阈值以及可中断调度等方面不断改进, 但大多数方法仍主要依据空间使用状态触发 GC, 对主机写入负载变化的感知与适应能力仍相对有限.

在 GC 过程中, 回收块的选择策略会影响系统的空间回收效率与磨损均衡效果. Wu 等人^[5]提出了经典的贪婪算法 Greedy, 每次选择无效页最多的物理块作为回收对象. van Houdt^[16]提出了从全局块集合中随机选出 d 个候选块, 再从中选择有效页最少的块作为回收块. Kawaguchi 等人^[6]提出的 CB 算法在贪婪算法基础上引入时间维度, 在选择回收块时综合考虑无效页比例和页失效年龄. 为了兼顾空间利用率与设备耐久性, Chiang 等人^[7]提出了 CAT 算法, 在 CB 算法的基础上增加考虑了物理块的擦除次数. Yan 等人^[8]设计了一种可切换的回收策略, 常规情况下选择无效页最多的块, 但若系统出现磨损不均现象, 则转而选择擦除次数最小的块. Yan 等人^[19]综合考虑无效页数量与页写入序号 (page sequence number, PSN) 选择回收块, 并在擦除次数差异超过阈值时调整为最小擦除块优先. Yu 等人^[9]同时考虑块的有效页比例和擦除次数, 以及无效页的失效时间间隔进行回收块的选择. Lee 等人^[24]融合擦除次数、擦除时间间隔与迁移代价等多个因素构建评分模型, 以平衡回收带来的性能与寿命开销. Sun 等人^[25]提出了 Light-GC, 通过预测不同块未来的无效页增长趋势, 预估回收影响以提高回收效率. Cheng 等人^[12]提出了 Lifespan-based GC, 通过估算块的“预期寿命”来决定其回收优先级, 以实现更长的设备寿命与更少的数据迁移开

销. Salkhordeh 等人^[26]从算法效率角度出发, 提出了常数时间代价-效益策略, 通过构建按有效页数划分的多链表结构, 实现回收块选择的常数时间复杂度, 从而降低内存负担, 提升系统可扩展性.

在确定回收块后, 若该块中存在有效页, 则需将其迁移至其他物理块以释放空间. 受主机访问行为影响, 数据在更新频率上表现出明显差异: 更新频繁者被称为热数据, 而长期未被更新的数据则被视为冷数据. 准确识别并分类迁移冷热数据, 有助于提升垃圾回收的空间回收效率, 降低擦除操作带来的磨损开销. 部分方法未区分冷热数据^[10,11,22], 将有效页迁移至同一目标块, 可能导致冷热数据混合分布. 为此, 已有研究基于数据的更新行为识别其热度特征. CAT^[7]基于页的失效时间判断其活跃度, 将冷热数据分别迁移至不同目标块. Tjioe 等人^[18]提出了 WECCO, 利用逻辑页的写入次数与时间戳构建热度指标, 若某页热度超过全局平均值, 则被视为热数据. Kwon 等人^[17]提出了 FeGC, 根据页的更新间隔是否大于平均值来划分热度类别. Lin 等人^[27]提出了 Dual Greedy, 在写入时根据旧页的写请求次数判断热度, 将频繁更新的页作为热数据, 并进行分类存储. FaGC^[8]根据逻辑页的更新间隔时间区分冷热数据, FaGC+^[19]则通过页写入序号 PSN 反映逻辑页的更新活跃程度, PSN 越小说明更新越频繁. Hwang 等人^[28]提出了 MCSGC, 综合考虑逻辑页当前时间与初始写入时间之间的间隔来区分热度. Thermo-GC^[21]采用页的迁移次数判断其活跃等级, 区分初次迁移页与多次迁移页, 并利用块级信息反映页热度. TPGC^[9]融合逻辑页的写入时间戳与写入序号 PSN, 实现热度感知与识别.

对于区分数据热度的 GC 方法, 研究者提出了多种有效页迁移策略. WECCO^[18]将热数据迁移至擦除次数较低的块, 冷数据迁移至擦除次数较高的块, 实现冷热数据的分类存储. Dual Greedy^[27]在新页写入阶段根据数据热度进行初始分类, 将热数据与冷数据分别写入专用的热页块和冷页块. MCSGC^[28]将不同热度的数据迁移至对应热度活跃块, 同时对于新写入数据统一分配擦除次数最小的块. Thermo-GC^[21]在迁移后标记页热度信息, 并在下一轮 GC 中避免将其再次迁移至相同位置, 降低重复迁移带来的写放大开销. FeGC^[17]与 FaGC^[8]均采用基于擦除次数的迁移策略, 分别将热数据迁移至擦除次数最小的空闲块、冷数据迁移至擦除次数最大的空闲块, 提升磨损均衡性. FaGC+^[19]与 TPGC^[9]则进一步将逻辑页热度划分为 4 个等级, 将冷数据迁移至擦除次数最大的空闲块, 次冷数据迁移至擦除次数第 2 大的空闲块, 热数据迁移至擦除次数最小的空闲块, 次热数据至擦除次数第 2 小的空闲块, 实现更细粒度的迁移策略. 现有冷热数据识别与迁移方法已经从简单冷热二分类逐渐发展为多等级热度划分策略, 但多数方法仍采用较为固定的冷热数据识别策略, 在控制迁移开销与提升磨损均衡之间仍存在优化空间.

此外, 也有研究将垃圾回收机制与其他系统模块协同设计, 以实现更加高效智能的 SSD 整体管理方案. Wu 等人^[29]提出了 CAGC, 将内容去重操作嵌入至 GC 过程, 缓解写入路径延迟并提升页迁移效率. Wang 等人^[30]提出了 CDA-GC, 通过缓存调度优化 GC 效率, 在高负载环境下显著提升性能表现. Wang 等人^[31]提出了 SepBIT, 一种基于块失效时间推断的数据分离策略, 将生命周期相近的数据块聚集到同一组, 减少因 GC 引发的有效页迁移次数, 从而降低日志结构化存储系统中的写放大. Kim 等人^[32]提出了 D2FS 架构, 将 GC 控制权交由设备端执行, 实现文件系统与 FTL 之间的协同调度, 避免文件系统与设备层之间的 GC 干扰问题. Lange 等人^[33]提出了在线算法 Gladiator, 利用机器学习预测逻辑页的生命周期, 将请求按热度进行分级存储, 并结合误差感知清理、鲁棒清理与指数提升等机制, 有效降低 SSD 的写放大. Zhou 等人^[34]提出了一种面向深度学习大数据场景的 GC 优化方法, 通过在自主设计的 SSD 架构中集成冷热数据分离与解耦式 GC 机制, 显著降低了 GC 带来的延迟与系统资源占用. Yang 等人^[35]提出了一种基于未来访问温度预测的 GC 优化方法, 通过训练神经网络模型预测有效页未来的访问频率, 以实现数据的冷热分类. Liu 等人^[36]提出了一种 GC 辅助的数据压缩机制, 通过感知工作负载访问特征, 在 GC 过程中动态调整有效页的压缩比. Luo 等人^[37]提出了一种基于回声状态网络的自学习热数据预测方法, 通过预测未来热点数据, 优化 SSD 的垃圾回收与磨损均衡过程. Sun 等人^[38]利用设备端部署的轻量机器学习模型, 对写入页的寿命进行实时预测, 以降低写放大并提升数据分布效率. 上述研究从系统架构、缓存管理、数据布局、智能预测等维度拓展了 GC 机制的设计空间, 为构建高效、智能、协同的存储系统提供了重要思路.

综上, 尽管现有 GC 方法已取得了显著进展, 但仍存在以下不足.

(1) 多数 GC 触发机制主要依赖于空间使用状态, 如空闲页比例或空间利用率等指标, 缺乏对主机 I/O 请求负载变化的感知与适配能力. 当系统写入负载发生变化时, 此类触发机制难以及时调整 GC 触发频率, 从而可能在高

负载情况下频繁执行 GC 并增加迁移开销, 或在低负载阶段延迟空间回收。

(2) 数据热度识别方法多依赖时间间隔、更新时间戳或写入次数等指标, 虽然能够在一定程度上反映逻辑页的更新行为, 但对历史访问特征与近期写入变化的综合评估仍不充分, 在准确反映数据写入活跃度方面仍存在优化空间。

(3) 有效页迁移策略普遍将冷热数据与高磨损块绑定, 缺乏热度与磨损等级之间更灵活的匹配机制。许多方法通常将热数据迁移至擦除次数较低的块、冷数据迁移至擦除次数较高的块, 虽然有助于实现磨损均衡, 但在数据热度差异较大的情况下仍可能引入额外的迁移开销。

此外, 近年来也有研究尝试将 GC 机制与文件系统、缓存管理、压缩策略或机器学习预测模型等系统模块协同设计, 进一步拓展了 GC 的设计空间, 体现出跨层优化与智能调度的研究趋势。但这类方法多聚焦于系统整体的集成性能, 缺乏对 GC 机制本身的关键策略进行深入优化。因此, 有必要在综合考虑写入负载特征、逻辑页热度以及物理块磨损状态的基础上, 对垃圾回收机制的触发策略与数据迁移策略进行系统改进, 以在延长 SSD 寿命的同时有效控制 GC 迁移开销。

1.3 问题分析与研究动机

随着 SSD 容量规模不断增长以及数据密集型应用的持续发展, 存储系统在长期运行过程中面临更高的写入压力与更加复杂的访问模式。同时, 在航天等高可靠嵌入式应用场景中, 存储设备不仅需要具备较高的性能, 还需要在长期运行条件下保持良好的寿命特性和运行稳定性。在此背景下, 垃圾回收机制对系统写放大、设备寿命以及整体运行效率的影响日益显著。因此, 如何在延长 SSD 寿命的同时有效控制 GC 迁移开销, 成为垃圾回收机制设计中的关键问题。

尽管现有垃圾回收方法在 GC 触发机制、数据热度识别以及迁移策略等方面进行了大量研究, 但不同优化策略在系统寿命延长能力与迁移开销控制之间仍可能存在明显的性能权衡。因此, 本文选取两类具有代表性的 GC 策略进行对比分析: 一类是以 Greedy 算法为代表的低迁移开销策略, 其主要通过优先回收无效页比例较高的物理块来减少有效页迁移数量; 另一类是以 TPGC 为代表的热度感知策略, 通过识别逻辑页更新频率并进行分类迁移, 从而提升系统磨损均衡能力。

为揭示不同 GC 策略在系统寿命与迁移开销方面的性能差异, 本文选取写请求比例较高的 prxy_0 数据集作为代表性负载, 对优先回收无效页最多块的 Greedy 方法^[5]和具备冷热数据分类迁移能力的 TPGC 方法^[9]进行对比实验分析。Greedy 代表典型的低迁移开销 GC 策略, 而 TPGC 代表引入冷热数据识别机制的 GC 策略, 两者在设计思想上具有较强代表性。实验以系统首个闪存块达到擦除上限作为运行终止条件, 以保证不同 GC 方法在寿命评估上的可比性。关于该负载的具体特征及实验配置将在第 3 节实验部分进行详细说明。

如图 2 所示, 图 2(a) 展示了不同 GC 方法在系统运行过程中的写放大因子变化情况, 图 2(b) 展示了 GC 过程中迁移有效页数量随系统运行进程的变化情况。其中横坐标表示主机写入次数, 用于反映系统运行进程; 图 2(a) 的纵坐标为写放大系数, 图 2(b) 的纵坐标为 GC 迁移的有效页数量。可以观察到, Greedy 方法在整个运行过程中始终保持较低的写放大水平, 表明其在垃圾回收过程中迁移的有效页数量相对较少。然而, 由于该方法仅依据无效页比例选择回收块, 未显式考虑数据热度分布以及块间磨损均衡, 导致系统更早出现达到擦除上限的闪存块, 从而限制系统整体可承载的主机写入量。相比之下, TPGC 通过引入数据热度识别机制, 将不同更新频率的数据进行分类迁移, 从而在一定程度上延缓高磨损块的出现并提升系统寿命。但该方法需要频繁执行冷热数据分类与迁移, 导致 GC 过程中迁移的有效页数量显著增加, 从而使系统写放大因子长期处于较高水平。

上述实验结果表明, 现有 GC 方法在系统寿命延长能力与迁移开销控制之间仍存在明显的性能权衡: 以 Greedy 为代表的低迁移策略能够有效降低写放大, 但系统更早出现达到擦除上限的闪存块, 从而限制其可承载的主机写入量; 而以 TPGC 为代表的热度感知策略虽然能够在一定程度上延缓高磨损块的出现并提升系统寿命, 却往往需要迁移大量有效页, 导致系统写放大水平显著增加, 从而引入额外的写入开销。

针对上述问题, 本文提出一种写入模式感知与热度分级的垃圾回收方法 WHGC。该方法从 GC 触发策略和有

(1) 基于写入模式感知的 GC 触发机制. 现有垃圾回收方法多基于固定空闲空间阈值或空间利用率触发 GC 操作, 缺乏对主机写入负载变化的动态感知能力. 当系统写入模式发生变化时, 固定阈值策略难以及时调整 GC 触发时机, 可能导致 GC 过早或滞后执行. 针对这一问题, WHGC 通过实时监测主机写入行为特征, 引入写入模式指标动态调节 GC 触发阈值, 使垃圾回收能够根据系统负载状态自适应调整触发时机, 从而降低 GC 对主机 I/O 性能的影响, 并提升空间回收的灵活性.

(2) 基于指数衰减模型的逻辑页热度识别方法. 现有热度识别方法多依赖时间戳、访问间隔或写入次数等指标, 通常采用线性度量方式, 难以同时兼顾历史访问特征与近期写入变化. 为提高热数据识别的准确性, WHGC 利用全局写入序号构建指数衰减型逻辑页热度更新机制, 使逻辑页热度能够随写入行为持续动态变化, 在保留历史访问信息的同时更加突出近期写入行为, 从而更准确地反映逻辑页的更新活跃度.

(3) 基于热度等级与擦除等级映射的有效页迁移机制. 传统迁移策略通常将冷热数据简单映射至高磨损块或低磨损块, 匹配粒度较粗, 在热度判断不准确时容易导致迁移开销增加或磨损分布失衡. 为解决该问题, WHGC 通过对逻辑页热度与物理块擦除次数进行多等级划分, 并建立热度等级与擦除等级之间的映射关系, 在有效页迁移过程中根据热度等级选择匹配的目标块, 实现有效页的分级迁移与磨损均衡协同优化, 从而在降低 GC 迁移开销的同时延长 SSD 使用寿命.

2.1 写入模式感知的 GC 触发机制

当前多数垃圾回收方法通常采用固定阈值策略来触发 GC 操作^[9,11,19], 即当 SSD 中空闲页比例 (free page proportion, FPP) 低于某一预设值时立即执行 GC. 这种策略虽然实现简单, 但无法根据系统运行负载和主机写入行为的变化动态灵活调整 GC 的触发时机, 尤其在写入强度剧烈变化的场景下, 固定阈值易造成 GC 触发过早或滞后, 进而引发系统响应延迟或空间回收不及时等问题. 为增强 GC 触发的灵活性, 已有研究引入动态阈值机制^[10,22], 通过结合空间利用率等信息动态调整触发阈值. 然而, 现有方法的阈值调控主要基于存储介质的使用状态, 未能充分感知主机写入行为的动态变化, 在系统负载突变时响应能力有限. 因此, 本文提出一种写入模式感知的 GC 触发机制, 通过持续检测主机写入请求行为, 动态识别当前写入模式, 进而调整 GC 的触发策略. 具体而言, WHGC 将系统运行状态划分为“忙碌状态”和“空闲状态”, 在 SSD 的每个分组内独立维护 3 类 GC 触发阈值, 分别用于调度不可中断或可中断的 GC 任务, 以适应系统不同运行状态下的空间回收需求. 接下来将依次介绍 WHGC 的 3 类触发阈值设计、写入模式指标 L 的定义方法, 各类参数对阈值的调控机制, 以及基于 3 类阈值的 WHGC 触发策略逻辑.

2.1.1 触发阈值设计

为了更精确地反映 SSD 局部空间使用情况, 避免全局阈值下局部单元提前耗尽空闲页而触发频繁 GC, WHGC 在系统运行过程中为 SSD 的每个分组独立维护 3 类 GC 触发阈值, 分别为紧急触发阈值 $Th_{emergency}$, 忙碌态触发阈值 Th_{busy} , 空闲态触发阈值 Th_{idle} , 以满足不同运行状态下的 GC 需求. 已有研究表明, 这种分组独立管理阈值的策略能够提升 GC 触发的精确性与空间利用率^[10,22]. 基于该设计思想, WHGC 3 类触发阈值的定义与计算方法如下.

(1) 紧急触发阈值 $Th_{emergency}$: 当系统为写入请求分配空闲页时, 若当前分组的 FPP 低于该固定阈值, 系统立即执行 NGC 操作以保障写请求的及时处理, 该阈值为预设常量型参数, 如公式 (1) 所示:

$$Th_{emergency} = \alpha \quad (1)$$

(2) 忙碌态触发阈值 Th_{busy} : 当 FPP 介于 $Th_{emergency}$ 与 Th_{busy} 之间时, 系统仍需执行 NGC, 但阈值根据写入模式指标 L 与当前分组中的无效页比例 R_{inv} 动态调整, 如公式 (2) 所示:

$$Th_{busy} = \alpha + \beta \times \frac{1}{1 + e^L} \times R_{inv} \times \ln(1 + R_{inv}) \quad (2)$$

(3) 空闲态触发阈值 Th_{idle} : 当 FPP 介于 Th_{busy} 与 Th_{idle} 之间时, 系统不立即执行 GC 操作, 而是生成 IGC 请求节点, 延迟至系统空闲时期逐步调度执行, 如公式 (3) 所示:

$$Th_{idle} = \alpha + \omega \times \frac{1}{1 + L} \times R_{inv} \times \ln(1 + R_{inv}) \quad (3)$$

上述 3 个阈值需满足关系为 $0 < Th_{emergency} < Th_{busy} < Th_{idle} < 1$. 其中, 参数 α 、 β 和 ω 分别为 3 类触发阈值的基

准调控参数,取值范围为(0,1),用于控制 GC 触发的基准水平与变化幅度,可根据 SSD 的具体应用场景灵活设定.其中, Th_{busy} 与 Th_{idle} 的动态变化还受到主机写入模式指标 L 和当前分组中的无效页比例 R_{inv} 的共同影响,反映系统运行状态对 GC 触发策略的实时调节能力.为系统化阐述上述公式中写入模式指标 L 的定义方法及其在阈值调控中的作用,第 2.1.2 节将展开详细说明.

2.1.2 写入模式指标 L

为实现对主机写入行为模式的动态感知,本文设计了写入模式指标 L ,用于度量当前系统的写入密度特征.该指标以单位写入数据量所对应的主机写请求次数为依据,反映当前主机写入请求的强度与碎片程度.其定义如公式(4)所示:

$$L = \frac{\text{density}}{\text{density}_{\text{max}}} \quad (4)$$

其中,

(1) density 表示当前写入周期内的写入密度,定义为主机写请求次数 N_{write} 与写入数据量 S_{write} 之比,计算公式如下:

$$\text{density} = \frac{N_{\text{write}}}{S_{\text{write}}} \quad (5)$$

(2) $\text{density}_{\text{max}}$ 表示系统运行过程中记录的最大 density 值,用于对写入密度进行归一化处理,将 L 的取值范围限制在(0,1]之间.

在实际实现中,系统以累计写入数据量 S_{write} 达到 SSD 总容量的 $\theta\%$ (θ 为可配置参数) 作为一个写入周期的判定条件.每完成一个写入周期,系统统计一次当前的写入密度 density ,并更新 L 值,以实时反映主机写入模式的变化趋势. density 越大,表示单位数据量对应的写请求数量越多,主机发起的写请求比例越高,写入碎片化程度越严重.后文图 4 展示了两种写入模式下的写请求分布情况与对应的写入密度 density 计算示例.其中,(a) 为碎片化写入模式,(b) 为顺序写入模式.两个示例均完成了 6 页有效数据的写入,假设每页大小为 4 KB,则两个示例总计写入数据量均为 24 KB.示例 (a) 产生了 6 次写请求,计算得到 $\text{density} = 0.25$,而示例 (b) 仅产生 2 次写请求,计算可得 $\text{density} = 0.083$,在相同写入数据量下,写入密度差异显著.该示例验证了所设计的 density 指标能够准确刻画不同写入模式下的请求密集程度,具备良好的写入行为感知能力.相比于传统的 IOPS 或写入速率作为负载评估方式, density 指标更加轻量,且能准确捕捉主机写入行为的离散程度与突发性.

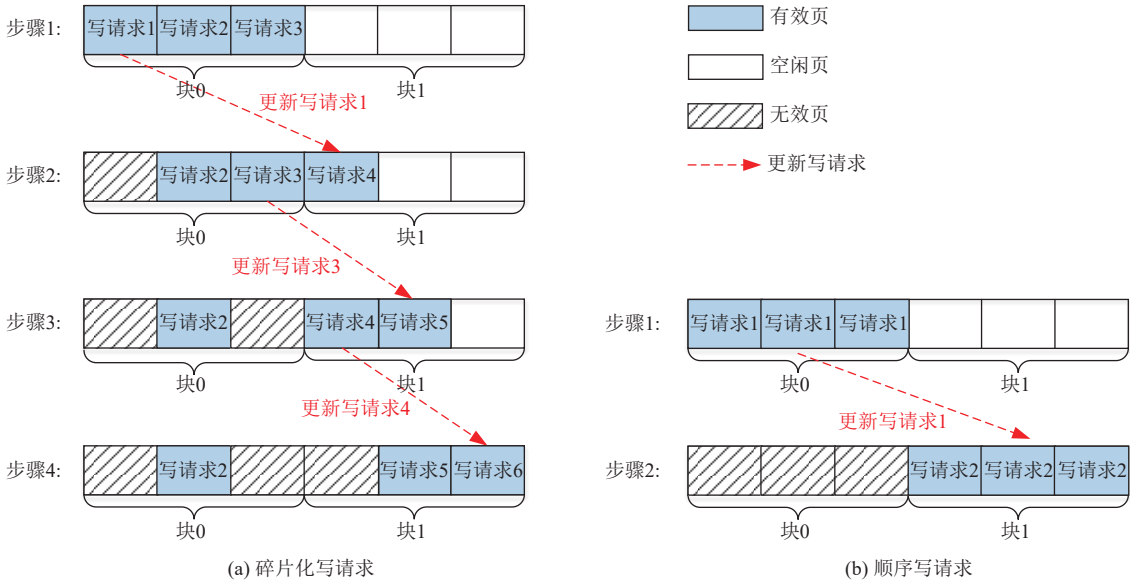


图 4 不同写入模式下的写请求分布示例及写入密度计算对比

通过对 $density$ 的归一化处理, 可动态构建写入模式指标 L , 为 GC 触发策略提供实时准确的负载判别依据. 因此, WHGC 将 L 应用于 Th_{busy} 和 Th_{idle} 的计算中. 当 L 较大时, 系统写入以高频小文件请求为主, 此时若触发 GC, 可能导致有效页迁移量显著上升, 影响系统性能; 而当 L 较小时, 系统写入顺序性强, 请求集中, 适宜执行 GC 实现低成本空间回收. 因此, 引入 L 可增强 GC 触发的负载敏感性与自适应能力.

2.1.3 阈值调控机制分析

为适应 SSD 系统在不同运行状态下的空间回收需求, WHGC 引入 3 类触发阈值, 分别对应系统所处的紧急、忙碌和空闲阶段, 构建分级调控的 GC 触发策略. 各类阈值在功能定位与调控方式上互为补充, 形成具有写入模式感知能力与空间自适应性的触发机制.

具体而言, 3 类阈值分别面向不同系统运行状态进行设计. 其中, 紧急触发阈值 $Th_{emergency}$ 用于应对写请求无法获得空闲页的极端情形, 由预设的常量型参数 α 定义, 在系统运行过程中保持不变, 用以保障系统在空间紧张状态下仍可强制触发 NGC 操作, 可确保在空闲页严重不足时仍能成功为写请求分配可用资源, 避免写入中断, 提升系统运行的连续性与处理稳定性.

忙碌态触发阈值 Th_{busy} 面向系统负载较高但写入仍持续进行的阶段. 为减轻此时 GC 操作对主机写入性能的干扰, WHGC 采用指数函数 $\frac{1}{1+e^L}$ 对 Th_{busy} 进行动态调控. 当 L 较小时, 表示写入请求集中且顺序性强, 阈值保持在较高水平, 有利于积极回收空间; 当 L 增大, 反映写入碎片化加剧, 函数值快速衰减, 阈值相应降低, 从而抑制 GC 的频繁触发, 减少对主机读写流程的影响. 该函数具备良好的非线性调节能力, 能够根据系统写入模式的变化灵活调整 GC 触发强度.

空闲态触发阈值 Th_{idle} 面向系统处于空闲状态但有空间回收需求的阶段. 此时 WHGC 采用变化更平缓的函数 $\frac{1}{1+L}$ 对阈值进行调控, 使 Th_{idle} 始终高于 Th_{busy} , 更适合在系统空闲阶段以较低代价逐步执行 GC 操作, 提升 IGC 操作的调度灵活性. 该策略适用于系统空闲状态, 表示系统空间尚未严重紧缺但可提前回收空间资源.

在上述两个动态阈值中, WHGC 进一步引入 SSD 当前分组的无效页比例 R_{inv} 作为空间状态调节因子, 以增强触发策略对空间使用情况的响应能力. 调节项 $R_{inv} \times \ln(1 + R_{inv})$ 随 R_{inv} 单调递增, 可准确反映当前分组的空间利用率. 当 R_{inv} 较大时, 表示无效页比例较高, 应更积极执行 GC. 对数项 $\ln(1 + R_{inv})$ 有助于抑制在 R_{inv} 增大时导致的阈值波动, 能够提升阈值变化的平稳性. 此外, 参数 α 、 β 和 ω 分别控制 3 类触发阈值的基础水平与动态变化幅度. 其中, α 为通用基准阈值, 统一设定各类触发阈值的起始水平; β 和 ω 分别用于调节 Th_{busy} 和 Th_{idle} 的变化强度, 用于灵活调整系统在不同运行状态下的 GC 触发频率. 各参数可根据 SSD 的使用寿命目标与典型负载特征进行灵活配置, 具备良好的可调节性与工程适应性.

综上, WHGC 所提出的 3 类触发阈值及其调控机制, 融合了主机写入行为与空间状态信息, 具备显著的自适应性与判别能力. 该机制不仅能有效应对多样化写入行为带来的空间管理挑战, 也为后续回收块选择与有效页迁移策略提供了稳定而精准的策略基础.

2.1.4 WHGC 触发策略逻辑

WHGC 触发策略的具体实现如算法 1 所示. 在处理写入请求时, 系统根据 SSD 当前写入分组的空闲页比例 FPP , 与 3 类触发阈值进行比较, 执行如下调度策略.

- 若 $FPP < Th_{emergency}$, 系统选择无效页最多的块作为回收块, 并立即执行 NGC (第 1–3 行).
- 若 $Th_{emergency} < FPP < Th_{busy}$, 系统仍需执行 NGC, 但采用回收块选择机制, 基于评分策略选择分数最小的块作为回收块 (第 4–6 行, 回收块选择机制见第 2.2 节).
- 若 $Th_{busy} < FPP < Th_{idle}$, 系统创建 IGC 请求节点, 在空闲时期调度执行 IGC, 回收块同样由评分策略确定 (第 7–9 行).
- 若 $FPP > Th_{idle}$, 表明空间充足, 系统清除当前分组中已挂起的 IGC 请求, 释放资源占用 (第 10–12 行).

算法 1. WHGC 触发策略.

 输入: 当前写入分组空闲页比例 FPP , 3 类触发阈值 $Th_{emergency}$ 、 Th_{busy} 、 Th_{idle} .

```

1. if  $FPP < Th_{emergency}$  then
2.   VictimBlock  $\leftarrow$  SelectBlockMaxInvalid() // 选择无效页最多的块作为回收块
3.   TriggerEmergencyGC(VictimBlock) // 执行 NGC
4. else if  $Th_{emergency} < FPP < Th_{busy}$  then
5.   VictimBlock  $\leftarrow$  SelectBlockMinScore() // 基于评分策略选择回收块
6.   TriggerBusyGC(VictimBlock) // 执行 NGC
7. else if  $Th_{busy} < FPP < Th_{idle}$  then
8.   VictimBlock  $\leftarrow$  SelectBlockMinScore() // 基于评分策略选择回收块
9.   EnqueueRequestIGC(VictimBlock) // 创建 IGC 请求节点
10. else
11.   ClearPendingIGC() // 空间充足, 清除挂起的 IGC 请求节点
12. end if
  
```

2.2 回收块选择机制

在 GC 被触发后, 系统需要确定当前应回收的目标块. 回收块的选择策略影响有效页迁移的开销与磨损分布水平, 进而影响整体的写入性能与 SSD 的使用寿命. 传统回收块选择方法多采用贪婪策略, 即选择无效页最多块作为回收块, 以快速释放空间资源^[5,22]; 也有方法设计综合评分函数, 综合考虑块的无效页比例与擦除次数, 以兼顾空间利用与磨损状态^[8,10,18]. 此外, 还有方法引入块中无效页的失效时间作为指标, 优先回收失效时间较长的无效页^[17,27]; 还有研究融合空间状态、磨损状态与更新频率等多维信息构建评分机制, 以提升回收决策的综合性与适应性^[9,28].

尽管上述方法各具优势, 但在实际运行过程中仍存在以下问题: 一方面, 在系统运行进入后期, 整体磨损积累趋于严重时, 部分高磨损块若仍被频繁回收, 会加剧产生坏块的风险, 影响数据可靠性; 其次, 为追求极致磨损均衡, 部分方法倾向优先回收擦除次数最小的块, 虽然能减少块间磨损差距, 但由于这类回收块常含有大量有效页, 可能会导致迁移数据量激增, 影响系统写入性能.

为实现性能与均衡的协同优化, WHGC 设计了一种融合空间状态与磨损状态的回收块评分机制, 在每次触发 GC 时, 系统对当前分组内的所有块根据评分函数 (公式 (6)) 进行评分, 并选择分数最小的块作为回收块, 其具体实现逻辑如算法 2 所示.

$$Score_i = (1 - \mu) \left[(1 - \lambda)(1 - R_{inv(i)}) + \lambda \frac{N_{erase(i)}}{E_{max} + 1} \right] + \mu \frac{N_{erase(i)}}{E_{max} + 1} \quad (6)$$

其中, 各参数含义如下.

(1) $R_{inv(i)}$: 块 i 的无效页比例.

(2) $N_{erase(i)}$: 块 i 的擦除次数.

(3) E_{max} : 当前分组中的最大擦除次数.

(4) μ : 系统当前磨损程度系数, 定义如公式 (7) 所示, 其中 E_{avg} 表示当前分组的平均擦除次数, E_{limit} 为块的擦除上限, k 为调节因子.

$$\mu = 1 - e^{-k \frac{E_{avg}}{E_{limit}}} \quad (7)$$

(5) λ : 空闲块比例, 定义为分组的空闲块个数 N_{free} 与总块数 N_{total} 之比, 如公式 (8) 所示:

$$\lambda = \frac{N_{free}}{N_{total}} \quad (8)$$

算法 2. 回收块选择算法 SelectBlockMinScore.

输入: 当前写入分组中所有块的状态信息;

输出: 分数最小的回收块 VictimBlock.

```

1.  $Score_{min} \leftarrow \infty$ 
2. VictimBlock  $\leftarrow$  NULL
3. for each block  $i$  in current write plane do
4.   Compute  $Score_i$  for block  $i$ 
5.   if  $Score_i < Score_{min}$  then
6.     VictimBlock  $\leftarrow i$ 
7.      $Score_{min} \leftarrow Score_i$ 
8.   end if
9. end for
10. return VictimBlock

```

公式 (6) 的评分函数可拆分为 3 部分, 分别对应空间状态感知、轻度磨损调节与全局磨损保护这 3 种不同的回收目标. 第 1 项 $(1-\mu)(1-\lambda)(1-R_{inv(i)})$ 以空间使用状态为主导, 反映系统对块中无效页比例的关注程度, 主要在系统处于早期阶段, 整体磨损较轻时发挥主导作用. 当空闲块比例 λ 较低 (空间紧张) 时, 系数 $(1-\lambda)$ 趋近于 1, 系统更倾向于回收无效页比例高的块, 以快速释放可用空间、保障主机写入请求的及时处理; 而当 λ 较大 (空间充足) 时, 该项权重降低, 回收策略向磨损均衡导向过渡. 第 2 项 $(1-\mu)\lambda\frac{N_{erase(i)}}{E_{max}+1}$ 为轻度磨损调节项, 主要在系统空间资源充足 (λ 较大), 磨损尚轻时 (μ 较小) 发挥作用. 相比第 1 项仅关注空间使用情况, 该项引入块的擦除次数 $N_{erase(i)}$, 使系统在性能导向策略下, 能适度规避高磨损块, 降低其提前进入坏块状态的风险, 从而实现性能与寿命之间的平衡过渡. 第 3 项 $\mu\frac{N_{erase(i)}}{E_{max}+1}$ 为磨损均衡主导项, 是评分函数中最具“寿命导向”的部分, 当系统运行进入晚期, E_{avg} 持续上升, 导致 μ 值增大, 该项逐渐主导回收决策, 使系统倾向于回收擦除次数较低的块, 主动避免高磨损块的频繁擦除, 从而延缓其失效过程, 有效提升系统可靠性与 SSD 的使用寿命.

以上 3 项的协同调节依赖于两个核心因子: 空间状态因子 λ 与磨损感知因子 μ . 其中, μ 的增长受公式 (7) 所定义的非线性函数控制, 其变化速率由调节因子 k 决定. 较大的 k 值使系统更早以磨损均衡为导向, 强调寿命保护; 较小的 k 则允许系统长时间维持性能优先策略. 实际部署中可根据 SSD 的目标场景灵活配置 k 的数值, 以平衡性能与可靠性的优先级.

综上, WHGC 的回收块选择机制融合了空间利用与磨损状态两个维度, 能根据 SSD 所处生命周期阶段, 动态切换回收策略导向. 在系统早期强化性能保障, 在系统晚期强化寿命保护, 最终在写放大、使用寿命与系统响应之间取得合理平衡. 该评分机制也为后续的有效页迁移策略提供了稳定而合理的回收块基础.

2.3 逻辑页热度识别机制

在 GC 过程中, 一旦确定了回收块, 系统首先判断其是否包含有效页: 若该块中全部为无效页, 则可直接擦除, 无需执行数据迁移; 若仍存在有效页, 则需进一步执行有效页迁移操作. 此时面临以下 3 个关键问题: 1) 是否需要有效页进行分类迁移? 2) 若需要, 应依据何种属性进行划分? 3) 分类后应如何制定迁移策略? 上述 3 方面的设计会对 GC 的执行效率与 SSD 系统的可靠性产生重要影响.

现有部分 GC 方法在迁移有效页时并不区分数据的冷热属性^[10,11], 而是将回收块中的全部有效页迁移至同一个目标块中, 容易导致冷热数据混合存储, 增加写放大与块间磨损不均, 降低回收效率并造成资源浪费. 也有部分方法使用页的写入时间戳信息^[8,17]或累计写入次数^[27]记录热度信息, 或在主机写入阶段根据逻辑页更新次数进行分类存储^[24,25]. 这些方法存在热度识别精度低, 更新滞后等问题, 难以及时感知主机对数据访问行为的变化, 导致

迁移策略难以适应动态变化的访问模式. 针对上述问题, WHGC 引入基于热度的分类迁移机制, 能够准确识别并合理利用有效页的热度属性, 进而提升 GC 效率与系统可靠性, 并延长 SSD 使用寿命.

在实现页级热度识别时, 首先需明确记录对象是逻辑页 (logical page number, LPN) 还是物理页 (physical page number, PPN). 在 SSD 的运行过程中, 逻辑页由主机系统发出, 是主机可感知的访问单元, 而物理页是闪存芯片中实际存储数据的地址单元. 由于闪存采用异地更新机制, 每次逻辑页更新都会写入至新的物理页, 并通过 FTL 动态维护逻辑页与物理页的映射关系. 物理页的位置在实际运行中会因地址映射、垃圾回收或磨损均衡等机制频繁变化, 导致其热度状态难以稳定记录, 且无法准确反映主机访问行为. 相比之下, 逻辑页作为主机层的访问单元, 其访问频度与时间特性具备更强的可追踪性与稳定性. 因此, WHGC 采用逻辑页粒度维护热度信息, 使系统能够准确捕捉逻辑页的写入频率与更新间隔等活跃特征, 为后续的热度分级与数据迁移策略提供基础支持.

WHGC 设计了一种指数衰减型的逻辑页热度识别机制, 用于在主机每次写入过程中准确更新逻辑页的热度信息. 系统维护全局写入序号 (write sequence number, WSN), 用于表示主机写入操作的时序. 每当逻辑页被写入 (包括首次写入和更新写入) 时, WSN 值递增, 同时更新该逻辑页的热度值 $Heat[LPN]$, 以动态反映其写入活跃度. 逻辑页热度的更新方式如下所示:

$$Heat[LPN] = Heat[LPN] \times \gamma^{\Delta s} + 1 \quad (9)$$

$$\Delta s = WSN_{current} - WSN_{last}[LPN] \quad (10)$$

其中, 参数含义如下.

- (1) $Heat[LPN]$: 逻辑页 LPN 的热度值.
- (2) $WSN_{current}$: 当前写入时的 WSN .
- (3) $WSN_{last}[LPN]$: 逻辑页 LPN 最近一次写入时的 WSN .
- (4) Δs : 逻辑页 LPN 当前写入与最近一次写入的 WSN 差值.
- (5) γ : 热度衰减因子, 控制历史热度的遗忘速度, 取值范围为 (0, 1).

图 5 展示了各关键变量之间的结构关系及逻辑页热度更新的执行路径, 包括主机写入触发、逻辑页识别与热度表项的同步更新. 其对应的数值计算过程如算法 3 所示, 系统在每次逻辑页写入时, 通过计算写入序号差值 Δs , 对 $Heat[LPN]$ 执行指数衰减操作, 并在此基础上增加 1, 用于反映当前写入对该页热度的激活效果, 强调其近期活跃性, 从而实现热度值 $Heat[LPN]$ 的动态维护. 该机制在逻辑页粒度上持续记录热度变化, 能够敏锐捕捉主机访问行为的变化, 为后续数据迁移策略提供准确热度依据. 其设计以写入序号差异为核心依据, 能够兼顾历史热度与近期写入的权重. 控制因子 γ 决定了衰减速度, γ 越小, 热度衰减越快, 系统对近期写入操作越敏感; γ 越大, 热度衰减越慢, 系统对历史热度保留得越多. 系统可根据实际系统需求与工作负载特性灵活配置 γ 值, 以更合理地反映逻辑页的访问频率与更新特征.

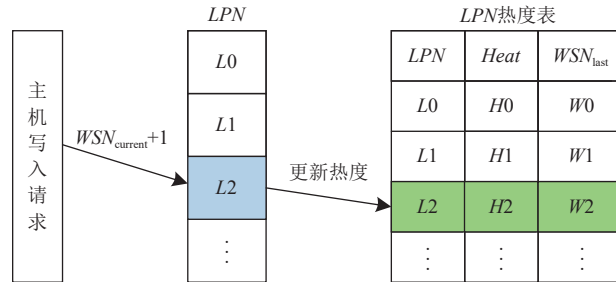


图 5 逻辑页热度更新示意图

算法 3. 逻辑页热度更新算法.

输入: 逻辑页编号 LPN , 更新前的 $Heat[LPN]$, 当前全局写入序号 $WSN_{current}$, 逻辑页最近一次写入时的 $WSN_{last}[LPN]$;
输出: 更新热度表 $Heat[LPN]$.

```

1.  $\Delta s \leftarrow WSN_{current} - WSN_{last}[LPN]$  // 计算写入序号差值
2.  $Heat[LPN] \leftarrow Heat[LPN] \times \gamma^{\Delta s} + 1$  // 指数衰减并加权激活当前写入
3.  $WSN_{last}[LPN] \leftarrow WSN_{current}$  // 更新写入序号记录
4. return  $Heat[LPN]$ 

```

相比于传统基于时间戳或写入次数差值的线性热度度量方法, WHGC 所提出的指数衰减型热度机制可自然弱化历史写入的影响, 更突出最近被频繁访问的逻辑页, 有效避免冷数据因早期写入而长期维持高热度的问题, 从而提升热数据识别的准确性. 此外, 该机制仅依赖当前与上次逻辑页的写入序号, 无需维护时间戳或滑动窗口, 空间开销小, 且不依赖系统时钟, 适配性强, 适用于多种 SSD 实际部署场景. 在 WHGC 框架中, 该热度识别机制将在第 2.4 节中指导有效页迁移的目标块选择, 实现基于热度等级匹配的数据迁移策略, 从而进一步提升 WHGC 在空间回收与寿命延长方面的整体表现.

2.4 热度分级与有效页迁移机制

在完成回收块选择后, 若该块中包含有效页, 系统需对其进行迁移操作. 迁移策略不仅会影响写放大与系统延迟, 还会显著影响磨损均衡效果, 进而影响 SSD 的可靠性. 因此, 设计一套高效、合理的有效页迁移机制, 对提升 GC 性能与系统可靠性具有关键作用.

现有 GC 方法普遍采用二元映射策略, 即将冷数据迁移至擦除次数最大的块, 热数据迁移至擦除次数最小的块^[9,19]. 尽管该策略能够达到冷热数据分离的效果, 但在实际运行中存在两方面问题. 首先, 热度判断往往依赖固定的访问频率或时间间隔划分, 忽略了逻辑页热度的连续性特征, 导致数据的热度容易被误判, 从而影响目标块的准确匹配. 其次, 该策略将数据热度与极端磨损块绑定, 缺乏灵活性, 一旦热度判断出现偏差, 热数据可能频繁写入高磨损块, 冷数据长期占用低磨损块, 反而加剧块间磨损不均衡, 缩短 SSD 的使用寿命.

为此, 本文提出一种基于热度等级与擦除等级映射的有效页迁移机制. 不同于传统将冷数据迁移至高磨损块, 热数据迁移至低磨损块的迁移策略, 该机制不再追求对数据冷热的绝对划分, 而是将逻辑页热度与物理块的擦除次数分别划分为多个等级, 并建立两者之间的映射关系, 根据等级匹配原则完成有效页迁移. 具体实现包括以下 3 个部分.

(1) 有效页热度与块擦除次数的等级划分

在回收块所在分组内, 系统将所有有效页对应的逻辑页热度值 $Heat[LPN]$ 等值划分为 n 个热度等级 (0 表示最冷数据等级, $n-1$ 表示最热数据等级); 同时, 将该分组内所有物理块的擦除次数等值划分为 n 个擦除等级 (0 表示最小擦除次数等级, $n-1$ 表示最大擦除次数等级). 由此, 每个有效页与物理块均具有明确的“热度等级”或“擦除等级”属性. 其中, 划分等级数 n 为系统初始化时设定的固定参数, 用于控制热度与擦除状态的匹配粒度, 在运行过程中保持不变.

(2) 等级映射与目标块选择

在完成有效页热度等级和目标块擦除等级的划分后, 系统需为每个有效页选择匹配的目标块以执行迁移操作. 对于热度等级为 m 的有效页, 系统按照以下优先级规则进行目标块选择 (完整流程见算法 4).

- 优先级 1. 系统查找擦除等级为 $e = n - m - 1$ 的块集合, 若该集合中存在空闲块, 则选择其中擦除次数最小的块作为目标块 (算法 4 第 7-18 行).

- 优先级 2. 若擦除等级 e 中无空闲块, 系统在该等级内查找非满块 (仍有空闲页) 集合, 计算每个块的平均热度 $BlockHeat[blk]$, 如公式 (11) 所示:

$$BlockHeat[blk] = \frac{1}{N_{valid}} \sum_{i=1}^{N_{valid}} Heat[LPN_i] \quad (11)$$

其中, blk 为候选块块号, N_{valid} 为块内有效页数量, $Heat[LPN_i]$ 为该块中第 i 个有效页对应的逻辑页热度值. 将当前有效页热度值 $Heat[LPN]$ 与每个候选块的 $BlockHeat[blk]$ 相减, 选择热度差值绝对值最小的块作为目标块, 以实

现相似热度匹配 (算法 4 第 19–32 行).

• 优先级 3. 若擦除等级 e 中所有块均为满块 (没有空闲页), 系统在相邻等级集合 $e' = \{e-1, e+1, e-2, e+2, \dots\}$ 中继续执行优先级 1 与优先级 2 的查找过程, 按照与 e 的差值递增顺序依次遍历各等级, 在差值相同的情况下优先考虑较小的等级 (如 $e-1$ 优先于 $e+1$), 直至找到可用目标块 (算法 4 中第 3–6 行与第 7–32 行循环体).

算法 4. 基于热度-擦除等级映射的目标块选择策略.

输入: 划分等级数 n , 有效页的逻辑页号 LPN , 有效页的热度等级 m , 有效页的热度值 $Heat[LPN]$, 擦除等级块集合数组 $EraseLevel[]$;

输出: 目标块 $TargetBlock$.

```

1.  $e \leftarrow n - m - 1$  // 匹配的目标擦除等级
2. Initialize  $TargetBlock \leftarrow NULL$ ,  $MinErase \leftarrow \infty$ ,  $MinHeatDiff \leftarrow \infty$ 
3. for  $offset \leftarrow 0$  to  $n - 1$  do // 优先级 3: 逐步向相邻擦除等级扩展
4.   for each  $e'$  in  $\{e - offset, e + offset\}$  do
5.     if  $e' < 0$  or  $e' \geq n$  then
6.       continue
7.     // 优先级 1: 在擦除等级  $e'$  中选择擦除次数最小的空闲块
8.     for each  $blk$  in  $EraseLevel[e']$  do
9.       if  $blk$  is FreeBlock then // 空闲块
10.        if  $EraseCount(blk) < MinErase$  then
11.           $TargetBlock \leftarrow blk$ 
12.           $MinErase \leftarrow EraseCount(blk)$ 
13.        end if
14.      end if
15.    end for
16.    if  $TargetBlock \neq NULL$  then
17.      return  $TargetBlock$  // 找到空闲块立即返回
18.    end if
19.    // 优先级 2: 在非满块中选择平均热度最接近的块
20.    for each  $blk$  in  $EraseLevel[e']$  do
21.      if  $blk$  is not full and is not FreeBlock then
22.         $BlockHeat[blk] \leftarrow AverageHeat(blk)$  // 计算块的平均热度
23.         $Diff \leftarrow |Heat[LPN] - BlockHeat[blk]|$ 
24.        if  $Diff < MinHeatDiff$  then
25.           $TargetBlock \leftarrow blk$ 
26.           $MinHeatDiff \leftarrow Diff$ 
27.        end if
28.      end if
29.    end for
30.    if  $TargetBlock \neq NULL$  then
31.      return  $TargetBlock$  // 找到热度接近块立即返回
32.    end if

```

33. end for

34. end for

为更直观理解目标块的选择逻辑, 图 6 展示了一个热度等级与擦除等级映射匹配的示例. 假设当前回收块中有 3 个有效页 ($PPN0$ 、 $PPN1$ 、 $PPN2$), 其对应的逻辑页号为 $LPN1$ 、 $LPN2$ 、 $LPN0$. 根据逻辑页热度信息, 并结合划分等级数 $n=3$, 系统将这 3 个有效页的热度等级分别确定为 $m=0$ 、 $m=1$ 、 $m=2$. 同时, 将当前分组的所有物理块按擦除次数划分为 3 个等级: 块 0 的擦除等级 $e=0$, 块 1 和块 2 为 $e=1$, 块 3 为 $e=2$. 根据等级映射关系 $e=n-m-1$, 系统依次为每个有效页查找匹配的目标块: $PPN0$ ($m=0$) 对应擦除等级 $e=2$, 块 3 为空闲块, 直接将 $PPN0$ 迁移至块 3; $PPN1$ ($m=1$) 对应擦除等级 $e=1$, 该等级中无空闲块, 系统计算块 1 和块 2 的平均热度后, 识别出块 2 与 $PPN1$ 的热度值更接近, 因此将 $PPN1$ 迁移至块 2; $PPN2$ ($m=2$) 对应擦除等级 $e=0$, 块 0 无可用空间, 系统按回退机制查找相邻等级 $e=1$, 最终选择与 $PPN2$ 热度最接近的块 1 作为目标块, 并将 $PPN2$ 迁移至块 1. 最终, 3 个有效页分别迁移至与其热度等级完全或近似匹配的目标块.

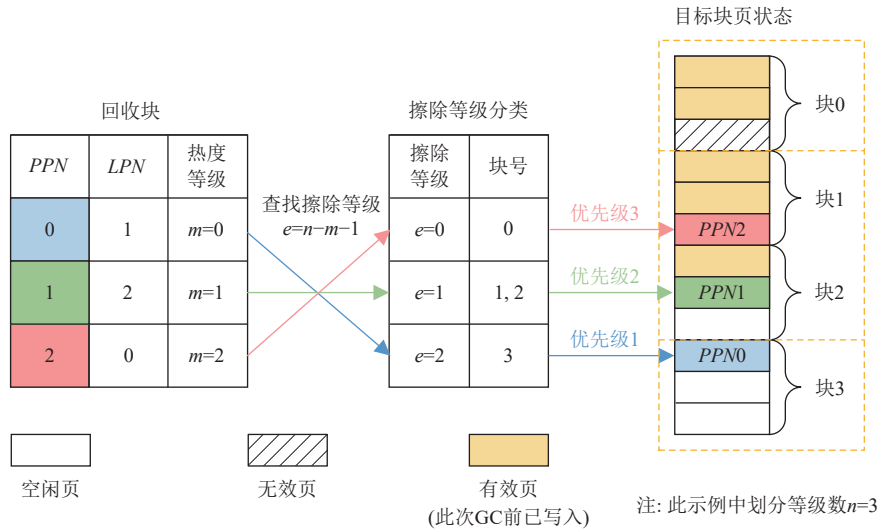


图 6 热度等级与擦除等级映射机制下的有效页迁移示例

(3) 有效页迁移与 GC 执行方式差异

在目标块确定后, 系统根据两种 GC 模式采用不同的迁移策略: 对于 NGC, 在 GC 触发后, 系统立即迁移回收块中的所有有效页并擦除该块; 对于 IGC, 系统创建 IGC 节点, 确定回收块, 但并不立即确定目标块. 随后, 系统在空闲阶段确认有效页状态, 并根据热度分级与有效页迁移机制, 为每个有效页选择合适的目标块进行迁移. NGC 适用于需要高效快速回收的场景, 但可能阻塞系统 I/O 操作的执行; IGC 适用于系统空闲阶段, 能够降低 GC 操作对正常 I/O 操作的干扰, 该机制在不同 GC 模式下保持一致的热度映射与目标块选择策略, 主要区别在于迁移时机与调度方式的差异化设计, 以适应系统在不同运行状态下的 GC 需求.

综上, 本文提出的热度分级与有效页迁移机制突破了传统冷热数据二元划分的局限性, 通过逻辑页热度与物理块磨损状态的双维度协同, 构建了基于等级映射、热度匹配与有序回退的细粒度迁移控制策略. 相较于现有方法, 该机制在系统鲁棒性与调度灵活性方面具有显著优势, 主要体现为以下 4 个方面.

(1) 分级映射的系统化设计

采用等值划分策略, 分别对逻辑页热度和块擦除次数进行多等级划分, 细化数据与资源状态的管理粒度, 构建了更具灵活性的映射体系. 相比传统的冷热二元划分方法, 该策略提升了热度识别的准确性, 为每个有效页提供明确的目标块选择范围, 增强了迁移策略对复杂数据分布的适应能力, 提升了策略的鲁棒性与系统控制精度.

(2) 基于热度的目标块匹配策略

在有效页迁移过程中, 当无空闲块可用时, 系统在目标擦除等级的块集合中, 根据已写入页的平均热度, 选择与当前有效页热度相近的块作为目标块, 实现热度相近页的集中写入. 该策略减少了冷热数据混写导致的更新冲突, 降低了频繁迁移带来的开销, 同时提升了空间利用效率与写入效率.

(3) 动态容错与等级回退机制

当目标擦除等级无可块时, 系统启动相邻等级回退机制, 按照与目标擦除等级的差值递增顺序, 逐级在相邻擦除等级中查找可用块, 确保迁移操作不中断. 该策略增强了系统在资源紧张或数据分布异常情况下的适应能力, 提升了迁移操作的连续性与鲁棒性.

(4) 磨损均衡与寿命延长能力

通过将逻辑页热度等级与物理块磨损等级建立映射关系, 系统在迁移过程中兼顾当前页的更新活跃度与目标块的擦除状态, 避免热数据聚集至高磨损块、冷数据长期占用低磨损块, 从而降低块失效风险, 延长 SSD 使用寿命.

因此, 该机制在热度识别、目标块匹配、容错设计与寿命管理等方面实现协同优化, 具有更强的灵活性、适应性与耐久性, 特别适用于嵌入式或高可靠场景下的 SSD 管理需求.

3 实验分析

3.1 实验环境与数据集

为评估所提出的垃圾回收方法在实际应用中的性能与耐久性表现, 本文采用开源 SSD 仿真平台 SSDsim^[39]进行跟踪驱动的模拟实验. SSDsim 是当前 SSD 研究中广泛使用的模拟器, 其模拟机制经过多项研究验证, 能够较为准确地复现 SSD 的实际运行行为, 在学术研究中具有良好的代表性与通用性.

实验所采用的 SSD 配置如表 1 所示. 为贴近嵌入式系统及成本敏感型 SSD 的实际部署环境, 实验中将 SSD 总容量设为 4 GB, 物理块擦除限制设为 500 次^[40], 以模拟可靠性要求较高的嵌入式设备的典型特征. 该配置能够在有限擦除周期内更早显现不同垃圾回收方法对系统寿命的影响差异, 有助于更敏感地评估其在 SSD 寿命控制方面的实际效果.

表 1 SSDsim 仿真平台配置参数

参数	数值	参数	数值
通道数	2	闪存页大小	8 KB
芯片数/通道	1	读时延	20 μs
晶圆数/芯片	2	写时延	200 μs
分组数/晶圆	1	擦除时延	1.5 ms
块数/平面	2 048	地址映射策略	页映射
页数/块	64	块擦除限制	500次

为全面评估所提垃圾回收方法在不同负载场景下的性能与有效性, 本文选用了来自微软剑桥研究院公开发布的 6 个公开数据集^[41]. 该数据集在 SSD 存储系统研究中被广泛采用, 能够较好地反映真实系统中的 I/O 访问特征. 这些工作负载涵盖了多种真实应用环境, 具备代表性强, 差异性显著等特点, 有助于测试垃圾回收机制在不同 I/O 模式与访问特征下的适应性表现. 表 2 总结了所选数据集的主要特征, 所覆盖的场景既包括写密集型负载 (如 prxy_0, rsrch_0), 也包括读写相对均衡的混合负载 (如 usr_0, hm_0). 其中, 写请求的平均大小在 4.65–10.28 KB 之间, 体现了不同工作负载在访问模式上的差异性.

3.2 实验对比方法

为全面评估 WHGC 的性能和优势, 本文选择了 4 种具有代表性的 GC 方法作为对比对象, 分别为 Greedy、DA-GC、USIGC 与 TPGC. 这些方法在 GC 触发策略、回收块选择机制、冷热数据处理方式、有效页迁移策略及调度模式等方面具有明显差异, 能够较全面地覆盖现有主流 GC 机制的设计思路, 为实验分析提供多维度对照基础.

表 2 实验数据集

数据集	请求总数	写请求比例 (%)	写请求平均大小 (KB)	读请求平均大小 (KB)
prxy_0	12519868	96.94	4.65	8.33
rsrch_0	1433655	90.68	8.73	10.92
usr_0	2237889	59.58	10.28	40.92
ts_0	1801734	82.42	8.01	13.68
hm_0	3993316	64.50	8.34	7.37
src2_0	1557814	88.66	7.09	8.10

(1) Greedy^[5]: 经典的贪婪算法, 采用固定触发阈值, 仅在系统忙碌状态下执行 NGC. 回收块选择方式为贪婪式策略, 即选择当前无效页最多的块进行回收. 该方法不区分冷热数据, 所有有效页统一迁移至同一目标块.

(2) DA-GC^[10]: 采用两个动态触发阈值, 支持在系统空闲状态下执行 IGC, 在忙碌状态下执行 NGC. 回收块选择依据评分策略, 综合考虑块的无效页比例与擦除次数等状态信息. 该方法不区分冷热数据, 所有有效页统一迁移至同一目标块.

(3) USIGC^[11]: 采用两个固定触发阈值, 同样支持在系统空闲状态下执行 IGC, 在忙碌状态下执行 NGC. 回收块选择基于无效页比例与擦除次数的评分策略. 该方法不区分冷热数据, 所有有效页统一迁移至同一目标块.

(4) TPGC^[9]: 采用固定触发阈值, 仅在系统忙碌状态下执行 NGC. 其回收块选择综合考虑块的有效页比例、擦除次数、无效页比例以及页的失效时间. 该方法通过写入序号和时间戳判定页的热度等级, 实现冷热数据划分. 在迁移阶段, 根据有效页的热度值选择合适的目标块, 实现冷热数据的分类迁移.

为进一步对比本文提出的方法 WHGC 与上述 4 种 GC 方法的设计异同, 表 3 总结了各方法在 GC 触发条件、回收块选择机制、冷热数据识别、迁移策略及调度模式等方面的关键特性.

表 3 各方法设计特性

方法	触发阈值类型	回收块选择机制	是否区分冷热数据	有效页迁移策略	调度模式
Greedy	固定	无效页最多	否	统一迁移	NGC
DA-GC	动态	评分	否	统一迁移	IGC+NGC
USIGC	固定	评分	否	统一迁移	IGC+NGC
TPGC	固定	评分	是	分类迁移	NGC
WHGC (本文)	动态	评分	是	分类迁移	IGC+NGC

3.3 实验结果与分析

为全面客观地评估各垃圾回收方法的性能, 本文选取以下 6 个关键指标作为对比标准: 1) 主机写入量; 2) 写放大因子; 3) GC 迁移有效页数量; 4) GC 触发频率; 5) 擦除次数标准差; 6) 平均响应时间. 在实验参数设置中, 为保证实验结果的公平性并充分展示各垃圾回收方法的实际效果, Greedy、DA-GC、USIGC 和 TPGC 方法均采用原始论文中所提出的最优参数配置, 以确保其在实验中达到最佳表现. 针对本文提出的 WHGC 方法, 则采用统一的参数配置, 具体为: $\alpha = 0.01$, $\beta = 0.25$, $\omega = 0.01$, $\theta = 0.01$, $k = 4$, $\gamma = 0.9$, $n = 4$. 在实验过程中, 系统按照固定主机写入间隔对运行状态进行周期性统计, 即每完成 20 万次主机写入后记录一次系统状态, 统计各项性能指标, 并观察其随主机写入规模变化的趋势. 此外, 为保证评估的一致性, 所有实验均在系统运行中第 1 个块达到擦除上限时终止.

3.3.1 主机写入量

图 7 展示了 5 个 GC 方法在 6 种数据集下的主机写入量, 统计指标为系统首次出现擦除上限块时累计的主机写入量. 实验结果以 Greedy 为基准进行归一化, 横坐标括号内为 Greedy 的实际主机写入量. 由于仿真设定以首次出现擦除上限块为终止条件, 主机写入量越多代表 SSD 在首次出现失效块前完成的写入操作越多. 因此, 该指标可用于评估各类 GC 方法在延长 SSD 使用寿命方面的有效性.

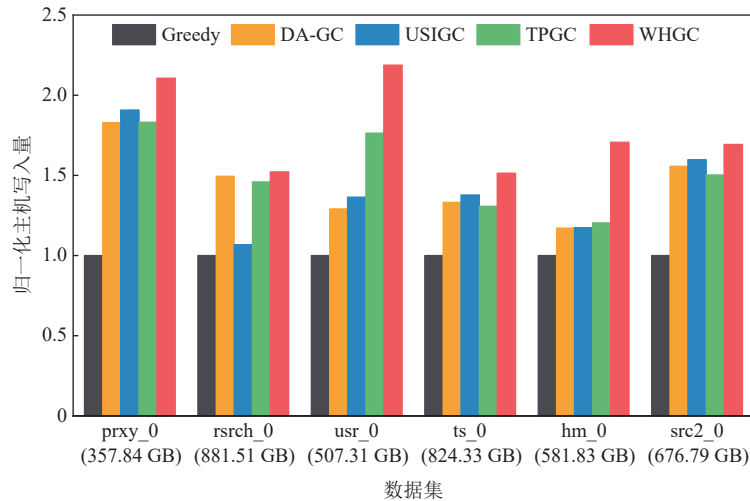


图7 归一化主机写入量

整体来看, WHGC 在所有数据集上均实现了最多的主机写入量, 相较于 Greedy、DA-GC、USIGC 和 TPGC 平均提升幅度分别约为 78.78%、25.73%、29.04% 和 18.88%。该结果表明, WHGC 在多种负载类型下均具备稳定而显著的寿命延长能力, 表现出良好的鲁棒性与通用性。

从数据集特征进一步分析: 在写请求平均大小显著小于页大小 (8 KB) 的负载中, 如 prxy_0, 主机写入高度碎片化, 易导致频繁触发 GC 和有效页迁移。此类场景下, Greedy 因缺乏磨损均衡与冷热感知机制, 导致最先出现失效块; DA-GC 与 USIGC 虽具备一定磨损感知能力, 但回收块选择与迁移过程未考虑页热度, 仍存在无效迁移开销; TPGC 在固定阈值触发机制下执行 GC, 且其磨损均衡策略倾向选择擦除次数最小的块, 导致迁移成本上升, 影响寿命表现。相比之下, WHGC 基于主机写入模式自适应调整 GC 触发阈值, 并结合指数衰减式热度识别, 有效识别并分类迁移冷热数据, 在此类高碎片度场景下可显著延长设备寿命。

在写请求大小接近页粒度, 写入比例较高的负载中, 如 rsrch_0、ts_0 和 src2_0, 回收块中多聚集无效页, GC 操作多为直接擦除, 无需迁移有效页, 有利于简化回收过程。此类场景下, Greedy、DA-GC 和 USIGC 等方法因优先选择无效页最多的块, 无需热度识别即可取得较好的 GC 效果。TPGC 为了维持长期的磨损均衡, 在特定间隔后会强制选择擦除次数最小的块作为回收对象, 尽管提升了磨损均衡性, 但引入了大量额外迁移, 反而抑制了主机写入能力。相比之下, WHGC 虽因冷热迁移机制受限未完全发挥优势, 但其回收块选择机制融合空间利用率与磨损状态, 主动保护高磨损块, 避免其频繁被选为回收块。因此, 即使在迁移操作需求较低的场景下, WHGC 仍实现了最多的主机写入量, 表现出良好的负载适应性与磨损控制能力。

在读写混合型负载中, 如 usr_0 与 hm_0, 写入比例下降且读写交替频繁, 数据热度分布不均, 使得 GC 难以高效回收无效页, 从而影响整体效率。Greedy、DA-GC 与 USIGC 均缺乏热度感知机制, 易导致数据频繁迁移, 加速部分块的磨损, 缩短设备寿命。TPGC 虽引入热度划分, 但冷热区分不够准确, 寿命提升有限。WHGC 所采用的基于写入序号驱动的指数衰减热度识别机制, 结合热度等级映射与分级迁移策略, 能够在热度动态波动场景下有效引导数据迁移, 实现更优的冷热数据分类与磨损均衡, 显著延长设备使用寿命。

综上所述, Greedy 因完全忽略磨损状态与热度信息, 主机写入量始终最少; DA-GC、USIGC、TPGC 等方法在特定数据集下表现尚可, 但效果受负载特性波动较大, 缺乏稳定性; WHGC 通过写入模式感知、回收块选择、逻辑页热度识别与热度等级映射迁移的协同优化, 在多种负载特性下均实现了稳定而显著的寿命延长效果, 体现出其对系统写入行为的感知能力与策略动态调整能力, 具备良好的负载自适应性。

3.3.2 写放大系数

图 8 展示了在 6 种数据集下, 5 个 GC 方法的写放大系数随主机写入次数的变化趋势。横坐标为主机写入次

数, 纵坐标为写放大系数. 该指标定义为闪存实际写入数据量与主机写入数据量之比, 其中闪存实际写入量包括主机写入以及垃圾回收过程中产生的数据迁移写入. 写放大系数越低, 表示系统额外写入开销越小, 空间利用效率越高.

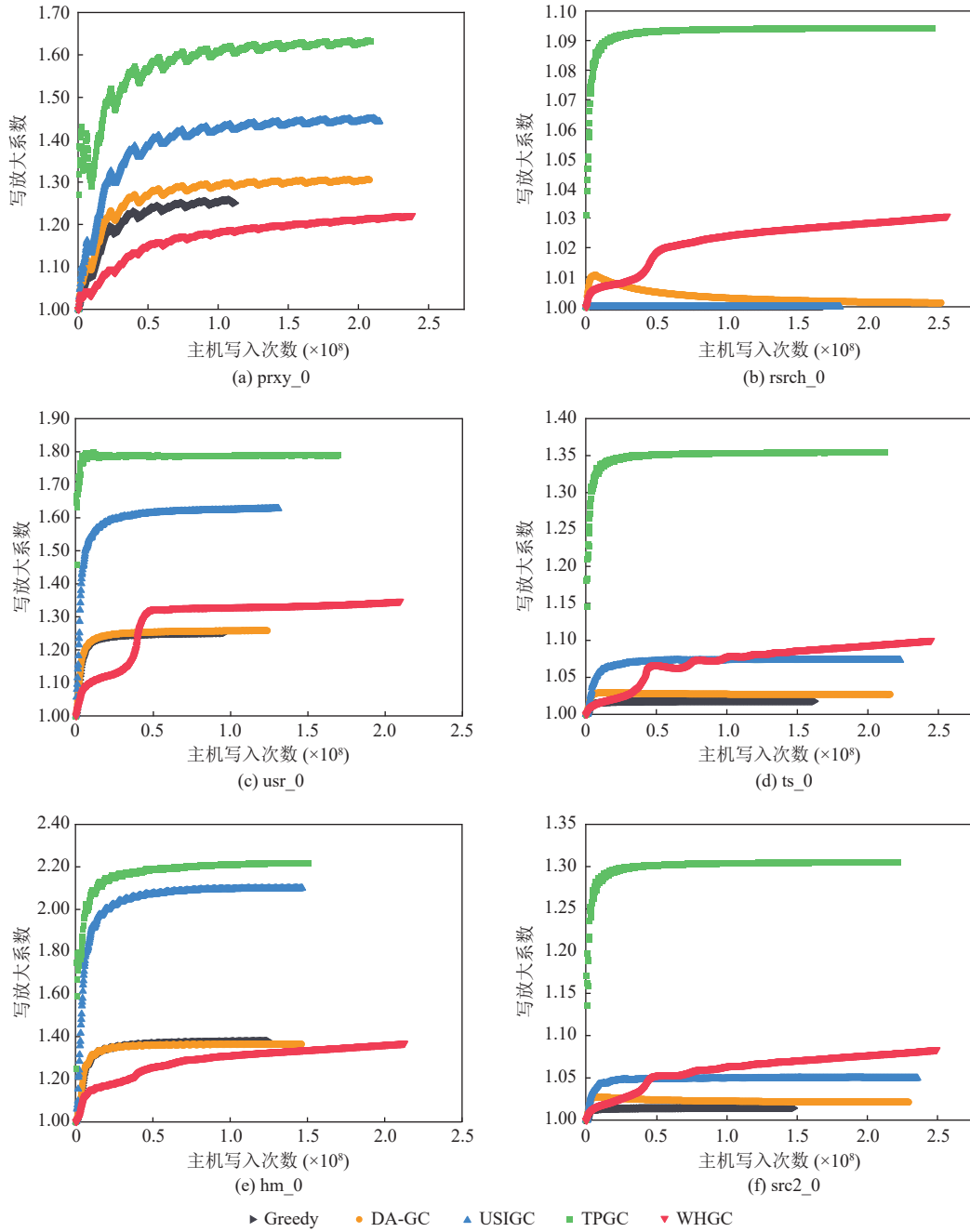


图8 写放大系数变化趋势

从整体趋势来看, 不同方法在各类负载下的写放大表现存在明显差异, 主要源于数据集特征与方法机制的适配性不同. 在图8(a)中, prxy_0 负载中写请求比例高且粒度远小于页大小, 频繁的小文件更新导致有效页迁移难以

避免. WHGC 通过热度识别与等级映射迁移策略, 有效隔离冷热数据, 提升了无效页聚集程度, 显著降低写放大系数, 在该场景下表现最优. 如图 8(b)、(d) 和 (f) 所示, rsrch_0、ts_0 和 src2_0 等负载写入粒度接近页大小, 回收块中常聚集完整无效页, 可直接擦除而无需迁移有效页. 此类场景下, Greedy、DA-GC 和 USIGC 等方法通过优先选择无效页最多的块, 达到较低写放大. TPGC 在固定周期内优先选取擦除次数小的块, 产生大量有效页迁移, 写放大最高. WHGC 虽然因主动保护高磨损块引入一定迁移开销, 导致写放大上升, 但该设计有效提升了磨损均衡性, 可显著延缓坏块出现, 延长 SSD 使用寿命, 且整体写放大始终低于 TPGC, 表现出良好的可靠性与性能平衡能力. 图 8(c) 和 (e) 展示了不同 GC 方法在 usr_0 与 hm_0 两个读写混合型负载下的表现. usr_0 的写入粒度较大, 频繁跨页写入影响热度识别精度, 导致 WHGC 有效页迁移量略高于 Greedy 与 DA-GC, 写放大略有上升. 而 hm_0 的写入粒度接近页大小, 冷热边界清晰, WHGC 热度识别与映射迁移机制得以充分发挥, 写放大低于其他方法.

综上, WHGC 在小粒度频繁更新, 有效页迁移不可避免的场景中能够有效控制写放大; 在写入粒度接近页大小, 无效页易于自然聚集的负载下, 虽因均衡性策略引入部分迁移, 写放大略有增加, 但始终保持在较低水平. 该结果表明 WHGC 能有效感知不同写入模式, 动态平衡写入效率与磨损均衡, 在多种工作负载下均展现出良好的适应性与鲁棒性.

3.3.3 GC 迁移有效页数量

图 9 展示了各 GC 方法在 6 种数据集下的有效页迁移数量随主机写入次数的变化趋势. 横坐标为主机写入次数, 纵坐标为 GC 迁移的有效页数量. 该指标用于衡量垃圾回收过程中系统为保持空间可用性引起的数据迁移开销, 数值越高代表迁移成本越大, 体现出 GC 策略的资源消耗水平与执行效率.

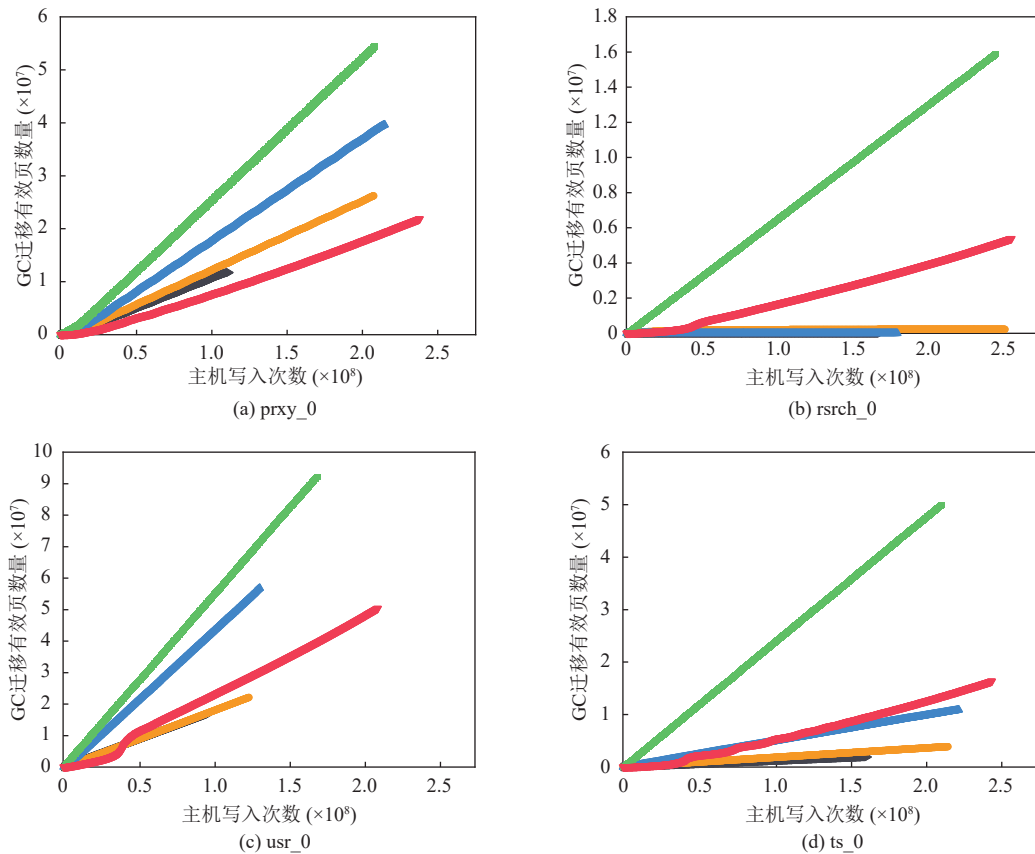


图9 GC 迁移有效页数量变化趋势

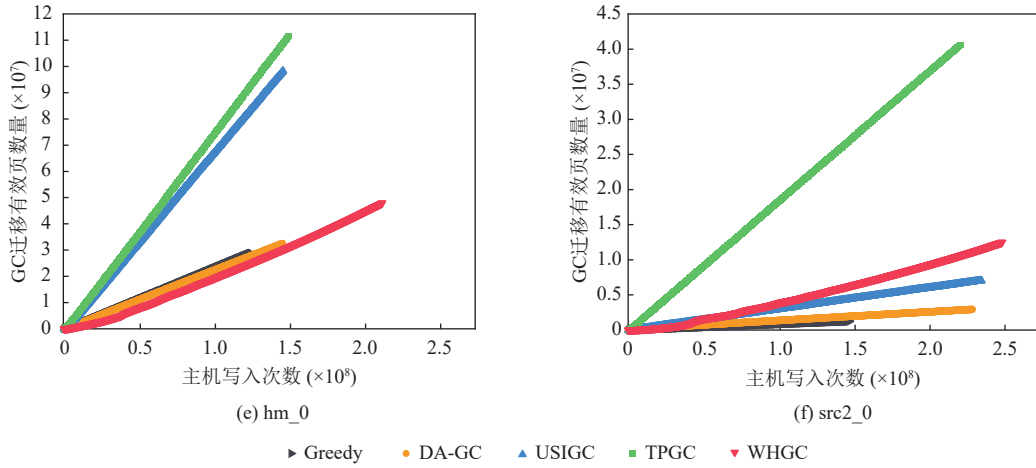


图9 GC迁移有效页数量变化趋势(续)

从整体表现来看,在不同数据集下,各方法的有效页迁移数量排序趋势与图8中写放大系数的排序高度一致。尽管两者在曲线形态上存在差异(写放大呈非线性增长,迁移量为线性累积),但在相对变化上呈现明显的一致性。这种耦合性源于两者在机制层面的本质联系:GC迁移的有效页数量是写放大系数的主要来源之一。结合第3.3.2节写放大系数的分析可知,TPGC为实现更激进的磨损均衡策略,导致其在所有数据集下均产生最多的有效页迁移量;WHGC在rsrch_0、ts_0和src2_0负载中引入适度的主动迁移策略,使得有效页迁移数量略高于Greedy、DA-GC和USIGC,但其显著改善了磨损分布,延长了系统寿命。从写放大系数的对比结果来看,WHGC始终控制在较低水平,体现出其迁移策略在效率与寿命之间实现了更优平衡。

综上,GC迁移有效页数量与写放大系数之间存在高度耦合关系,反映了各方法在空间利用、寿命控制与资源消耗方面的综合设计能力。WHGC在不同类型负载下均表现出稳定、适度的迁移控制策略,验证了其良好的通用适应性与策略鲁棒性。

3.3.4 GC触发频率

图10展示了5个垃圾回收方法在6种数据集下的归一化GC触发频率对比情况。该指标以Greedy为基线,反映单位主机写入量下系统触发GC操作的平均频率,数值越高代表系统在同等写入量下更频繁地执行垃圾回收操作,从而反映出各策略在空间回收方面的调度密度。

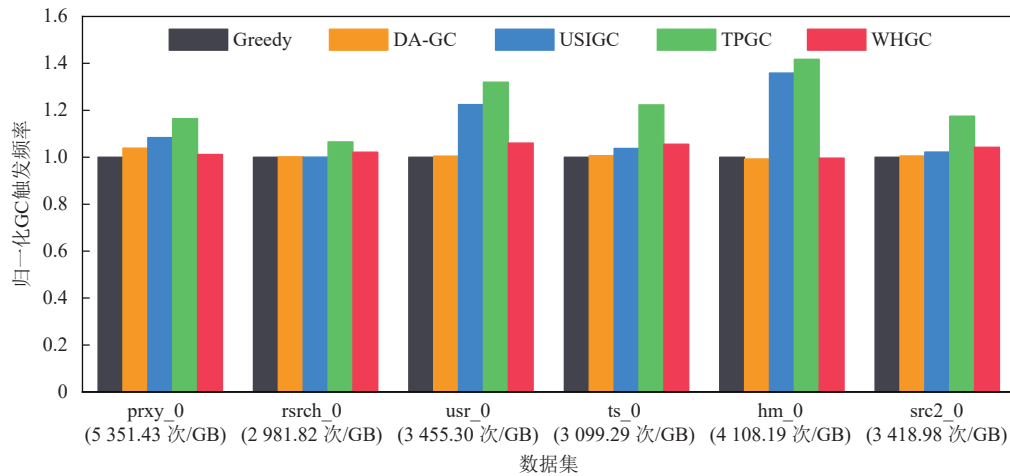


图10 归一化GC触发频率

从整体趋势来看, 该指标与写放大系数和 GC 迁移有效页数量的排序表现一致, 呈现明显的耦合关系: 当 GC 方法为回收更多无效页面而被频繁触发时, 往往也伴随着更多的有效页迁移与更高的写放大. 三者分别体现了 GC 策略在调度频率与迁移成本上的关键特征, 可共同反映其资源消耗与性能开销.

由图 10 可知, TPGC 在所有数据集下均表现出最高的 GC 触发频率, 这与其为维持磨损均衡而频繁回收低磨损块的设计目标一致. USIGC 在部分数据集下维持较高触发密度, 说明其在系统空闲状态下的 IGC 策略被频繁调度执行. 相较而言, WHGC 在不同数据集下的表现更具差异化与自适应性: 在 prxy_0 和 hm_0 中触发频率最低, 这是因为写入模式感知机制能够准确识别高写入强度和集中更新行为, 从而有效避免了不必要的回收操作; 在 usr_0 这类冷热数据混合分布的负载中, WHGC 需积极进行冷热数据分离, 触发频率略高于 Greedy 和 DA-GC, 但仍显著低于 USIGC 与 TPGC; 在 rsrch_0、ts_0 和 src2_0 负载中, WHGC 为缓解块间磨损差异引入适度的主动迁移操作, 从而在控制 GC 触发频率的同时, 兼顾空间利用效率与设备寿命.

综上, GC 触发频率作为衡量策略调度密度的重要指标, 与写放大系数和 GC 迁移有效页数量共同构成对 GC 策略效率的三维评估体系. WHGC 在各类负载下均展现出稳定性与成本控制能力, 验证了其良好的通用性.

3.3.5 擦除次数标准差

图 11 展示了在 6 种数据集下, 5 个 GC 方法的擦除次数标准差随主机写入次数的变化趋势. 横坐标为主机写入次数, 纵坐标为擦除次数标准差. 该指标反映了 SSD 所有块擦除次数的分布离散程度, 标准差越小表示磨损分布越均匀, 能够延缓部分块提前达到擦除上限, 从而提升 SSD 整体使用寿命. 因此, 该指标可从块级角度反映闪存块之间的磨损均衡程度.

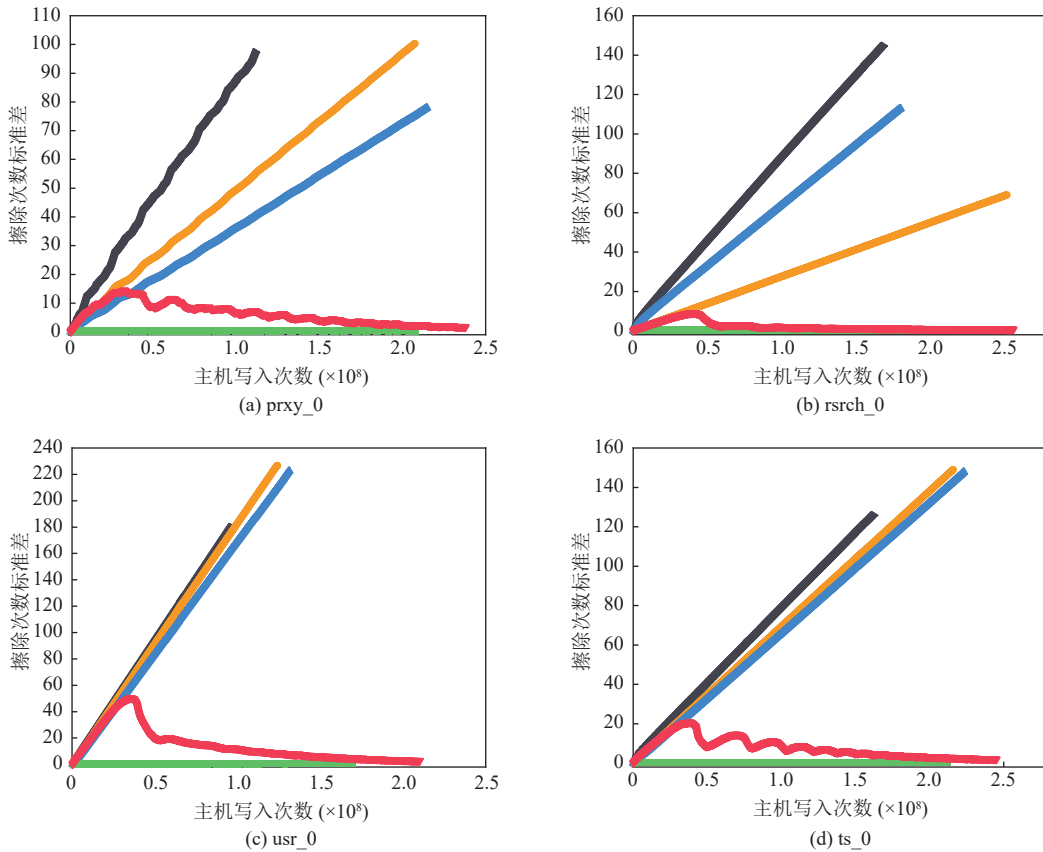


图 11 擦除次数标准差变化趋势

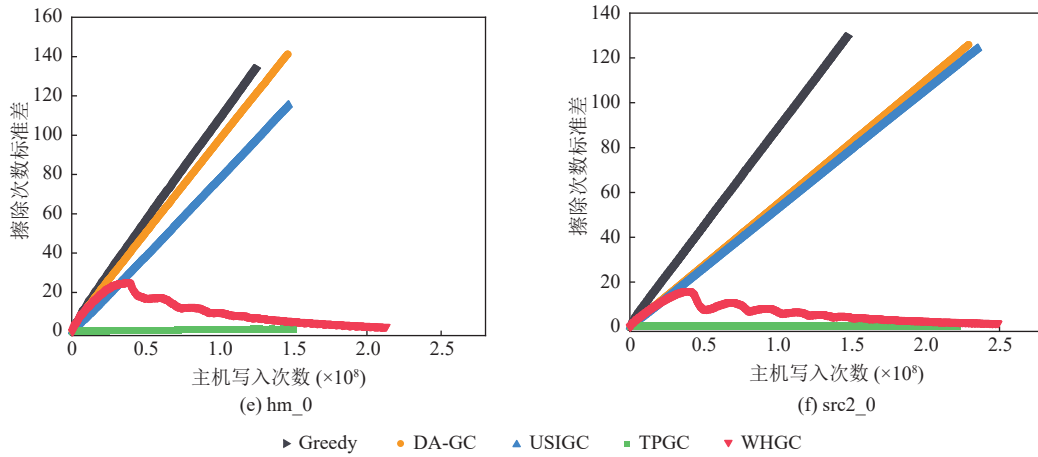


图 11 擦除次数标准差变化趋势 (续)

整体来看, 该指标与主机写入量、写放大系数、GC 迁移有效页数量和 GC 触发频率表现出高度相关性. 当擦除次数标准差较大时, 说明擦除操作在部分块上分布较为集中, 这些块更容易提前达到擦除上限, 从而限制系统能够承载的累计主机写入量. 为缓解磨损集中问题, 系统往往需要更频繁地执行垃圾回收与数据迁移操作, 导致 GC 触发频率与有效页迁移数量增加, 进而提升写放大系数. 相反, 当擦除次数标准差较低时, 说明擦除操作能够在各块之间更加均匀地分布, 从而在不过度增加迁移开销的情况下延缓擦除上限块的出现, 使系统能够支持更高的主机写入量.

从图 11 可以看出, TPGC 在所有负载中均实现了最低的擦除次数标准差, 体现出其磨损均衡导向的策略设计. 然而, 该效果以显著增加写放大系数、有效页迁移量及 GC 触发频率为代价, 最终导致其主机写入量低于 WHGC, 说明其磨损均衡的提升依赖于较高的系统资源开销. Greedy、DA-GC 和 USIGC 在写放大系数、有效页迁移量与 GC 触发频率指标中数值较低, 但擦除次数标准差普遍偏高, 暴露出这些方法在磨损控制方面的不足, 长期运行中, 这类方法易出现磨损集中现象, 加速坏块产生, 限制 SSD 的整体寿命提升空间.

相比之下, WHGC 在 6 种负载中均将擦除次数标准差控制在较低水平, 虽略高于 TPGC, 但显著低于 Greedy、DA-GC 和 USIGC. 虽然在部分场景中增加了有效页迁移与 GC 调度频率, 但整体资源开销显著低于 TPGC. 这表明 WHGC 通过热度识别与分级迁移策略在不同块之间更均衡地分配擦除操作, 有效缓解了磨损集中现象, 从而延缓部分块提前达到擦除上限的情况. 同时, WHGC 在控制磨损均衡的过程中并未引入过高的数据迁移开销, 因此其写放大系数仍保持在较低水平. 因此, WHGC 在磨损均衡与性能开销之间实现了更优的动态平衡, 既保持较低的资源消耗, 又具备良好的稳定性与多负载适应能力. 从而使 SSD 能够支持更高的累计主机写入量, 这从块级磨损分布角度解释了 WHGC 在寿命指标上的优势.

3.3.6 平均响应时间

图 12 展示了在 6 种数据集下, 5 个 GC 方法的平均响应时间, 所有结果均以 Greedy 为基准进行了归一化处理. 横坐标括号内为 Greedy 的实际平均响应时间, 纵坐标为归一化后的平均响应时间, 数值越小表示系统响应越快.

整体来看, TPGC 在所有负载中均表现出最高的平均响应时间, 说明其以磨损均衡为导向的回收策略虽有效, 但也引入了显著的响应延迟开销. 相比之下, WHGC 在 prxy_0 和 hm_0 负载中取得了最优平均响应时间, 在 usr_0 负载中也优于 USIGC 与 TPGC. 在 rsrch_0、ts_0 和 src2_0 负载中, WHGC 的平均响应时间略高于 Greedy、DA-GC 与 USIGC, 主要由于其为增强系统耐久性, 引入了更多的有效页迁移与 GC 调度, 从而对响应延迟造成了一定影响. 但值得强调的是, WHGC 通过在系统空闲时执行 IGC, 有效实现了 GC 操作与主机 I/O 的解耦, 显著缓解了 GC 对正常请求的阻塞. 尽管引入了适度的系统开销, 其平均响应时间仍控制在合理范围内, 未对 I/O 性能造成显著负面影响. 综上所述, WHGC 在保障系统耐久性的同时, 实现了响应性能与磨损控制的协同优化, 体现出良好的实用性与负载适应能力.

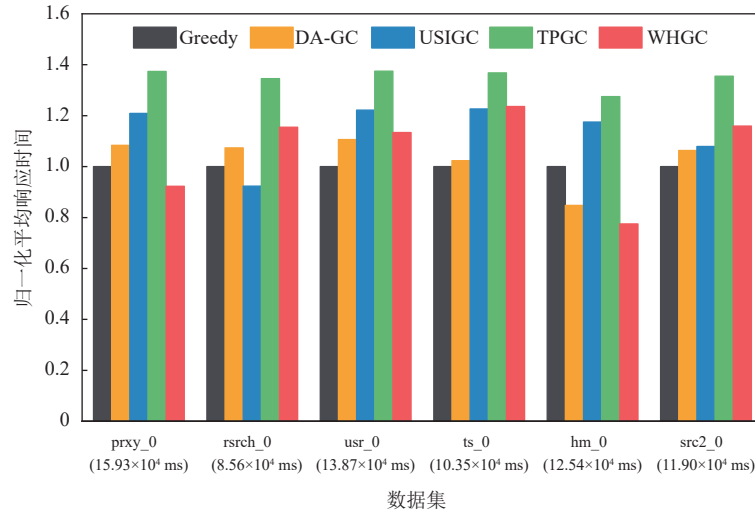


图 12 归一化平均响应时间

3.3.7 时间与空间开销分析

为支持写入模式感知的 GC 触发机制、回收块评分选择以及逻辑页热度分级迁移策略, WHGC 在传统 FTL 结构基础上维护了一定的运行时状态信息, 并在写入路径和 GC 路径中引入少量额外计算. 本节从时间复杂度与空间开销两个方面分析 WHGC 的实现代价, 以评估该方法在实际 SSD 控制器环境中的可行性.

3.3.7.1 时间开销分析

WHGC 方法的时间开销主要来源于写入路径和垃圾回收路径中的相关计算过程. 其中, 写入路径主要涉及逻辑页热度更新与写入模式统计计算, 而垃圾回收路径主要涉及回收块选择与目标块匹配等操作. 其时间开销分析如下.

(1) 写入路径

在主机写入请求到达时, 系统需要根据公式 (9) 和 (10) 更新逻辑页热度值. 该过程主要包括写入序号差值计算、指数衰减运算以及常数次算术操作. 由于每次写入仅涉及当前逻辑页对应的一条热度记录更新, 不需要遍历其他逻辑页或物理块, 因此该过程的时间复杂度为 $O(1)$. 此外, 系统还需要根据公式 (1)–(3) 计算 GC 触发阈值, 并统计写入密度以更新写入模式指标 L . 该过程仅涉及常数次变量更新与简单计算, 同样属于常数时间复杂度. 因此, WHGC 在写入路径上引入的额外计算开销为 $O(1)$, 不会对单次 I/O 请求的处理延迟产生明显影响.

(2) 垃圾回收路径

当系统触发垃圾回收时, WHGC 首先根据公式 (6)–(8) 计算回收块评分, 并在当前分组中的所有块中选择评分最高的块作为回收块. 设当前分组中物理块数量为 n , 则回收块选择过程需要遍历所有块并计算评分值, 其时间复杂度为 $O(n)$, 该复杂度与传统垃圾回收方法在选择回收块时需要遍历所有块的过程一致, 因此不会引入额外复杂度增长.

在有效页迁移阶段, 系统根据热度等级与擦除等级之间的映射关系为回收块中的每个有效页选择目标块. 设回收块中有效页数量为 V , 则目标块选择过程在最坏情况下的复杂度为 $O(V \times n)$, 其中 V 的上界由单个块内页数决定, 属于固定结构参数. 同时, 该过程仅在 GC 触发时执行, 而 GC 触发频率通常低于主机写入请求频率.

因此, WHGC 在写入路径中的额外计算开销为 $O(1)$, 而 GC 阶段的主要复杂度为 $O(n)$, 整体时间复杂度与现有垃圾回收方法保持同一数量级, 不会显著增加 SSD 控制器的运行负担.

3.3.7.2 空间开销分析

WHGC 在传统 FTL 结构基础上引入额外状态信息, 用于支持写入模式感知与热度分级迁移机制, 其新增空间

开销主要来源于以下 3 个部分.

(1) 分组级阈值与写入模式统计信息

为了实现写入模式感知的 GC 触发机制, 系统需要维护若干全局统计变量, 包括主机写请求次数、写入数据量、写入密度值、归一化写入模式指标以及动态 GC 触发阈值等统计量. 这类变量均属于系统级统计信息, 其数量为常数级规模, 与 SSD 容量和逻辑页数量无关, 因此其空间复杂度为 $O(1)$, 对系统整体内存占用影响较小.

(2) 块级状态信息

在回收块选择阶段, WHGC 需要利用块的擦除次数、无效页比例以及平均擦除次数等信息计算评分函数, 以确定回收块. 这些块级状态信息在多数 FTL 实现中已经作为基本状态维护, 用于支持空间管理和磨损统计, 因此 WHGC 仅增加少量辅助统计变量, 其额外空间开销仍属于常数级规模.

(3) 逻辑页热度表

WHGC 新增空间开销的主要来源是逻辑页热度表. 系统为每个逻辑页维护热度信息, 用于描述该页的更新频率和访问活跃度. 每个热度记录需要存储逻辑页当前热度值以及最近一次写入序号. 若热度值采用 32 位浮点数表示, 写入序号采用 32 位整数表示, 则每个逻辑页约需要 8 字节存储空间. 设 SSD 用户容量为 C , 页大小为 P , 则逻辑页数量为 $N_{LPN} = C/P$, 因此热度表总体空间开销约为 $M_{Heat} = N_{LPN} \times 8 \text{ B}$. 以容量为 32 GB、页大小为 4 KB 的 SSD 为例, 逻辑页数量约为 8.39×10^6 , 对应热度表额外开销约为 $8.39 \times 10^6 \times 8 \text{ B} \approx 64 \text{ MB}$. 在基于页映射的 FTL 结构中, 地址映射表通常需要为每个逻辑页维护 4~8 B 的映射信息, 其整体规模通常在数十 MB 级别, WHGC 新增的热度表开销与映射表处于相同数量级. 在当前 SSD 控制器设计中, 控制器通常配备用于维护映射表和运行时状态信息的专用 DRAM 缓存, 因此该规模的额外空间开销在工程实现中属于可接受范围. 此外, 在资源受限的系统中, 还可以通过定点化表示或逻辑区间级聚合等方式进一步降低空间占用.

因此, WHGC 所引入的额外空间开销主要来源于逻辑页热度表, 其规模与逻辑页数量呈线性关系, 而其余统计信息均为常数级开销, 整体空间开销处于可控范围.

3.4 不同容量配置下的性能分析

为评估 WHGC 方法在不同 SSD 容量配置下的适用性与稳定性, 本节在主实验 4 GB 容量配置基础上增加 32 GB 容量配置进行对比实验. 在保持其余系统参数一致的前提下, 将 SSD 容量扩展为 32 GB, 并将擦除上限设置为 1000 次. 为使系统能够更快进入垃圾回收阶段, 32 GB 容量实验采用 80% 比例的预填数据进行初始化. 较小容量由于系统中可用空闲块数量有限, 更容易触发垃圾回收行为, 从而能够更加清晰地观察不同 GC 策略之间的性能差异; 而较大容量配置则能够反映更大规模 SSD 的运行特征. 因此, 通过在 4 GB 与 32 GB 两种容量配置下进行对比分析, 可以评估不同容量条件下各方法性能变化趋势的一致性. 本节选取具有代表性的工作负载 prxy_0 作为实验数据集, 考虑到 Greedy 方法是垃圾回收研究中广泛采用的经典基线策略, 且被大量研究工作作为对比基准, 本节选取 Greedy 作为对比方法, 重点分析 WHGC 在不同容量配置下的性能变化趋势. 实验选取写放大系数与擦除次数标准差两个指标, 分别反映系统额外写入开销与磨损均衡程度. 上述两个指标是评价垃圾回收策略效率与磨损均衡能力的典型指标, 能够反映 GC 策略在不同容量配置下的关键行为特征.

图 13(a) 展示了在 4 GB 与 32 GB 两种容量配置下 Greedy 与 WHGC 方法的写放大系数随主机写入次数变化的趋势, 其中横坐标为主机写入次数, 纵坐标为写放大系数. 当 SSD 容量由 4 GB 扩展至 32 GB 后, 系统可用空闲块数量增加, 垃圾回收触发频率相应降低, 因此写放大系数整体变化趋势更加平缓. 尽管不同容量配置下曲线的变化斜率存在差异, 但两种容量下方法之间的相对关系保持一致: WHGC 在整个运行阶段均表现出低于 Greedy 的写放大水平. 由于实验在系统首次出现擦除上限块时终止, 曲线在横坐标方向的延伸范围反映了系统可支持的主机写入规模. 实验结果表明, WHGC 在两种容量配置下对应的主机写入量分别约为 753 GB 和 1261 GB, 均显著高于 Greedy 方法在两种容量配置下的 358 GB 和 623 GB, 表明 WHGC 能够在不同容量条件下有效降低额外写入开销并延缓系统失效.

图 13(b) 展示了不同容量配置下两种方法的擦除次数标准差变化趋势, 其中横坐标为主机写入次数, 纵坐标

为擦除次数标准差. 随着主机写入次数增加, 各方法的擦除次数标准差均逐渐增大, 但增长速率存在明显差异. Greedy 方法由于缺乏磨损感知机制, 容易导致部分块被频繁擦除, 从而产生明显的磨损集中现象; 相比之下, WHGC 在回收块选择与有效页迁移过程中综合考虑块磨损状态与数据热度分布, 能够有效分散擦除操作, 使块间磨损分布更加均匀. 尽管在不同容量配置下 WHGC 曲线的变化过程存在一定差异, 但其总体表现一致, 即 WHGC 始终维持较低的擦除次数标准差水平, 表明该方法能够在不同容量条件下有效缓解块间磨损集中问题.

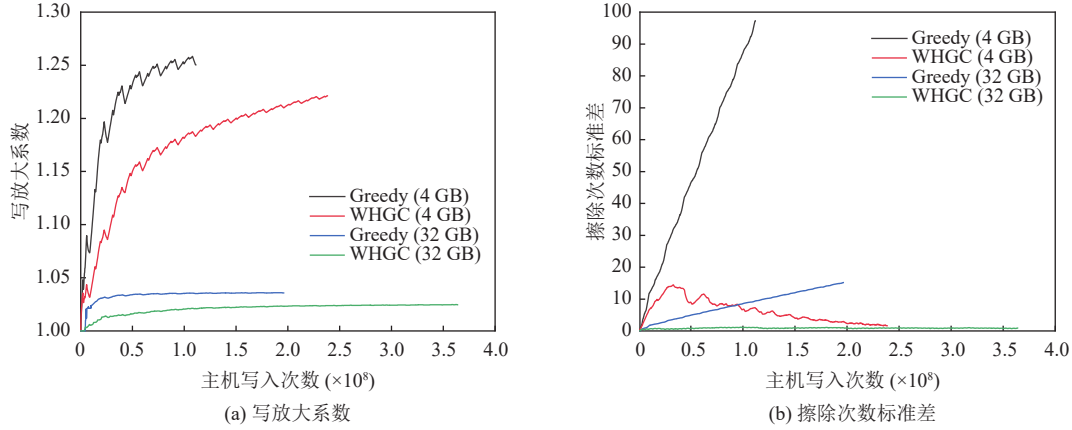


图 13 不同容量配置下的性能变化趋势

尽管 SSD 容量及擦除上限等系统配置变化会对垃圾回收行为产生一定影响, 但 WHGC 在写放大开销与块间磨损均衡两个关键方面均保持稳定表现, 表明该方法的优势主要来源于其机制设计本身, 而非特定系统配置所带来的偶然影响.

3.5 消融实验结果分析

为进一步验证 WHGC 中各关键机制的有效性与独立贡献, 本文设计了两组消融实验, 分别评估写入模式感知阈值机制与热度分级迁移机制对系统性能的影响. 通过在保留其余模块不变的前提下, 逐一屏蔽上述关键机制, 并与完整 WHGC 方法进行对比, 可揭示各模块在控制 GC 触发频率、降低写放大系数与有效页迁移开销等方面的具体作用, 从而验证其设计的合理性与必要性. 本组实验基于 SSDsim 平台进行, 实验配置与表 1 一致. 完整 WHGC 方法采用与主实验相同的参数设置. 所使用的数据集为 prxy_0 负载, 该数据具有高写入比例与频繁更新特征, 能够显著呈现不同机制对系统性能的影响.

3.5.1 动态阈值机制的影响分析

为评估 WHGC 中写入模式感知阈值机制对垃圾回收效率与系统开销的影响, 本节设置对比方案 WHGC (Fixed TH), 将 GC 触发阈值设定为固定常数, 取消动态调节机制. 除触发机制外, 其余配置与完整 WHGC 方法 WHGC (Full) 保持一致.

图 14 展示了两种方案在 prxy_0 负载下的实验结果, 图 14(a) 为 GC 触发次数随主机写入次数的变化趋势, 横坐标为主机写入次数, 纵坐标为 GC 触发次数. 图 14(b) 为写放大系数随主机写入次数的变化趋势, 横坐标为主机写入次数, 纵坐标为写放大系数. 如图 14(a) 所示, 在相同主机写入次数下, WHGC (Fixed TH) 的 GC 触发次数高于 WHGC (Full), 表明固定阈值机制无法根据系统负载状态灵活调整触发时机, 易造成 GC 频繁执行. 而图 14(b) 显示其写放大系数也高于完整方法 WHGC (Full), 反映出频繁 GC 所带来的额外页迁移增加了系统写入开销. 相比之下, WHGC (Full) 通过写入模式感知机制, 能够根据 I/O 写入强度自适应调整 GC 触发阈值, 有效降低了不必要的 GC 执行次数, 优化了 GC 调度时机, 在保证系统空间可用性的同时减少了资源浪费, 并增加了主机写入量, 体现出更优的 GC 效率.

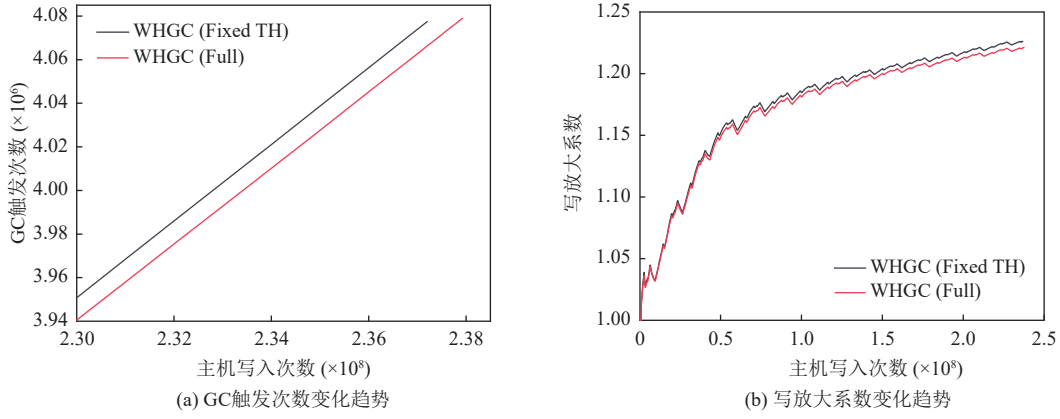


图 14 阈值机制消融实验结果

3.5.2 热度分级机制的影响分析

为验证 WHGC 中热度分级迁移机制的有效性, 本节设置对比方案 WHGC (No Heat), 将热度等级数 n 设置为 1, 使所有有效页热度视为一致, 取消热度与擦除等级之间的匹配策略. 此时迁移策略退化为“直接将有效页迁移至擦除次数最小的空闲块”. 与之对比的完整方法为 WHGC (Full), 采用热度识别、分级与擦除等级映射策略.

图 15 展示了热度分级机制的消融实验结果, 图 15(a) 为 GC 迁移有效页数量随主机写入次数的变化趋势, 横坐标为主机写入次数, 纵坐标为 GC 迁移有效页数量. 图 15(b) 为写放大系数随主机写入次数的变化趋势, 横坐标为主机写入次数, 纵坐标为写放大系数. 如图 15(a) 所示, WHGC (No Heat) 方案中, 由于缺乏热度分级与目标块匹配机制, 导致大量有效页被频繁迁移, 造成迁移页数量显著增加, 同时写放大系数也相应升高, 如图 15(b) 所示. 相比之下, WHGC (Full) 通过逻辑页热度识别与擦除等级匹配迁移, 提升了无效页聚集度, 降低了有效页迁移量, 从而在降低了 GC 迁移负担的同时, 有效控制了写放大. 实验结果表明, 热度分级机制对于提升 GC 效率、控制写放大具有显著作用, 在系统长期运行中展现出良好的性能优化潜力.

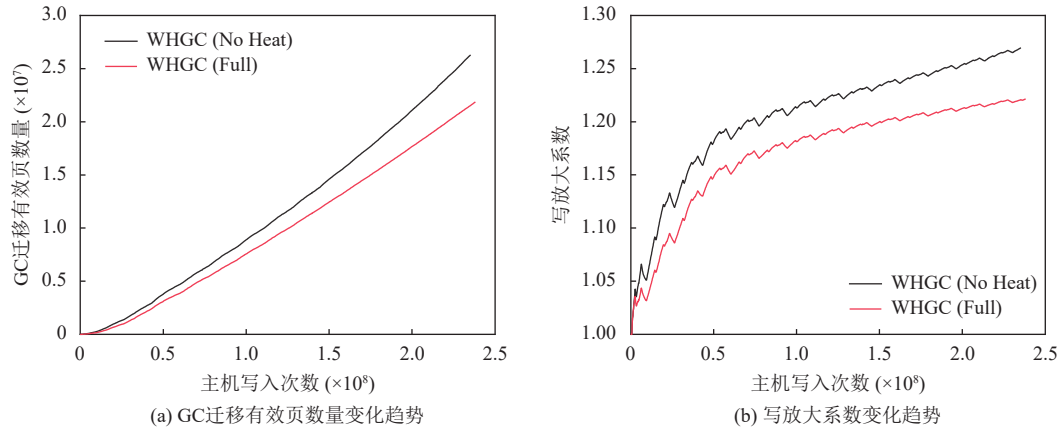


图 15 热度分级机制消融实验结果

3.6 参数影响分析

为进一步评估 WHGC 方法在热度相关参数设置下的鲁棒性与可调节性, 本文选取两项核心参数进行分析. 1) 热度等级划分数 n : 控制逻辑页热度与目标块擦除等级之间的匹配精度, 影响有效页迁移的分级灵活性; 2) 热度更新的指数衰减因子 γ : 决定逻辑页热度对历史访问的记忆深度, 反映热度更新对近期写入的敏感程度. 本组实验

沿用第 3.4 节消融实验的配置环境与数据集设置, 通过对比不同参数取值组合下的 GC 迁移有效页数量与写放大系数, 评估 WHGC 的参数调节空间与性能稳定性。

3.6.1 热度等级数 n 的影响分析

将热度等级数 n 分别设置为 2、4、6、8 进行实验, 图 16 展示了 n 为不同数值下系统的 GC 有效页迁移数量与写放大系数变化趋势。图 16(a) 为 GC 迁移有效页数量随主机写入次数的变化趋势, 横坐标为主机写入次数, 纵坐标为 GC 迁移有效页数量。图 16(b) 为写放大系数随主机写入次数的变化趋势, 横坐标为主机写入次数, 纵坐标为写放大系数。由图 16(a) 可知, 当 $n=2$ 时由于热度划分过于粗糙, 冷热页区分能力不足, 迁移策略难以发挥作用, 导致有效页迁移量明显上升, 同样写放大系数因此升高, 如图 16(b) 所示。随着 n 的提升, 系统在有效页热度识别与块擦除等级的映射策略中具备更高的匹配准确性, GC 迁移效率提升, 写放大显著降低。当 n 增至 6 和 8 时, 性能提升逐渐趋缓, 等级划分过细导致热度区分不明显, 因此 $n=6$ 和 $n=8$ 的有效页迁移量与写放大系数变化趋势基本一致。鉴于不同负载的数据访问特性差异较大, 热度等级数 n 作为 WHGC 方法中重要的调节参数, 可根据实际应用需求进行灵活配置, 以兼顾性能表现与系统开销。

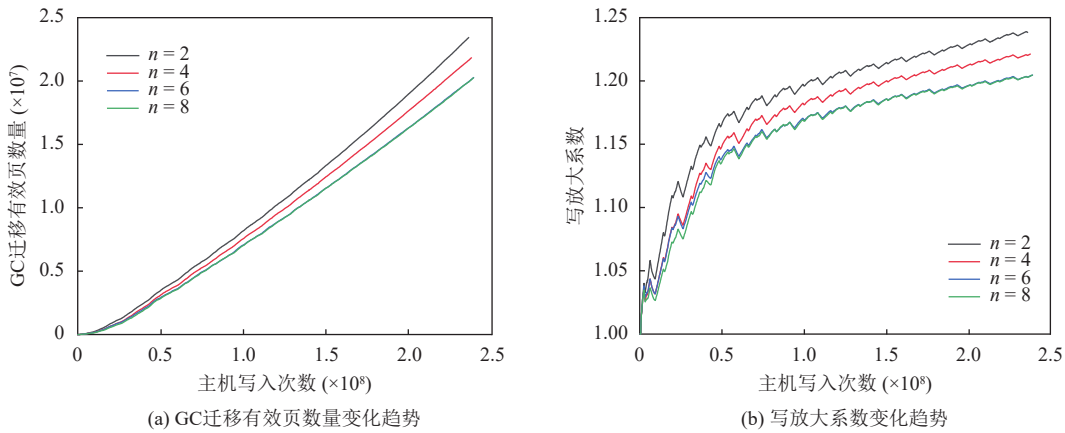
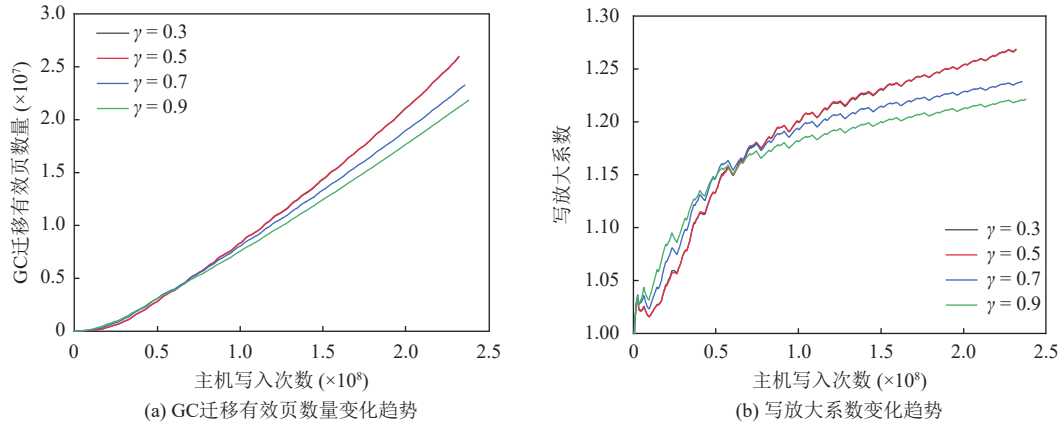


图 16 不同热度等级划分数 n 对系统性能的影响

3.6.2 热度衰减系数 γ 的影响分析

将热度指数衰减因子 γ 设置为 0.3、0.5、0.7、0.9, 评估其对 WHGC 方法的性能影响, 实验结果如图 17 所示。图 17 展示了不同 γ 值下系统的 GC 有效页迁移数量与写放大系数变化趋势。图 17(a) 为 GC 迁移有效页数量随主机写入次数的变化趋势, 横坐标为主机写入次数, 纵坐标为 GC 迁移有效页数量。图 17(b) 为写放大系数随主机写入次数的变化趋势, 横坐标为主机写入次数, 纵坐标为写放大系数。从图 17 可观察到, 当 $\gamma=0.3$ 和 $\gamma=0.5$ 时, 热度衰减速度较快, 系统更关注近期的写入行为, 但由于热度记忆窗口较短, 历史热度保留不足, 难以稳定捕捉逻辑页的整体访问趋势, 导致冷热页识别不稳定, 有效页迁移量与写放大系数较高, 二者在整体趋势上几乎重合。随着 γ 的增加, 热度更新对历史写入的保留程度增强, 使热度识别更具连续性与稳定性, 能够更准确地区分冷热数据, 从而降低无效迁移, 减少写放大, 提升 GC 回收效率。控制因子 γ 决定了热度值的衰减速度: γ 越小, 历史热度衰减越快, 系统更侧重于近期访问; γ 越大, 热度更新更具记忆深度, 更关注长期访问趋势。因此, 该参数的设定应综合考虑负载特性, 合理平衡响应灵敏度与热度稳定性, 以获得更优的识别效果与整体性能。

综合第 3.6.1 与 3.6.2 节的实验结果可以看出, WHGC 方法在不同参数取值范围下均保持稳定的性能变化趋势。当热度等级数或热度衰减系数在合理范围内变化时, 系统的 GC 迁移有效页数量与写放大系数虽然存在一定差异, 但整体趋势保持一致, 且性能变化幅度较为平稳。这表明 WHGC 方法对关键参数设置具有较好的鲁棒性, 无需对参数进行精细调节即可获得稳定的性能优化效果, 从而增强了方法在不同负载环境下的适用性。

图 17 不同热度指数衰减因子 γ 对系统性能的影响

4 总结

本文提出了一种写入模式感知与热度分级的垃圾回收方法 WHGC, 旨在提升 GC 执行效率并延长 SSD 使用寿命. 该方法通过设计写入模式感知的 GC 触发阈值机制, 可根据主机负载状态动态调整回收时机, 避免 GC 操作的过早或滞后触发; 引入指数衰减型逻辑页热度识别策略, 准确反映数据的写入活跃度, 实现冷热数据的有效区分; 并通过热度等级与擦除等级映射的有效页迁移策略, 实现数据热度与物理块磨损状态的合理匹配, 提升了有效页迁移效率与空间利用率. 实验结果表明, 相较于对比方法, WHGC 的主机写入量平均至少提升了 18.88%, 并在写放大系数、GC 触发频率、擦除次数标准差等多个关键指标上均表现出较优性能, 验证了所提方法的有效性与实用价值. 未来工作将尝试引入机器学习技术进一步提升数据热度识别的准确性, 并探索该方法与文件系统层的协同机制, 通过融合数据语义与生命周期信息, 优化 GC 决策流程, 进一步增强 SSD 的可靠性与智能化管理能力.

References

- [1] Yadgar G, Gabel M, Jaffer S, Schroeder B. SSD-based workload characteristics and their performance implications. *ACM Trans. on Storage*, 2021, 17(1): 1–26. [doi: [10.1145/3423137](https://doi.org/10.1145/3423137)]
- [2] Yang MC, Chang YM, Tsao CW, Huang PC, Chang YH, Kuo TW. Garbage collection and wear leveling for flash memory: Past and future. In: *Proc. of the 2014 Int'l Conf. on Smart Computing*. Hong Kong: IEEE, 2014. 66–73. [doi: [10.1109/SMARTCOMP.2014.7043841](https://doi.org/10.1109/SMARTCOMP.2014.7043841)]
- [3] Mao B, Wu SZ, Duan LD. Improving the SSD performance by exploiting request characteristics and internal parallelism. *IEEE Trans. on Computer-aided Design of Integrated Circuits and Systems*, 2018, 37(2): 472–484. [doi: [10.1109/TCAD.2017.2697961](https://doi.org/10.1109/TCAD.2017.2697961)]
- [4] Zhang XQ, Bhimani J, Pei SY, Lee E, Lee S, Seong YJ, Kim EJ, Choi C, Nam EH, Choi J, Kim BS. Storage abstractions for SSDs: The past, present, and future. *ACM Trans. on Storage*, 2025, 21(1): 2. [doi: [10.1145/3708992](https://doi.org/10.1145/3708992)]
- [5] Wu M, Zwaenepoel W. eNVy: A non-volatile, main memory storage system. *ACM SIGOPS Operating Systems Review*, 1994, 28(5): 86–97. [doi: [10.1145/381792.195506](https://doi.org/10.1145/381792.195506)]
- [6] Kawaguchi A, Nishioka S, Motoda H. A flash-memory based file system. In: *Proc. of the 1995 USENIX Technical Conf.* New Orleans: USENIX Association, 1995.
- [7] Chiang ML, Chang RC. Cleaning policies in mobile computers using flash memory. *Journal of Systems and Software*, 1999, 48(3): 213–231. [doi: [10.1016/S0164-1212\(99\)00059-X](https://doi.org/10.1016/S0164-1212(99)00059-X)]
- [8] Yan H, Yao Q. An efficient file-aware garbage collection algorithm for NAND flash-based consumer electronics. *IEEE Trans. on Consumer Electronics*, 2014, 60(4): 623–627. [doi: [10.1109/TCE.2014.7027335](https://doi.org/10.1109/TCE.2014.7027335)]
- [9] Yu FF, Yan H. An efficient hot-cold data separation garbage collection algorithm based on logical interval in NAND flash-based consumer electronics. *IEEE Trans. on Consumer Electronics*, 2023, 69(3): 431–440. [doi: [10.1109/TCE.2022.3228404](https://doi.org/10.1109/TCE.2022.3228404)]
- [10] Chen Z, Zhao YL. DA-GC: A dynamic adjustment garbage collection method considering wear-leveling for SSD. In: *Proc. of the 2020*

- Great Lakes Symp. on VLSI. ACM, 2020. 475–480. [doi: [10.1145/3386263.3406943](https://doi.org/10.1145/3386263.3406943)]
- [11] Yao YB, Kong XC, Bao JC, Xu X, Gu NH, Feng W. Uniform scheduling of interruptible garbage collection and request IO to improve performance and wear-leveling of SSDs. *The Journal of Supercomputing*, 2022, 78(7): 9691–9710. [doi: [10.1007/s11227-021-04294-2](https://doi.org/10.1007/s11227-021-04294-2)]
 - [12] Cheng W, Luo M, Zeng LF, Wang Y, Brinkmann A. Lifespan-based garbage collection to improve SSD's reliability and performance. *Journal of Parallel and Distributed Computing*, 2022, 164: 28–39. [doi: [10.1016/j.jpdc.2022.02.006](https://doi.org/10.1016/j.jpdc.2022.02.006)]
 - [13] Li ZY, Zhang GY, Wang Y. Flash-oriented coded storage: Research status and future directions. *ACM Trans. on Storage*, 2025, 21(1): 5. [doi: [10.1145/3708995](https://doi.org/10.1145/3708995)]
 - [14] Gao CM, Shi L, Liu K, Xue C, Shu JW. Architecture and technologies of flash memory based solid state drives. *Journal of Computer Research and Development*, 2021, 58(7): 1518–1532 (in Chinese with English abstract). [doi: [10.7544/issn1000-1239.2021.20200690](https://doi.org/10.7544/issn1000-1239.2021.20200690)]
 - [15] Luo YH, Lin MW. Flash translation layer: A review and bibliometric analysis. *Int'l Journal of Intelligent Computing and Cybernetics*, 2021, 14(3): 480–508. [doi: [10.1108/ijicc-02-2021-0034](https://doi.org/10.1108/ijicc-02-2021-0034)]
 - [16] van Houdt B. A mean field model for a class of garbage collection algorithms in flash-based solid state drives. In: *Proc. of the 2013 ACM SIGMETRICS/Int'l Conf. on Measurement and Modeling of Computer Systems*. Pittsburgh: ACM, 2013. 191–202. [doi: [10.1145/2465529.2465543](https://doi.org/10.1145/2465529.2465543)]
 - [17] Kwon O, Koh K, Lee J, Bahn H. FeGC: An efficient garbage collection scheme for flash memory based storage systems. *Journal of Systems and Software*, 2011, 84(9): 1507–1523. [doi: [10.1016/j.jss.2011.02.042](https://doi.org/10.1016/j.jss.2011.02.042)]
 - [18] Tjioe J, Blanco A, Xie T, Ouyang YM. Making garbage collection wear conscious for flash SSD. In: *Proc. of the 7th IEEE Int'l Conf. on Networking, Architecture, and Storage*. Xiamen: IEEE, 2012. 114–123. [doi: [10.1109/NAS.2012.20](https://doi.org/10.1109/NAS.2012.20)]
 - [19] Yan H, Huang Y, Zhou XZ, Lei YJ. An efficient and non-time-sensitive file-aware garbage collection algorithm for NAND flash-based consumer electronics. *IEEE Trans. on Consumer Electronics*, 2019, 65(1): 73–79. [doi: [10.1109/TCE.2018.2885102](https://doi.org/10.1109/TCE.2018.2885102)]
 - [20] Shahidi N, Kandemir MT, Arjomand M, Das CR, Jung M, Sivasubramaniam A. Exploring the potentials of parallel garbage collection in SSDs for enterprise storage systems. In: *Proc. of the 2016 Int'l Conf. for High Performance Computing, Networking, Storage and Analysis*. Salt Lake City: IEEE, 2016. 561–572. [doi: [10.1109/SC.2016.47](https://doi.org/10.1109/SC.2016.47)]
 - [21] Yang J, Pei SY. Thermo-GC: Reducing write amplification by tagging migrated pages during garbage collection. In: *Proc. of the 2019 IEEE Int'l Conf. on Networking, Architecture and Storage*. Enshi: IEEE, 2019. 1–8. [doi: [10.1109/NAS.2019.8834722](https://doi.org/10.1109/NAS.2019.8834722)]
 - [22] Qin Y, Feng D, Liu JN, Tong W, Zhu ZM. DT-GC: Adaptive garbage collection with dynamic thresholds for SSDs. In: *Proc. of the 2014 Int'l Conf. on Cloud Computing and Big Data*. Wuhan: IEEE, 2014. 182–188. [doi: [10.1109/CCBD.2014.28](https://doi.org/10.1109/CCBD.2014.28)]
 - [23] Lee J, Kim Y, Shipman GM, Oral S, Wang FY, Kim J. A semi-preemptive garbage collector for solid state drives. In: *Proc. of the 2011 IEEE Int'l Symp. on Performance Analysis of Systems and Software*. Austin: IEEE, 2011. 12–21. [doi: [10.1109/ISPASS.2011.5762711](https://doi.org/10.1109/ISPASS.2011.5762711)]
 - [24] Lee D, Jin Y, Jang H, Lee B. CATDOG: Cost-age-time data organized garbage collection. In: *Proc. of the 2022 IEEE Int'l Conf. on Networking, Architecture and Storage (NAS)*. Philadelphia: IEEE, 2022. 1–6. [doi: [10.1109/NAS55553.2022.9925402](https://doi.org/10.1109/NAS55553.2022.9925402)]
 - [25] Sun DS, Song YL, Chai YP, Peng BL, Lu FZ, Deng X. Light-GC: a lightweight and efficient garbage collection scheme for embedded file systems. In: *Proc. of the 23rd ACM/IFIP Int'l Middleware Conf.* Quebec: ACM, 2022. 216–227. [doi: [10.1145/3528535.3565246](https://doi.org/10.1145/3528535.3565246)]
 - [26] Salkhordeh R, Kremer K, Nagel L, Maisenbacher D, Holmberg H, Björling M, Brinkmann A. Constant time garbage collection in SSDs. In: *Proc. of the 2021 IEEE Int'l Conf. on Networking, Architecture and Storage (NAS)*. Riverside: IEEE, 2021. 1–9. [doi: [10.1109/NAS51552.2021.9605386](https://doi.org/10.1109/NAS51552.2021.9605386)]
 - [27] Lin WH, Chang LP. Dual Greedy: Adaptive garbage collection for page-mapping solid-state disks. In: *Proc. of the 2012 Design, Automation & Test in Europe Conf. & Exhibition (DATE)*. Dresden: IEEE, 2012. 117–122. [doi: [10.1109/DATE.2012.6176443](https://doi.org/10.1109/DATE.2012.6176443)]
 - [28] Hwang SH, Choi JH, Kwak JW. Migration cost sensitive garbage collection technique for non-volatile memory systems. *IEICE Trans. on Information and Systems*, 2016, E99. D(12): 3177–3180. [doi: [10.1587/transinf.2016EDL8131](https://doi.org/10.1587/transinf.2016EDL8131)]
 - [29] Wu SZ, Du CF, Li HJ, Jiang H, Shen ZR, Mao B. CAGC: A content-aware garbage collection scheme for ultra-low latency flash-based SSDs. In: *Proc. of the 2021 IEEE Int'l Parallel and Distributed Processing Symp. (IPDPS)*. Portland: IEEE, 2021. 162–171. [doi: [10.1109/IPDPS49936.2021.00025](https://doi.org/10.1109/IPDPS49936.2021.00025)]
 - [30] Wang KY, Tan HL, He ZH, Li JY, Li KQ. CDA-GC: An effective cache data allocation for garbage collection in flash-based solid-state drives. *Integration*, 2025, 102: 102359. [doi: [10.1016/j.vlsi.2025.102359](https://doi.org/10.1016/j.vlsi.2025.102359)]
 - [31] Wang QP, Li JH, Lee PPC, Ouyang T, Shi C, Huang LL. Separating data via block invalidation time inference for write amplification reduction in log-structured storage. In: *Proc. of the 20th USENIX Conf. on File and Storage Technologies*. Santa Clara: USENIX Association, 2022. 429–444.
 - [32] Kim J, Lee S, Oh J, Shin D, Won Y. D2FS: Device-driven filesystem garbage collection. In: *Proc. of the 23rd USENIX Conf. on File and Storage Technologies*. Santa Clara: USENIX Association, 2025. 337–353.

- [33] Lange T, Naor J, Yadgar G. Optimal SSD management with predictions. *ACM SIGMETRICS Performance Evaluation Review*, 2025, 53(1): 52–54. [doi: [10.1145/3744970.3727303](https://doi.org/10.1145/3744970.3727303)]
- [34] Zhou Q, Peng SR, Shen TR, Yin J, Sun TL, Xue XY. Garbage collection optimization with data separation for large data storage in deep learning applications. *Microelectronics Journal*, 2025, 158: 106620. [doi: [10.1016/j.mejo.2025.106620](https://doi.org/10.1016/j.mejo.2025.106620)]
- [35] Yang P, Xue N, Zhang YQ, Zhou YX, Sun L, Chen WW, Chen ZG, Xia W, Li JK, Kwon K. Reducing garbage collection overhead in SSD based on workload prediction. In: *Proc. of the 11th USENIX Workshop on Hot Topics in Storage and File Systems*. Renton: USENIX Association, 2019.
- [36] Liu LH, Gu YF, Zhu CH, Wu CT, Li J, Guo MY. GCC: Optimizing space efficiency and read latency of SSDs with workload-aware garbage collection aided compression. In: *Proc. of the 42nd IEEE Int'l Conf. on Computer Design (ICCD)*. Milan: IEEE, 2024. 332–339. [doi: [10.1109/ICCD63220.2024.00058](https://doi.org/10.1109/ICCD63220.2024.00058)]
- [37] Luo QW, Fang XX, Sun YC, Ai JQ, Yang CH. Self-learning hot data prediction: Where echo state network meets NAND flash memories. *IEEE Trans. on Circuits and Systems I: Regular Papers*, 2020, 67(3): 939–950. [doi: [10.1109/TCSI.2019.2960015](https://doi.org/10.1109/TCSI.2019.2960015)]
- [38] Sun PH, You LT, Zheng SG, Zhang WR, Ma RY, Yang J, Wang GZ, Zhu F, Li S, Huang LP. Learning-based data separation for write amplification reduction in solid state drives. In: *Proc. of the 60th Annual ACM/IEEE Design Automation Conf.* San Francisco: IEEE, 2023. 1–6. [doi: [10.1109/DAC56929.2023.10247795](https://doi.org/10.1109/DAC56929.2023.10247795)]
- [39] Hu Y, Jiang H, Feng D, Tian L, Luo H, Ren C. Exploring and exploiting the multilevel parallelism inside SSDs for improved performance and endurance. *IEEE Trans. on Computers*, 2013, 62(6): 1141–1155. [doi: [10.1109/TC.2012.60](https://doi.org/10.1109/TC.2012.60)]
- [40] Wu JJ, Li J, Sha ZB, Cai ZG, Liao JW. Adaptive switch on wear leveling for enhancing I/O latency and lifetime of high-density SSDs. *IEEE Trans. on Computer-aided Design of Integrated Circuits and Systems*, 2022, 41(11): 4040–4051. [doi: [10.1109/TCAD.2022.3197340](https://doi.org/10.1109/TCAD.2022.3197340)]
- [41] Traeger A, Zadok E, Joukov N, Wright CP. A nine year study of file system and storage benchmarking. *ACM Trans. on Storage*, 2008, 4(2): 5. [doi: [10.1145/1367829.1367831](https://doi.org/10.1145/1367829.1367831)]

附中文参考文献

- [14] 高聪明, 石亮, 刘凯, 薛春, 舒继武. 闪存固态硬盘系统结构与技术. *计算机研究与发展*, 2021, 58(7): 1518–1532. [doi: [10.7544/issn1000-1239.2021.20200690](https://doi.org/10.7544/issn1000-1239.2021.20200690)]

作者简介

王婷煜, 博士生, 主要研究领域为嵌入式操作系统, 文件系统, 存储系统.

杨孟飞, 博士, 研究员, 博士生导师, CCF 会士, 主要研究领域为空间飞行器嵌入式系统, 控制系统, 总体技术.

刘洪标, 博士, 主要研究领域为嵌入式操作系统架构, 实时任务调度.

姜菁菁, 博士, 工程师, CCF 专业会员, 主要研究领域为嵌入式操作系统, 文件系统.

李少峰, 博士, CCF 专业会员, 主要研究领域为嵌入式操作系统, 存储系统, 形式化验证.

刘波, 博士, 研究员, 博士生导师, CCF 高级会员, 主要研究领域为容错计算, 星载计算机体系结构.