

# MyWAL: 一种基于精简输入输出堆栈的键值存储系统性能优化方案

张晓<sup>1,2,3</sup>, 黎梦钰<sup>2,4</sup>, Michael NGULUBE<sup>1,2</sup>, 陈泳豪<sup>1,2</sup>, 赵一平<sup>1</sup>

<sup>1</sup>西北工业大学计算机学院, 中国西安市, 710072

<sup>2</sup>西北工业大学大数据存储与管理工信部重点实验室, 中国西安市, 710072

<sup>3</sup>西北工业大学空天地海一体化大数据应用技术国家工程实验室, 中国西安市, 710072

<sup>4</sup>西北工业大学软件学院, 中国西安市, 710072

**摘要:** 基于日志结构合并 (LSM) 树的键值 (KV) 存储系统可优化随机写入性能, 并提高读取性能, 因此被广泛应用于电子商务、在线分析和实时通信等现代数据存储系统。日志结构合并树将变更的KV数据存在内存中, 批量刷新至内存, 优化了随机写入效率, 但是在系统意外崩溃时会有数据丢失。为了避免内存中的数据丢失, RocksDB在更新内存之前, 会将数据写入写前日志 (WAL) 中。但是开启同步WAL后系统的写入性能会受到较大的影响。在本文中, 我们分析了利用本地文件系统保存WAL的一些缺陷, 在此基础上提出了一种新的WAL机制, 该机制根据WAL文件的特性直接管理原始设备 (或分区), 避免了无用的元数据更新, 同时保证了数据顺序写入磁盘。实验结果表明, 对于固态硬盘 (SSD) SSD上的小KV数据, MyWAL可以将RocksDB的数据写入性能提高5到8倍。在NVMe SSD和非易失性存储器 (NVM) 上, MyWAL可以将数据写入性能提高10% - 30%。此外, YCSB的结果表明, 与SpanDB相比, 写入延迟降低了50%。

**关键词:** 键值 (KV) 存储; 日志结构合并 (LSM) 树; 非易失性存储器 (NVM); NVMe SSD; 写前日志 (WAL)

<https://doi.org/10.1631/FITEE.2200496>