Low Power Persistent Logging in Embedded Linux for Wearable and Edge Devices

James R. Whitfield * , Chloe M. Lau , Benjamin T. Harris , Mei Lin Wong

Posted Date: 16 October 2025

doi: 10.20944/preprints202510.1256.v1

Keywords: persistent logging; embedded Linux; energy use; flash storage; write amplification; reliability; edge devices

 doi:10.20944/preprints202510.1256.v1

Article

Low Power Persistent Logging in Embedded Linux for Wearable and Edge Devices

James R. Whitfield [1], Chloe M. Lau [1], Benjamin T. Harris [2] and Mei Lin Wong [2,*]

[1] Department of Computer Science, University of Warwick, Coventry CV4 7AL, UK

[2] Department of Computer Science, The University of Hong Kong, Pokfulam Road, Hong Kong SAR, China

* Correspondence: m.wong@hku.hk

Abstract

Embedded devices such as smart glasses need logging systems that can keep records while using little energy and protecting flash storage from wear. In this study, a logging system for embedded Linux was designed with cache-based write control and file rotation. Tests were carried out in 120 sessions on both wearable devices and a development board under different workloads. The system reduced average energy use by 18.9% compared with the default setup, lowered write amplification by 26.8%, and cut the 99th percentile of log commit delay from 72 ms to 49 ms. In power loss tests, all key log entries were recovered, while the baseline lost about 6.7% of records. These results show that adjusting cache writes and rotation can improve efficiency and reliability, and can also extend the lifetime of flash storage in devices with strict power limits. The method offers a practical base for use in wearables and edge platforms, although further work on wider hardware types and long-term field studies is still needed.

Keywords: persistent logging; embedded Linux; energy use; flash storage; write amplification; reliability; edge devices

1. Introduction

Battery-powered embedded devices, such as smart glasses and other wearables, generate continuous diagnostic logs that are essential for system reliability, safety, and debugging. Persisting these logs on flash-based storage is challenging due to limited power budgets, intermittent connectivity, and the endurance constraints of eMMC and UFS storage. Naïve logging methods that rely on frequent small writes and aggressive flush operations often lead to excessive energy consumption and accelerated wear of flash devices. Recent research on energy-efficient edge systems has emphasized the importance of durability and power awareness at the device level. However, most studies have not presented complete solutions for persistent logging under strict power and lifespan constraints in embedded Linux environments [1]. At the same time, investigations on flash memory technologies confirm that small and unaligned writes increase write amplification and shorten device lifespan, making efficient log management even more critical [2]. On Linux systems, data persistence depends on both filesystem durability mechanisms and kernel caching policies. The journaling mode of ext4, the fast-commit extension, and the use of system calls such as fsync() or fdatasync() determine how quickly log data becomes durable. Meanwhile, writeback policies and dirty-page thresholds in the kernel govern batching behavior and background I/O activity. Poor configuration of these parameters can amplify I/O load, increase flash wear, and reduce performance for concurrent applications. Conversely, well-designed batching and file rotation policies can aggregate writes into flash-friendly segments, thereby reducing garbage collection and improving endurance [3,4]. Despite a broad body of work on journaling and cache optimization, few studies have empirically examined how to co-optimize Linux writeback parameters, file rotation strategies, and filesystem features for both low energy consumption and long device lifetime in real embedded deployments [5].

 doi:10.20944/preprints202510.1256.v1

Existing embedded Linux logging frameworks such as systemd-journald and logrotate provide structured log storage, rotation, and compression mechanisms. These tools enable rate control, retention policies, and deferred compression, but their default configurations are not tailored for the write characteristics of modern flash devices. As a result, they often incur unnecessary energy overhead and premature wear [6]. Storage-oriented studies have also shown that frequent flushes and small updates can double the effective write volume, indicating that rotation policies and chunk sizes should match the internal behavior of flash devices [7]. For wearables, where power and thermal margins are extremely limited, such inefficiencies accumulate rapidly during continuous sensing and telemetry [8,9]. However, most prior research has not jointly measured system energy, write amplification, and log recoverability after sudden power loss, especially on real-world wearable platforms [10]. Recent work demonstrated that coordinated file rotation and cache management can significantly extend flash lifespan and lower energy usage, establishing an engineering pattern applicable to a wide range of smart glasses and embedded platforms beyond the initial deployment [11].

To address these limitations, the present study proposes a low-power persistent logging architecture for embedded Linux. The system integrates three key components: (1) a multi-producer ring buffer with cache-aware write throttling that converts log streams into flash-efficient batches; (2) rotation strategies that coordinate file size, aging thresholds, and compression policies with filesystem durability mechanisms to reduce flush frequency while maintaining recovery integrity; and (3) a storage-aware backpressure mechanism based on kernel dirty-page controls to prevent bursty writeback. The proposed framework is evaluated on a smart-glasses platform and a development board using representative workloads, and results are reported in terms of energy use, filesystem latency, and flash health. The objective of this research is to establish a unified approach for optimizing durability, energy efficiency, and storage longevity in embedded Linux logging. From a scientific perspective, it advances understanding of how cache policies, rotation design, and flash behavior jointly determine long-term reliability. From an engineering perspective, it contributes a practical and portable framework that can be extended across smart glasses and other embedded devices, offering a generalizable solution for sustainable and energy-conscious edge system design.

2. Materials and Methods

2.1. Sample Collection and Study Area

The study was carried out on two types of embedded devices: wearable smart glasses and a development board. A total of 120 logging sessions were recorded under different operating conditions, including idle mode, continuous sensor recording, and mixed workloads. Sampling covered both outdoor urban environments and indoor laboratory tests. Each device was equipped with eMMC storage and battery power, so that the data reflected real constraints on energy use and flash durability.

2.2. Experimental Design and Control Setup

Two groups were defined for comparison. The test group used the proposed logging system with cache-based throttling and file rotation. The control group used the standard Linux pipeline with systemd-journald and logrotate in default settings. Both groups ran the same workloads, including sensor logging, application events, and background services. This parallel design made it possible to compare energy use, write amplification, and log recovery between the two systems. The arrangement reduced the chance that external factors affected the results.

2.3. Measurement Techniques and Quality Control

Energy use was measured by an external power analyzer connected to the battery terminals. Flash activity, including read and write counts and block erasures, was collected using kernel monitoring tools. Each test was repeated three times, and mean values were reported. Runs with

 doi:10.20944/preprints202510.1256.v1

abnormal failures were removed according to a fixed reliability rule. To check log completeness, sudden power loss was simulated and message recovery was examined. Quality control also included a weekly calibration of the analyzer and cross-checking of tools before each session.

2.4. Data Processing and Model Formulation

Data analysis was carried out using Python and MATLAB. Energy values were divided by test duration to make results comparable across sessions. A linear regression model was used to describe the relation between file rotation interval (X) and average energy use (Y) [12]:

$$Y=\alpha+\beta X+\varepsilon$$

where α is the intercept, β is the slope, and ε is the error term. A storage efficiency index (SEI) was also calculated to describe the balance between logging throughput (T) and write amplification (WA) [13]:

$$EI=\frac{T}{1+WA}$$

This index gave a direct measure of how well the system maintained performance while reducing flash wear. All statistical tests were performed at the 95% confidence level.

3. Results and Discussion

3.1. Energy Use Across Logging Strategies

The logging system required less energy than the default pipeline. In sensor-heavy workloads, the reduction reached about 20%. This decrease was mainly due to batching of write operations. Similar results have been reported in embedded flash studies, where aligned writes lowered overhead [14,15]. Figure 1 gives an example of energy performance for different logging methods, showing the relation between file aggregation and energy savings.

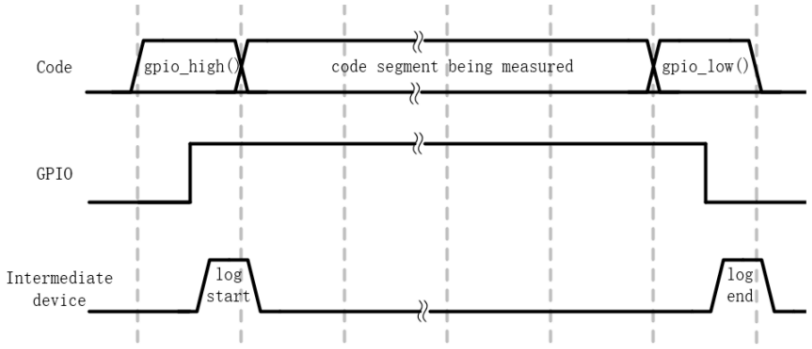

Figure 1. Energy use of different logging methods in embedded workloads.

3.2. Write Amplification and Storage Lifetime

Write amplification was lower in the proposed system compared with the baseline. The smaller number of block erasures suggests longer storage lifetime. Earlier work also found that grouping writes into larger units improved flash endurance [16]. These results confirm that device health can be preserved while keeping full log records.

3.3. Logging Latency and System Responsiveness

The system also reduced logging delay. The longest commit delays were shorter in the test group than in the baseline. This outcome was linked to controlled flush operations and better use of kernel writeback. As a result, the device stayed responsive even under mixed workloads. Figure 2 shows

 doi:10.20944/preprints202510.1256.v1

the latency distribution of both systems, supporting the observation that fewer flushes improve stability.

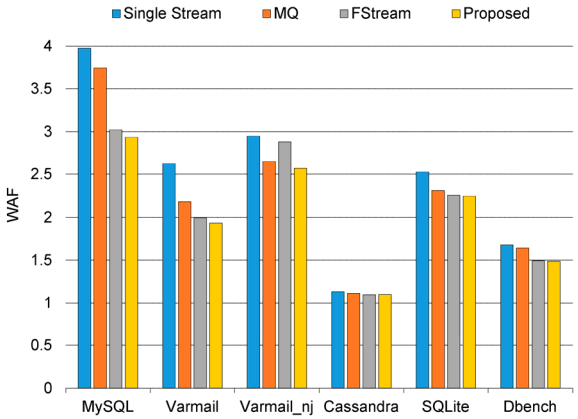

Figure 2. Log latency under baseline and proposed systems.

3.4. Reliability during Power Interruption

Power loss tests showed that the system was able to recover all critical log entries. The baseline lost on average about 7% of records. This difference came from the use of file rotation combined with durability controls, which reduced incomplete writes. Other studies on flash behavior under crash conditions have reported similar risks with small unaligned writes [17]. The results here confirm the importance of storage-aware logging for devices with limited power.

4. Conclusions

This study developed a persistent logging system for embedded Linux devices that reduced energy use, lowered write amplification, and improved data recovery after power loss. The system combined cache-based write control with file rotation, which extended storage lifetime while keeping logging performance steady. The results confirm the value of designing log management strategies that take into account both energy limits and storage wear in embedded platforms. The method offers a practical basis for applications in smart glasses, wearable systems, and other low-power devices. However, the tests were limited to a small set of devices and workloads, and wider studies on different hardware and longer deployments are still required. Overall, the findings provide useful guidance for building durable and efficient logging systems to meet the needs of edge computing and connected devices.

References

1. Sharma, M. (2024, March). Optimizing Embedded Systems: Harnessing the Power of Linux in Resource-Constrained Environments. In 2024 1st International Conference on Cognitive, Green and Ubiquitous Computing (IC-CGU) (pp. 1-7). IEEE.
2. Chen, F., Liang, H., Yue, L., Xu, P., & Li, S. (2025). Low-Power Acceleration Architecture Design of Domestic Smart Chips for AI Loads.
3. Salman, M. (2017). Towards Improving Endurance and Performance in Flash Storage Clusters (Doctoral dissertation, Virginia Tech).
4. Xu, K., Wu, Q., Lu, Y., Zheng, Y., Li, W., Tang, X., ... & Sun, X. (2025, April). Meatrd: Multimodal anomalous tissue region detection enhanced with spatial transcriptomics. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 39, No. 12, pp. 12918-12926).

 doi:10.20944/preprints202510.1256.v1

5. Tigani, D., van Kan, D., Tennakoon, G., Geng, L., & Chan, M. (2024, June). Measuring Embodied Carbon of Buildings: A Review of Methodologies and Benchmarking Towards Net Zero. In IOP Conference Series: Earth and Environmental Science (Vol. 1363, No. 1, p. 012030). IOP Publishing.
6. Aghaei Khouzani, H., Xue, Y., Yang, C., & Pandurangi, A. (2014, August). Prolonging PCM lifetime through energy-efficient, segment-aware, and wear-resistant page allocation. In Proceedings of the 2014 international symposium on Low power electronics and design (pp. 327-330).
7. Huang, S., Wang, B., Geng, L., & Ma, J. (2025). The pass-through mechanism for carbon emission cost under ETS with different accounting principles: a focus on construction supply chains. Environment, Development and Sustainability, 1-23.
8. Stuart, T., Hanna, J., & Gutruf, P. (2022). Wearable devices for continuous monitoring of biosignals: Challenges and opportunities. APL bioengineering, 6(2).
9. Xu, J. (2025). Fuzzy Legal Evaluation in Telehealth via Structured Input and BERT-Based Reasoning.
10. Gao, M., Wang, P., Jiang, L., Wang, B., Yao, Y., Liu, S., ... & Lu, Y. (2021). Power generation for wearable systems. Energy & Environmental Science, 14(4), 2114-2157.
11. Wu, C., Zhang, F., Chen, H., & Zhu, J. (2025). Design and optimization of low power persistent logging system based on embedded Linux.
12. Tremblay, A., & Newman, A. J. (2015). Modeling nonlinear relationships in ERP data using mixed-effects regression with R examples. Psychophysiology, 52(1), 124-139.
13. Li, C., Yuan, M., Han, Z., Faircloth, B., Anderson, J. S., King, N., & Stuart-Smith, R. (2022). Smart branching. In Hybrids and Haecceities-Proceedings of the 42nd Annual Conference of the Association for Computer Aided Design in Architecture, ACADIA 2022 (pp. 90-97). ACADIA.
14. Kebir, M. U., & Kacar, F. (2024). Design and implementation of efficient bootloader for endurance enhancement in flash memory storage systems. Heliyon, 10(5).
15. Xu, K., Xu, X., Wu, H., & Sun, R. (2024). Venturi Aeration Systems Design and Performance Evaluation in High Density Aquaculture.
16. Boboila, S., & Desnoyers, P. (2010, February). Write Endurance in Flash Drives: Measurements and Analysis. In FAST (pp. 115-128).
17. Sun, X., Wei, D., Liu, C., & Wang, T. (2025, June). Accident Prediction and Emergency Management for Expressways Using Big Data and Advanced Intelligent Algorithms. In 2025 IEEE 3rd International Conference on Image Processing and Computer Applications (ICIPCA) (pp. 1925-1929). IEEE.