

# JLFS: Journaling the Log-Structured Filesystem for Proactive Cleaning in Flash Storage

Andromachi Hatzieleftheriou and Stergios V. Anastasiadis

{ahatziel, stergios}@cs.uoi.gr

Dept. of Computer Science, University of Ioannina, Greece

## 1. Flash Memory

Attractive features

Idiosyncrasies

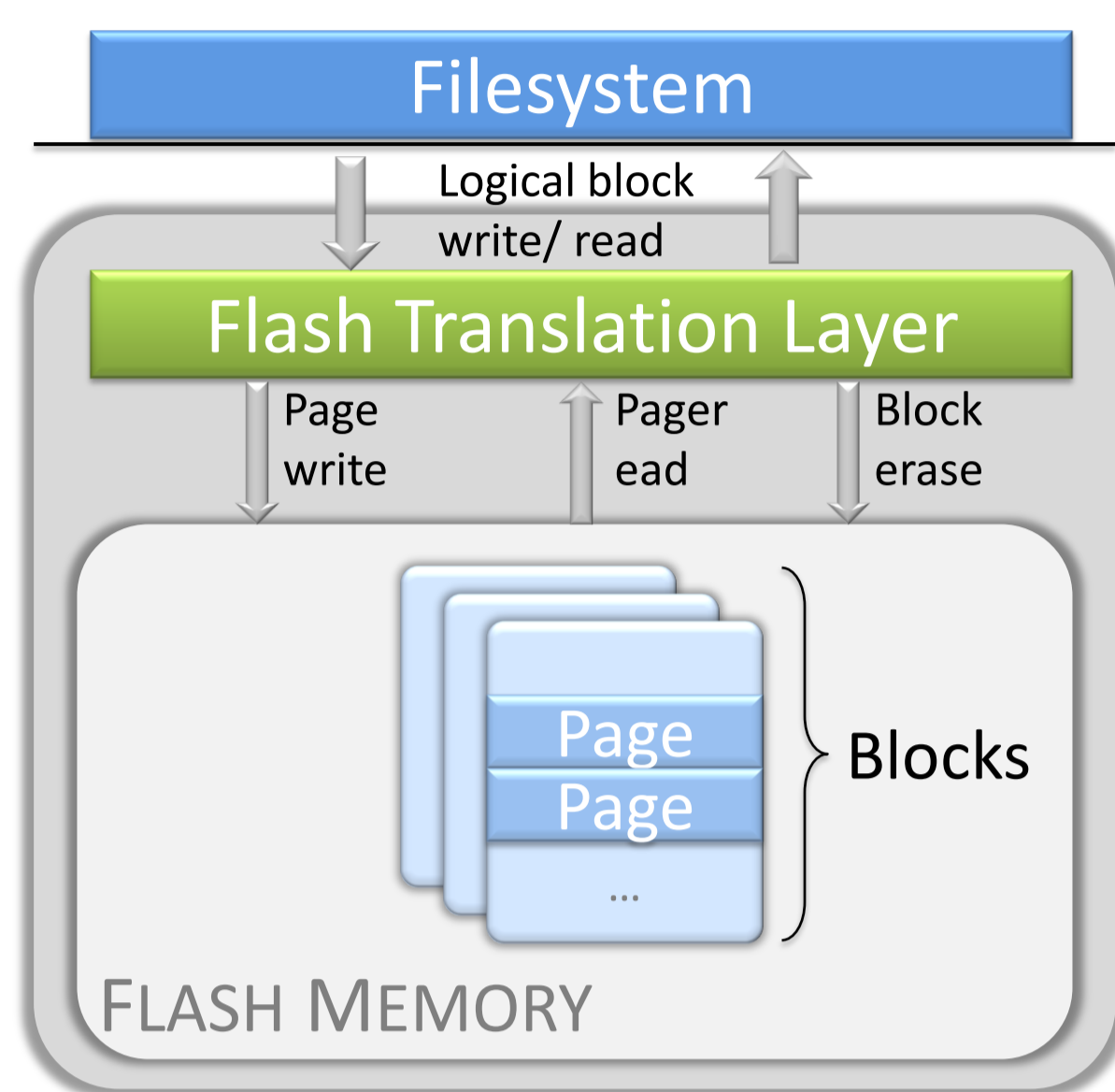
- workload-dependent performance

Low-level modifications

- flash translation layer

Flash-aware fs

- semantic information
- no functionality duplication
- retroactive decisions



## 2. Garbage Collection

Log-structured approach typically used

- avoid random writes
- need for recycling of invalid pages

FTL approach:

- separate hot/cold pages based on update frequency
- possible misclassification of access frequency
- bookkeeping overhead

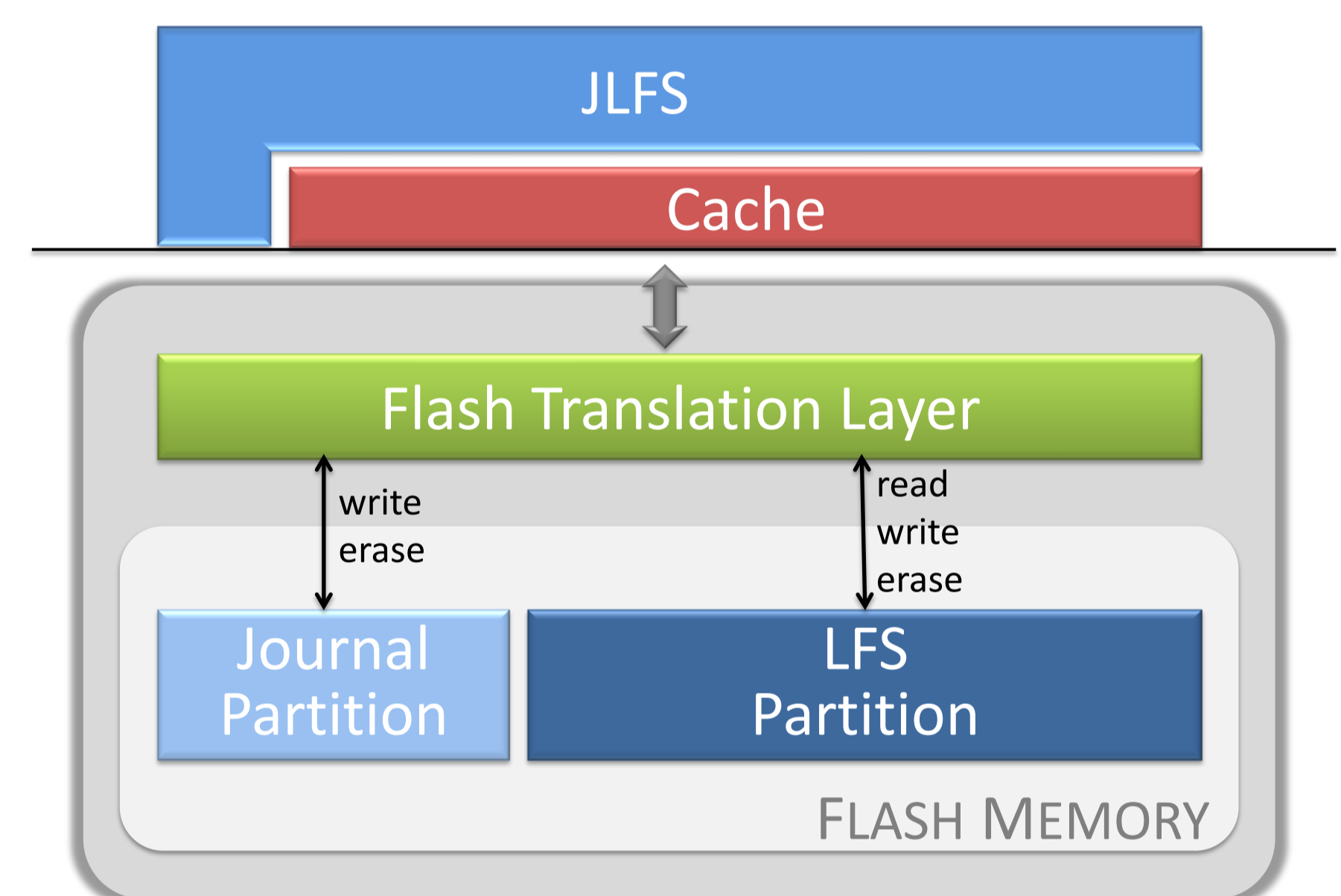
## 4. Proposed Architecture

JLFS - Journaled Log-structured File System

- combine journaling functionality with LFS

Flash storage management

- two distinct partitions; LFS and journal



## 3. Design Goals

- Avoid function duplication between fs and firmware
- Minimize write traffic without compromising persistence
- Minimize data relocation traffic
- Identify access characteristics using existing cache mechanisms

Fs level approach:

- use semantic knowledge and existing system operations to manage flash idiosyncrasies

## 5. JLFS Concepts

Journal responsibility

- proactively clean permanent state from frequently updated data and metadata

Idea

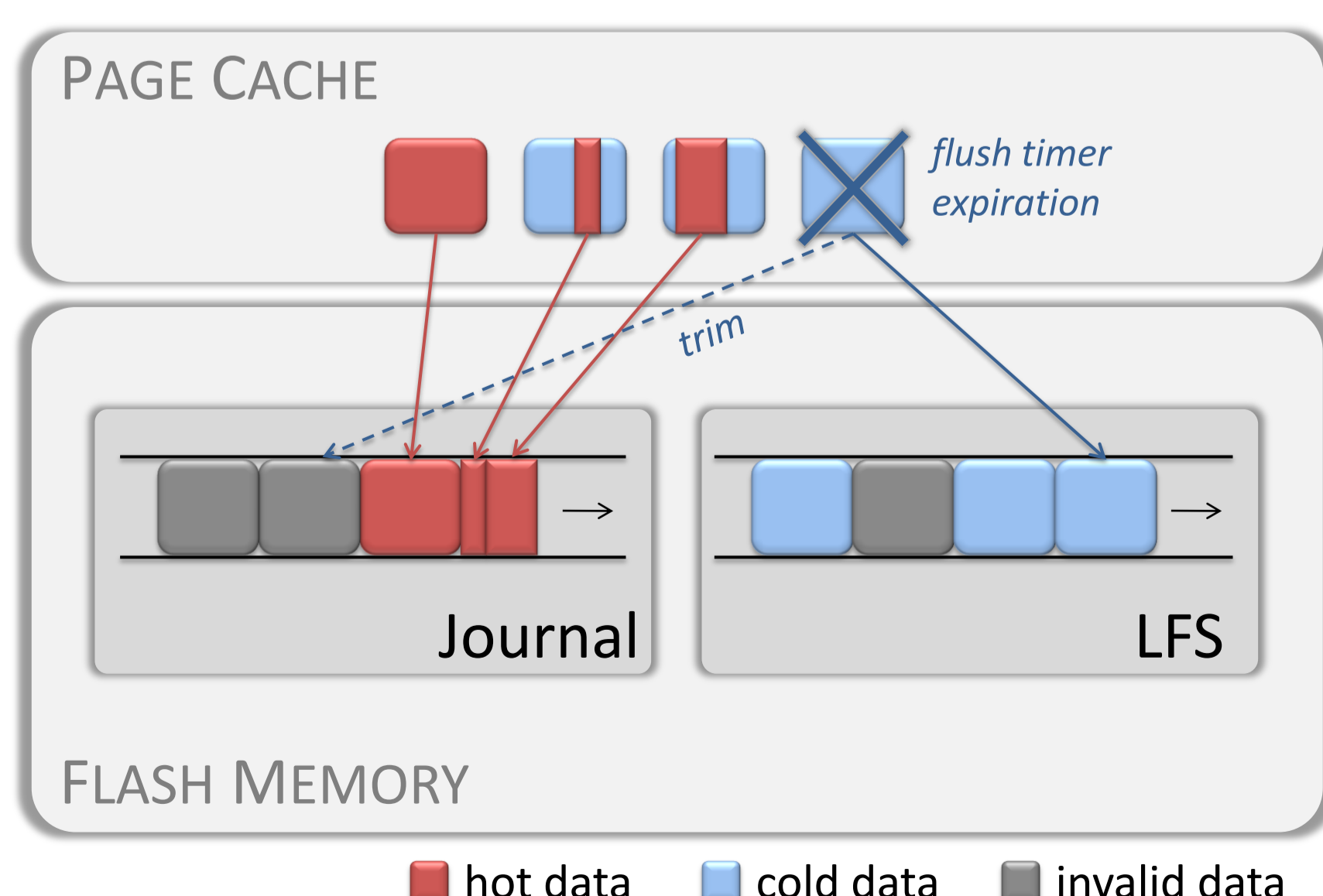
- categorize pages into hot/cold based on cache timers
- respectively transfer data to journal or to LFS partition

LFS partition

- contains mainly valid data → cold pages

Journal Partition

- valid blocks at the front and clean at the rest → hot pages



## 6. JLFS Features

- Relatively cheap writes
- Batching of multiple requests before reaching LFS
- Flash reads avoided during garbage collection
- Possible to adjust hotness boundary to workload
- Selectively journal data based on a write size threshold
  - avoid traffic duplication in case of sequential requests
- Journal data updates at subpage granularity
  - Reduce journal traffic in case of small writes
- Reduced recovery time

## 7. References

- Chen, F., Koufaty, D. A., and Zhang, X. *Understanding intrinsic characteristics and system implications of flash memory based solid state drives*. In ACM SIGMETRICS/IFIP Performance (2009).
- Hatzieleftheriou, A., and Anastasiadis, S. V. *Okeanos: Wasteless journaling for fast and reliable multistream storage*. In USENIX Annual Technical Conference (2011).
- Kawaguchi, A., Nishioka, S., and Motoda, H. *A flash memory based file system*. In USENIX Winter Technical Conference (1995).
- Lee, S., Shin, D., Kim, Y.-J., and Kim, J. *Last: locality-aware sector translation for nand flash memory-based storage systems*. Operating Systems Review 42, 6 (2008).