Revisiting Log-Structured File Systems for Low-Power Portable Storage

Andreas Weissel • Holger Scherl • Philipp Janda

Department of Computer Sciences 4 (Operating Systems)
University of Erlangen-Nuremberg
{weissel,scherl,janda}@cs.fau.de

Frank Bellosa

System Architecture Group
University of Karlsruhe
bellosa@ira.uka.de

IWSSPS'05

Motivation

- Mobile, embedded, portable devices
  - ◆ battery-powered
- Ever-increasing demand for storage space
  - ◆ hard disks as alternative to other storage media (flash memory)
  - ➜ significant power requirements (spindle motor)
  - ➜ operating modes with reduced power consumption (low-power idle, standby)

Motivation

- Mobile, embedded, portable devices
  - ◆ battery-powered

- Ever-increasing demand for storage space
  - ◆ hard disks as alternative to other storage media (flash memory)
  - ➜ significant power requirements (spindle motor)
  - ➜ operating modes with reduced power consumption (low-power idle, standby)

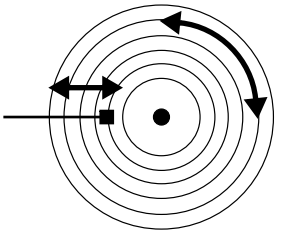
  - ➜ impact of file system and data layout on energy consumption?
  - ➜ revisiting log-structured file systems

Outline

- Guidelines for Energy Efficient File System Design
- Implementation
- Evaluation
- Energy-aware Free Space Management

Energy Consumption of Hard Disks

- Energy is wasted due to
  - disk seeks
  - rotational latencies

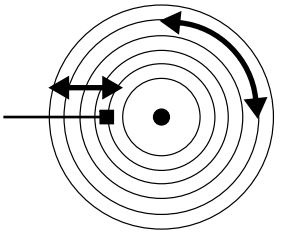


- IBM Travelstar

Mode	Power	Average Time	Energy
Reading	2.5 W	0.24ms (4 KB)	0.61 mJ
Writing	2.1 W	0.25ms (4 KB)	0.53 mJ
Seeking	2.3 W	12ms	27.6 mJ
Rotation	2.0 W	7.1ms	14.2 mJ
Idle	0.65 W	—	—
Standby	0.25 W	—	—

Energy Consumption of Hard Disks

- Energy is wasted due to
  - disk seeks
  - rotational latencies

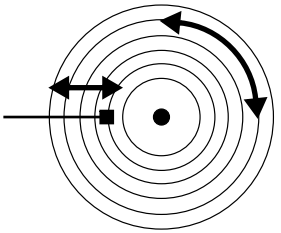


- IBM Travelstar

Mode	Power	Average Time	Energy
Reading	2.5 W	0.24ms (4 KB)	0.61 mJ
Writing	2.1 W	0.25ms (4 KB)	0.53 mJ
Seeking	2.3 W	12ms	27.6 mJ
Rotation	2.0 W	7.1ms	14.2 mJ
Idle	0.65 W	—	—
Standby	0.25 W	—	—

Energy Consumption of Hard Disks

- Energy is wasted due to
  - disk seeks
  - rotational latencies

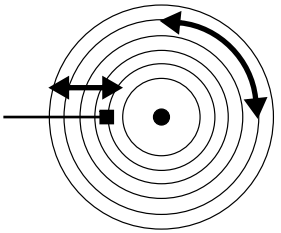


- IBM Travelstar

Mode	Power	Average Time	Energy
Reading	2.5 W	0.24ms (4 KB)	0.61 mJ
Writing	2.1 W	0.25ms (4 KB)	0.53 mJ
Seeking	2.3 W	12ms	27.6 mJ
Rotation	2.0 W	7.1ms	14.2 mJ
Idle	0.65 W	—	—
Standby	0.25 W	—	—

Energy Consumption of Hard Disks

- Energy is wasted due to
  - disk seeks
  - rotational latencies



- IBM Travelstar

Mode	Power	Average Time	Energy
Reading	2.5 W	0.24ms (4 KB)	0.61 mJ
Writing	2.1 W	0.25ms (4 KB)	0.53 mJ
Seeking	2.3 W	12ms	27.6 mJ
Rotation	2.0 W	7.1ms	14.2 mJ
Idle	0.65 W	—	—
Standby	0.25 W	—	—

Guidelines for Energy Efficient File System Design

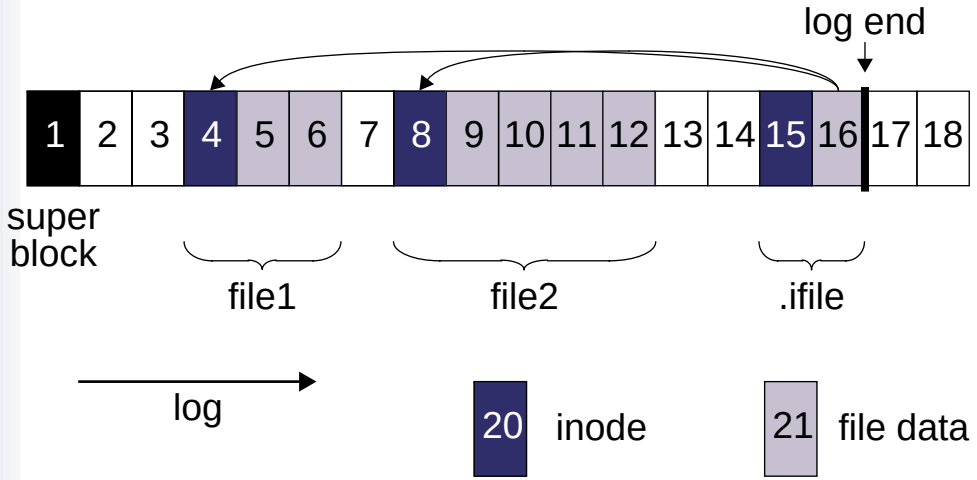
- Reduce disk seeks & rotational latencies
  - ➜sequential reads/writes
  - ➜increase length of idle periods; more opportunities to switch to standby
- Sequential writes
  - ◆abstraction of an infinite log, new data is always appended
  - ◆group meta data and data blocks together
  - ➜free space management
  - ➜different approaches with different energy requirements
- Sequential arrangement of file data
  - ◆reduce fragmentation
- Fast crash recovery
  - ◆checkpoint (regions)

Implementation

- *scherlfs* (student thesis)
  - log-structured file system similar to Rosenblum's LFS
- Inodes are stored dynamically in the log
  - translation of inode number to corresponding disk blocks
  - inode map .ifile (read-only file in root directory)

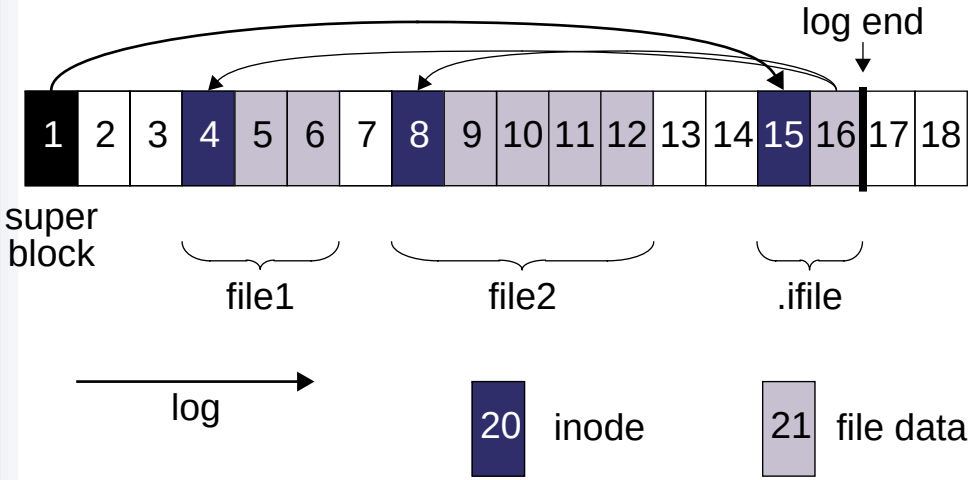
Implementation

- Inode map .ifile:
  translation of inode number to corresponding disk blocks



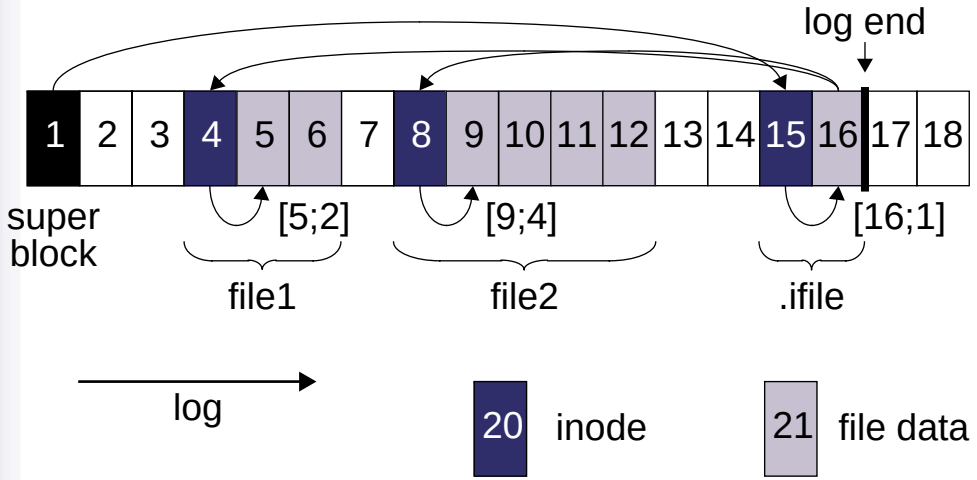
Implementation

- Checkpoints are defined by storing the position of the .ifiles' inode in the superblock



Implementation

- Mapping of file block numbers to logical block numbers on hard disk



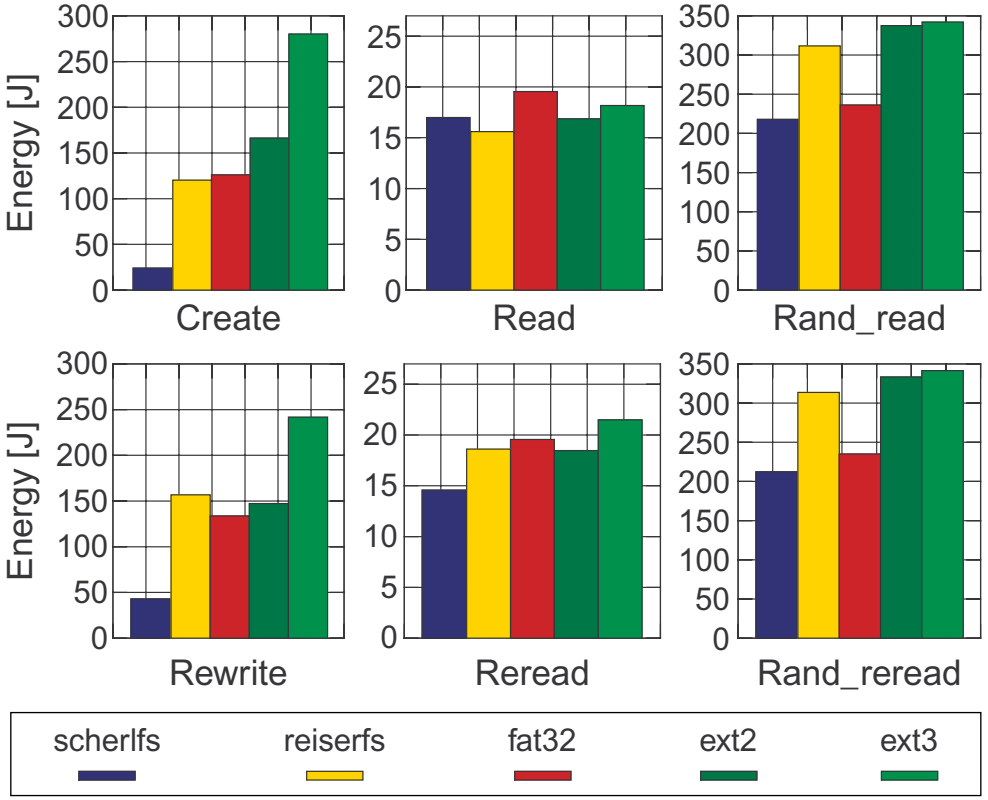
- indirect blocks can cause additional disk seeks
- ➜ *extents* (block number, length)

Evaluation

- Measure time & energy consumption of automated test runs
  - ◆ no free space management yet—avoid filling the file system completely
- Concentrate on influence of data layout on energy consumption
  - ◆ replay trace files of I/O operations without idle periods
  - ➜ even higher energy savings possible as idle periods are increased: more opportunities to switch hard disk to standby mode
  - ◆ synchronous writes
  - ◆ deactivate caching of file data in the Linux kernel; still caching meta data; internal caches not affected (e.g., FAT cache)
- Comparison with reiserfs (version 3), fat32, ext2 and ext3

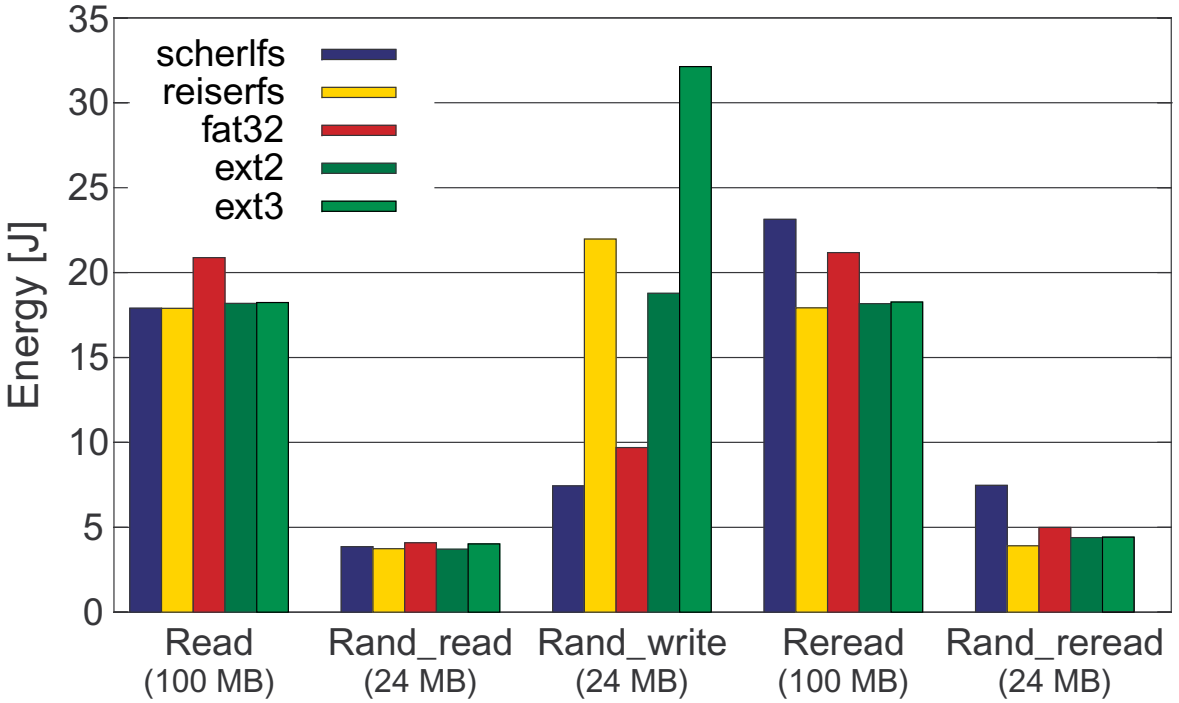
Evaluation

- Energy Consumption of Sequential File Operations (4 KB files)



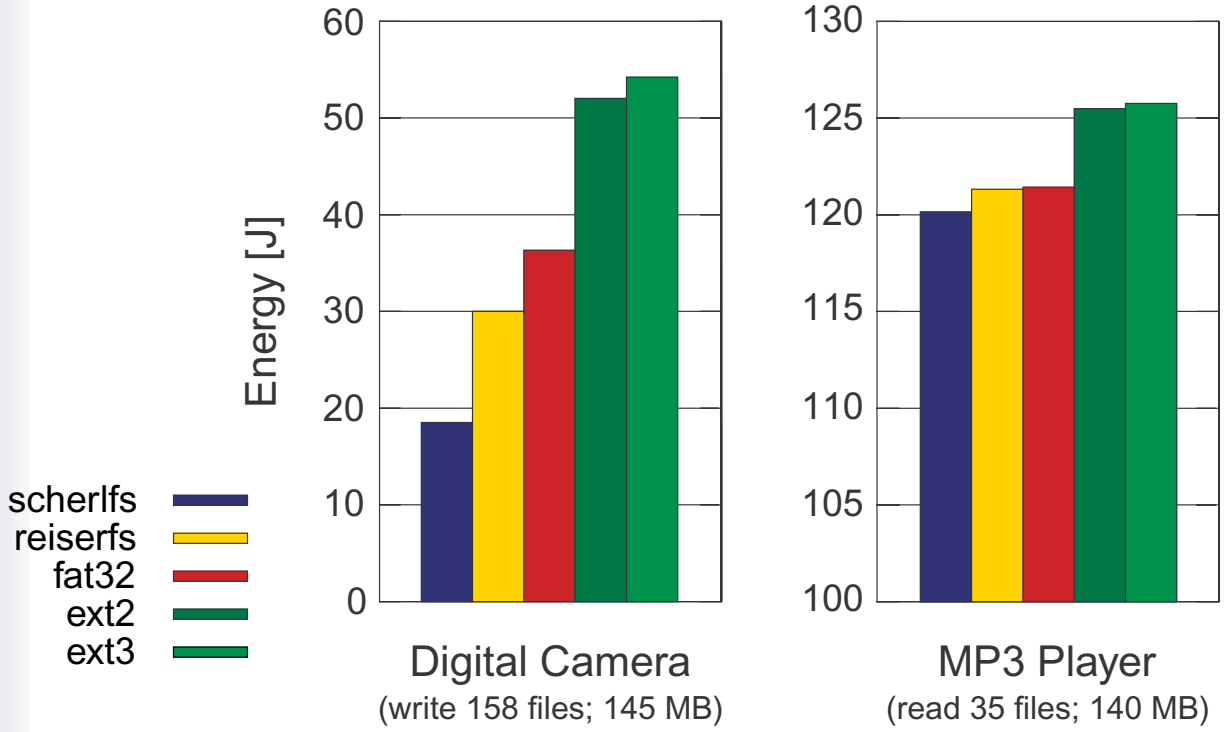
Evaluation

- Energy Consumption of Random Updates



Evaluation

- Digital Camera (writes), MP3 player (reads)



Outline

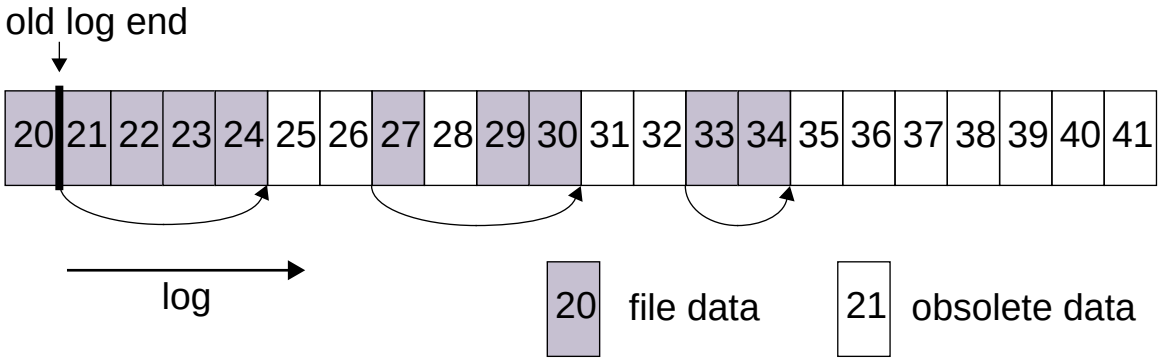
- Guidelines for Energy Efficient File System Design
- Implementation
- Evaluation
- Energy-aware Free Space Management

Approaches to Free Space Management

- At some time, the log will reach its tail
- Deleting & updating of data results in *holes* in the log
  - ➜reuse holes to create free space for the log
  - ➜reduce fragmentation
- Approaches
  - Threading
  - Copy and Compact
  - Hybrid approach

Approaches to Free Space Management

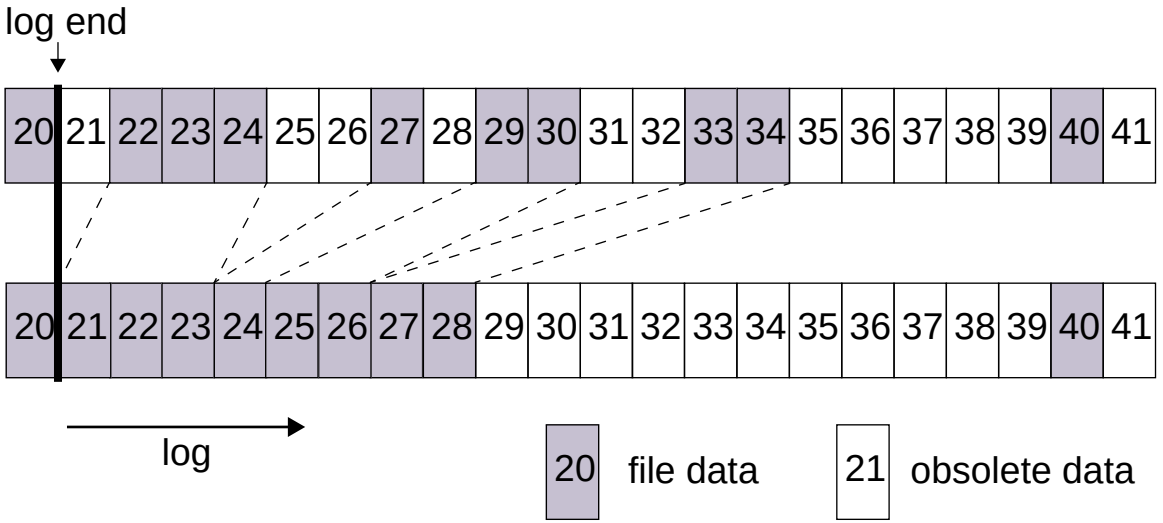
- Threading
  - ◆ log is threaded through the holes
  - ➜ writes to the log: overhead due to seeks & rotational latencies
  - ➜ fragmentation of file data



  - ◆ position and size of holes has to be known
  - ➜ special file to store information about holes in the log
  - ➜ holes as file data: extents point to free space

Approaches to Free Space Management

- Copy and Compact
  - ◆ eliminate holes by compacting (parts of) the log
  - ➜ overhead of copy operations
  - ➜ combine with defragmentation



Approaches to Free Space Management

- Hybrid approach (as in LFS)
  - ◆ divide storage space into segments (256 K–1 MB)
  - ➜ *copy and compact* on whole segments
    create empty segments by merging partially full segments
  - ➜ *thread* log through empty segments

Energy-aware Free Space Management

- When and how should free space management be invoked?
  - How much energy is available for cleaning?
    re-charging vs. battery-powered
  - How much free space should be generated?
  - How much memory is available for temporary data during cleaning?

- Copy and Compact combined with defragmentation
  (optimize data layout)
  - ➜during charging

Free Space Management: Battery Power

- Threading

  (+) no cleaning or copy operations

  (–) fragmentation

- Copy and Compact

  (–) additional overhead due to copying

  (+) group with other I/O operations, reduce effects on idle periods

  - limit number of copy operations:
    split large areas of file data to fill smaller holes in the log
    (–) fragmentation

Conclusion

- Data layout on hard disk & file system design have an impact on the energy consumption of the storage subsystem
- Sequential reads and writes to avoid disk seeks, rotational latencies

  ➜log-structured file system design
- Energy-aware free space management