

Logstructured Filesystems

Matthias Janke

Seminar Dateisysteme SS 2007

05.06.2007

Überblick

- 1 Motivation
- 2 Grundlagen
 - Design Prinzipien
 - Dateien schreiben
 - Dateien lesen
 - Platz Management
 - Recovery
 - Potentiale und Fallstricke
- 3 Implementationen
 - Überblick
 - JFFS2
 - YAFFS2
 - NilFS
 - LogFS
- 4 Quellen

Motivation

- historisch
 - I/O Bottleneck!!
 - schnelles Recovery
 - aktuell
 - Flashzellen
 - CD/DVD RW
 - DVD-RAM
 - HD DVD-RW, BD-RW?????
- ⇒ Medien mit beschränkter Anzahl Schreibzugriffe (1000-100000)

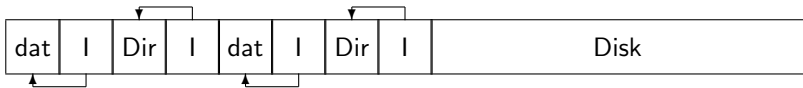
Design Prinzipien

- Minimierung von seeks
- Asynchrones I/O
- Optimierung der Schreibzugriffe im RAM
- Reads werden durch Caches im RAM optimiert

Dateien schreiben

- Diskzugriffe erfolgen sequentiell
- schreiben in Massen
- asynchron
- inodes und Daten werden sequentiell geschrieben.

Auf die Platte



Dateien lesen

- Optimierung durch größere Caches
- inodes sind nicht an fixen Positionen auf Platte
- inode Verwaltung durch map im RAM
- ansonsten identisch zu FFS

Platz Management

- Platz wird mittels Segmenten von fixer Größe verwaltet
- Segmente groß genug um seektime zu amortisieren
- Segmente enthalten summary Blocks
- nicht vollständig benutzte Segmente werden aussortiert
- Segmente werden nur bei explizitem sync, vollen Caches oder write-back

Recovery

- nur das letzte Segment ist betroffen
- restore braucht sich nur um das letzte Segment zu kümmern
- Checkpoint werden verwendet um konsistente Bereiche zu markieren

Small File Benchmarks

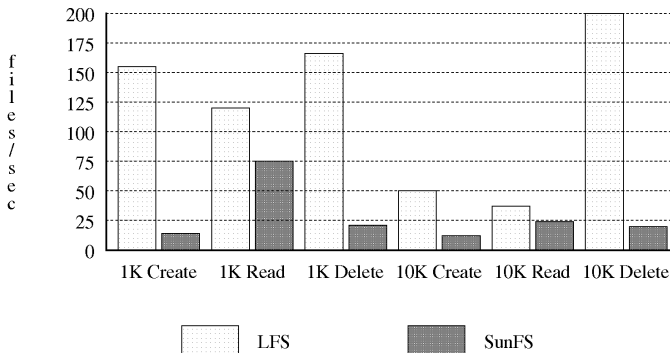


Figure 3 — Small file I/O test.

Measurements of creating, reading, and deleting many 1K and 10K files using LFS and the SunOS file system. The creation phase of the test measured the speed at which 10000 one-kilobyte and 1000 ten-kilobyte files could be created. Following the creation, the file cache was flushed and all the files were read (in the same order as they were created). Finally, we measured the speed at which the files could be deleted. All

Large File Benchmarks

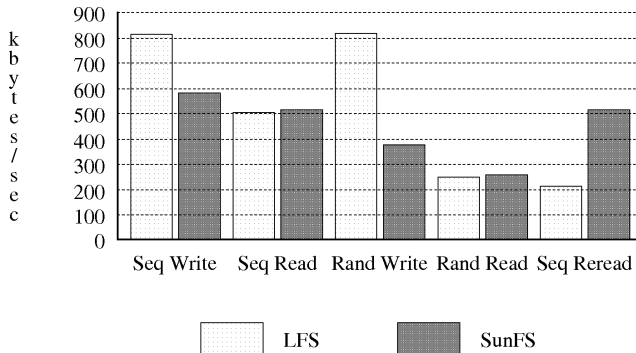


Figure 4 — Large file I/O test.

Transfer rates for reading and writing an 100 megabytes file. The figure shows the rate in kilobytes per second to create and write 100 megabytes sequentially, read 100 megabytes sequentially, write 100 megabytes randomly to the file, and read 100 megabytes randomly from the file. The final test was to read 100 megabytes sequentially after randomly writing the file.

Garbage Collection Benchmarks

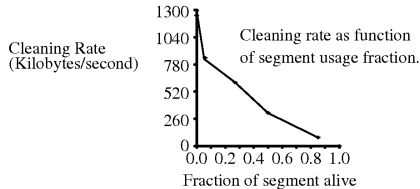


Figure 5 — LFS segment cleaning rate.

Measured rate of segment cleaning as a function of segment utilization. The rate is presented in kilobytes per second that clean segments can be generated.

Potentiale und Fallstricke

- Vorteile
 - Hoher Schreib Durchsatz
 - Hohe Datensicherheit
 - Schneller Restore
- Nachteile
 - Fragmentation
 - Random Access ist Katastrophal!

Logbased Dateisysteme im Überblick (Auswahl)

- LFS – Sprite OS
- BSD-LFS – *-BSD, aktuell NetBSD
- FossilFS – Plan9
- LinLogFS – Linux 2.2
- UDF – optische Medien
- NilFS – Linux 2.6
- LFS (SoC) – Linux 2.6, coop mit NilFS
- LogFS (OLPC) – Linux 2.6, nicht Logbased
- JFFS2 – Linux 2.6
- YAFFS2 – Linux 2.6

nach [2]

Überblick

- für NOR-flash
- speziell an Eigenschaften von Flash angepasst
- Kompression
- Verwaltungsdaten nicht komplett im RAM
- 4 Typen von Segmenten
- Hohe Formale Korrektheit
- der Flash hat immer recht!
- Linux 2.4 & 2.6, eCos

Überblick

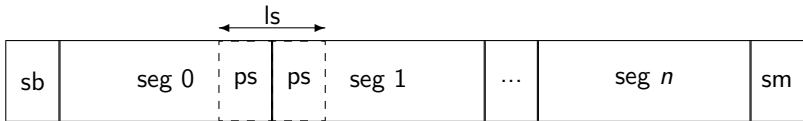
- speziell für NAND-flash
- sehr Fehlertolerant (CRC32 Überall)
- FS-logik im Userspace
- hoch Portabel
- Linux 2.6, WinCE, eCos, ...

NiFS Eigenschaften

- primär für high Performance I/O
- hohe Zuverlässigkeit
- hohe Verfügbarkeit
- hohe Skalierbarkeit
- als verteiltes bzw. Cluster Dateisystem nutzbar
- Benutzerfreundlich
- Linux 2.6 Modul

⇒ <http://www.osrg.net/nilfs>

Disklayout



sb = super block

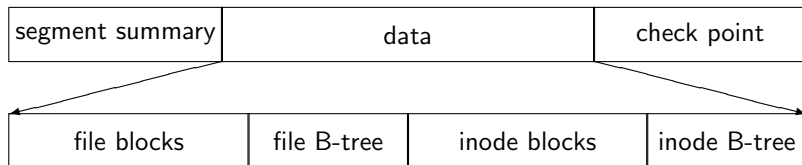
seg = segment

ps = partial segment

ls = logic segment

sm = segment management

Segmentlayout



Überblick

- basiert auf wandernden Bäumen → kein LFS!
- Wurzelknoten in konstanter Zeit auffindbar
- out-of-place Updates
- OLPC
- Linux 2.6

Quellen

 Rosenblum, M., and Ousterhout, J. "The LFS Storage Manager." Proceedings of the 1990 Summer Usenix, Anaheim, CA, June 1990, pp. 315-324.



http://en.wikipedia.org/wiki/Log-structured_file_system, 11:52, 17 May 2007



J. Engel R. Mertens, LogFS – finally a scalable flash file system



Nilfs team, the Nilfs version 1: overview, NTT Corporation



D. Woodhouse, JFFS: The Journaling Flash File System, Red Hat Inc.



C. Manning, YAFFS: the NAND-specific flash file system – Introductory Article, linuxdevices.org, 20 September 2002



Admin, YAFFS Development Notes, aleph1.co.uk