Power Management Praktikum

Implementierung eines Timeout-Mechanismus

Sebastian Döring
und Michael Schöffler

Aufgabenstellung

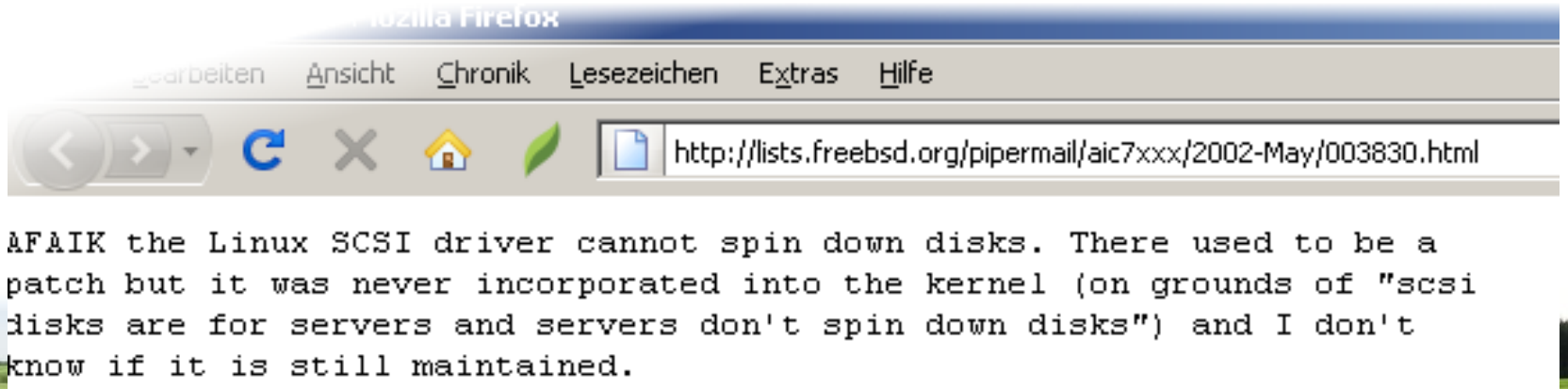
- DDT-Algorithmus *(Device Dependant Timeout):* Nach einer Inaktivitätsphase (break even time) soll die Festplatte heruntergefahren werden

- Erweiterung zum DDT/ES-Algorithmus *(Device Dependant Timeout / Early Shutdown):* Unmittelbar nach einer Phase von Zugriffen soll die Festplatte heruntergefahren werden

Randbedingung:
Der pdflush-Dämon (schreibt alle 5 Sekunden modifizierte Seiten, die älter als 30 Sekunden sind vom Cache auf die Platte) soll die Festplatte nachträglich nicht wieder hochfahren.

Herausforderungen

- Zielgerät ist eine IDE-Festplatte, diese konnte jedoch nur über die SCSI-Schnittstelle angesprochen werden
- Aktivierung des IDE-Treibers bei gleichzeitig aktivem SCSI-Support im Kernel nicht mehr möglich (bzw. uns nicht gelungen)
- Mehrere SCSI-Befehle funktionieren nicht mit IDE-Platten



Entwurf

DDT oder DDT/ES
(Modul-Parameter)

Break-even-Zeit
(Proc-Datei)

Prüft einmal pro Sekunde ob die Bedingungen für ein Abschalten erfüllt sind.

Erkennt Ende einer Phase (DDT/ES).

Schreibt Metainformationen in das SCSI-Device (z.B. Ende einer Phase).

Thread:
check_for_standby_task()

Aktivierung über Mutex.

Überführt die Festplatte in den Ruhezustand.

Thread:
scsi_power_task()

Modul: hd_power_management

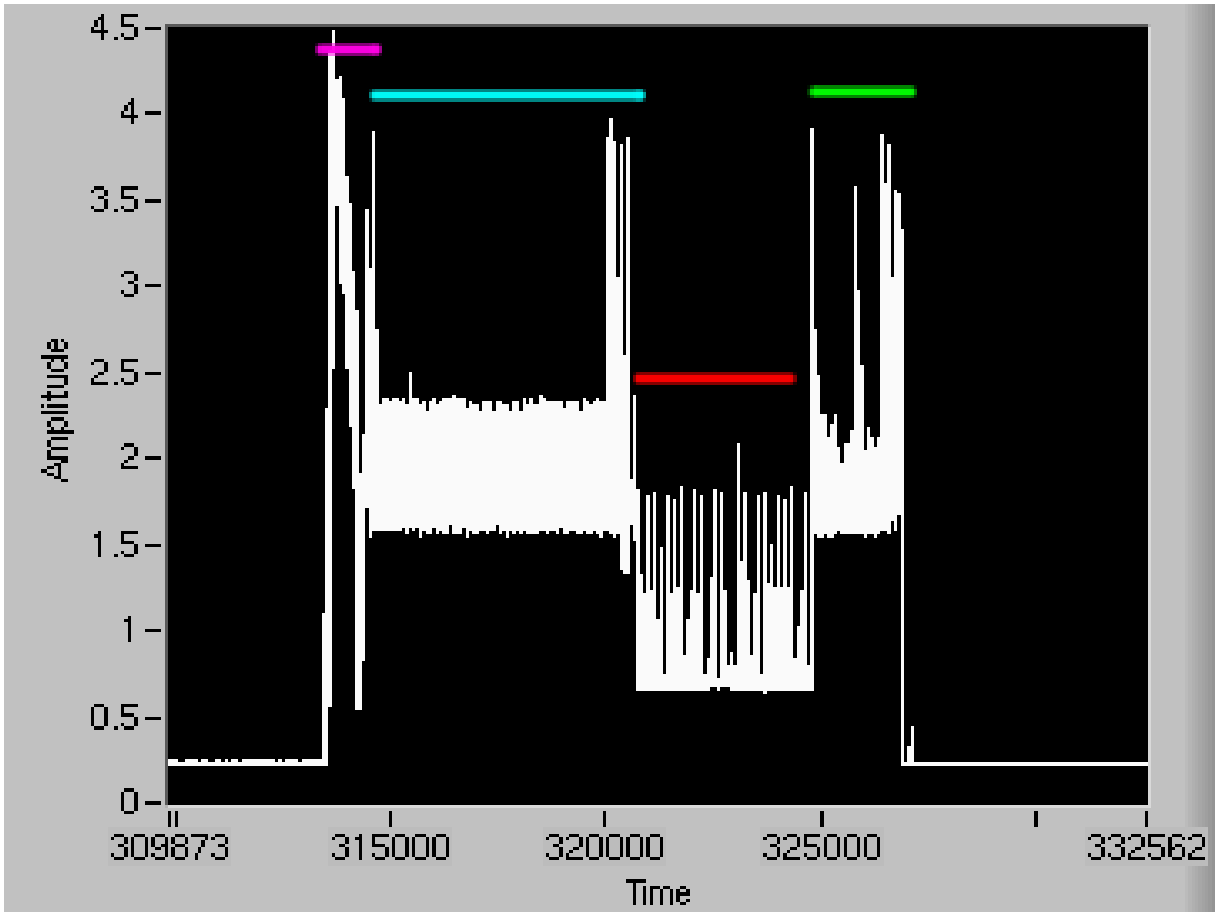
Kernel-Modifikationen

- Struktur scsi_device wurde um Zugriffsinformationen und den Festplattenzustand erweitert
  - int power_mode, new_power_mode;
    unsigned long break_even_period;
    struct timespec first_access, last_access;
    unsigned long busy_period;
    bool phase_end_reached;
- In der Funktion sd_prep_fn(...) werden Strukturerweiterungen behandelt
- IDE-Festplatte wird über eine globale Variable bekannt gemacht

Cache/HD-Synchronisation

- Synchronisierung des Caches mit der Festplatte funktioniert mit dem Systemaufruf sys_sync() sehr gut (~weniger als eine Sekunde)
- SCSI-Funktionen zeigten keine Wirkung
- Die Funktion wakeup_pdflush(...) aus „mm/page-writeback.c“ benötigte mehrere Sekunden für eine Synchronisierung

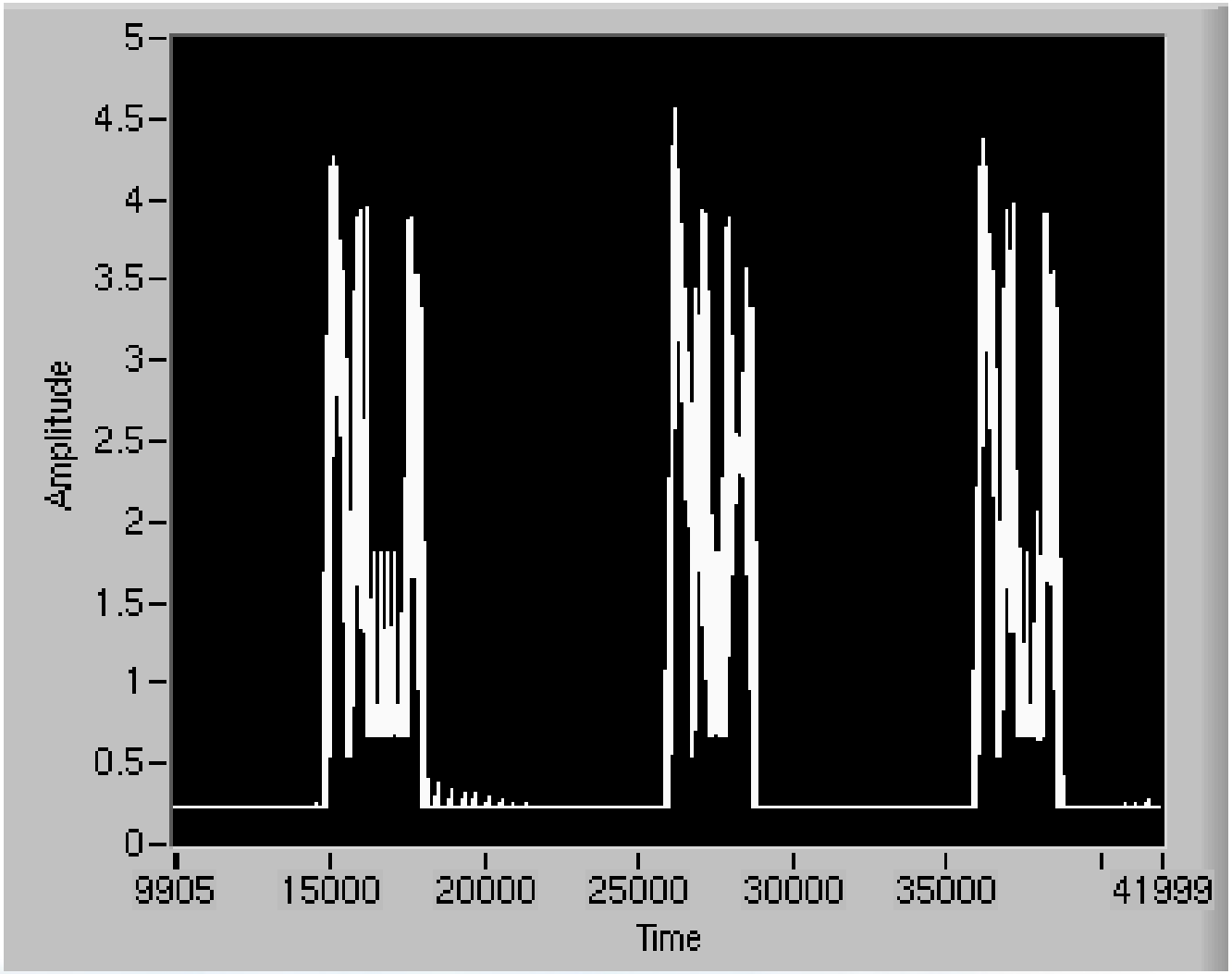
Ergebnisse DDT



- HD-Spinup
- Read/Write Access
- break_even_time
- syn_sync(), HD-Spindown

Break EvenTime: ~4.2 Sekunden

Ergebnisse DDT/ES



Nachteile des Verfahrens

- Hohe Verzögerung von Zugriffszeiten durch notwendiges Hochfahren der Festplatte (ca. 1.7s)
- DDT/ES: Gute Prognosen kaum möglich. Heuristik zu einfach, zu viele parallele Zugriffe in normaler Arbeitsumgebung (mp3, Office, Firefox, usw.)
- Mechanischer Verschleiß durch häufiges Anfahren der Platte. Max. 50 000 Spin-Ups ist geläufig bei heutigen Festplatten

Siehe z. B. Samsung-Spezifikationen

Exemplarisches Beispiel

- Festplatte anhalten
  - Dauer: 869 ms Energie: 2,27 W (1 976,62 mJ / 869 ms
- Festplatte anfahren
  - Dauer: 1 706 ms Energie: 1,42 W (2 428,46 mJ / 1706 ms)
- Verbrauch
  - Im Ruhezustand: ~0,23 W
  - Im Idle: ~0,64 W

=> Festplatte abschalten lohnt sich nach 4,24 Sekunden

Ungeklärte Phänomene

- Synchronisierung benötigt relativ viel Zeit
  - Tritt auffällig häufig bei kleineren Zugriffen auf