

Hard Disk Power management

Implementierung eines Timeout-Mechanismus

Gruppe 4

Überblick

- Die Algorithmen
- Implementierung
- Struktureller Aufbau
- Ansatzpunkte im IDE Treiber
- Scheduling und Synchronisierung
- Aufgetretene Probleme
- Auswertung

Algorithmen

DDT-Algorithmus

Festplatte in den Standby-Modus versetzen, falls

$$t_{la} + t_{be} \leq t$$

Die Variablen bedeuten:

t : die aktuelle Zeit

t_{la} : wann wurde zuletzt auf die Festplatte zugegriffen

t_{be} : die break-even-Zeit

DDT/ES-Algorithmus

Festplatte in den Standby-Modus setzen, falls

$$t_{ia} + t_{ba} \leq t$$

oder

$$t_{fa} + t_{fb} \leq t \leq t_{fa} + t_{fb} + t_1 \quad \text{und} \quad t_{fa} + t_2 \leq t$$

Die Variablen bedeuten:

t : die aktuelle Zeit

t_{ib} : die Länge der letzten aktiven Phase (Dauer eines I/O-Transfers, busy_period)

t_{fa} : der Zeitpunkt des ersten Zugriffs in der gerade aktiven Phase (first_access)

t_{la} : wann wurde zuletzt auf die Festplatte zugegriffen

t_{be} : die break-even-Zeit

t_1 : Toleranzwert beim Vergleich des letzten mit dem aktuellen Interval (2 Sekunden)

t_2 : kleiner Timeout, um das Ende der aktiven Phase zu erkennen (1 Sekunde)

Implementierung (I)

Struktureller Aufbau

Periodische Idle Überprüfung
Power Task

Datenstrukturupdate bei Plattenzugriff

Implementierung (II)

Ansatzpunkte im IDE Treiber

IDE-Treiber in struct beschrieben

`ide_disk.c`:

Init-Funktion Aufruf ergänzt

Request-Funktion Aufruf ergänzt

Implementierung (III)

Scheduling und Synchronisierung

Task bei init erzeugen
Blockieren über Mutex

```
WAKE_POWER_TASK(mode):  
    if (try_lock(caller)) {  
        new_power_mode = mode;  
        release(thread);  
    }
```

```
GO_TO_SLEEP():  
    release(caller);  
    lock(thread);
```

Aufgetretene Probleme

- Nach Aufgabe in ide implementiert (nicht libata)
- Race Conditions
- Kein blockieren in timer Interrupt
- Testrechner lassen sich remote ausschalten

Auswertung (I)

Testaufbau:

```
use strict;
```

```
my $offset = 0;
```

```
my $blocks = 512;
```

```
my $blocksize = 256;
```

```
while (1)
```

```
{
```

```
    system("dd if=/dev/hdc of=/dev/null bs=$blocksize  
count=$blocks " .
```

```
    "ibs=$blocksize skip=$offset");
```

```
    $offset += $blocks;
```

```
    sleep(30);
```

```
}
```

break_even_period = 10 sek.

Auswertung (II)

Durchschnittliche Leistungsaufnahme:

Ohne: 0.88018

DDT: 0.77967 → ca. 11,5%

DDT/ES: 0.72251 → ca. 17,9%