

Frequency Scaling mit PAST

auf einem Intel PXA Board

Powermanagement Praktikum

09.02.09

Andreas Leppert

Christian Helmer

Aufgabenstellung

- Frequency Scaling
 - Auf dem PXA Board
 - Mittels PAST Algorithmus
- Linux Scheduler anpassen
 - Aktuelle Auslastung bestimmen
 - CPU Frequenz entsprechend ändern

PAST Algorithmus

- PAST
 - Betrachtet einzelne Zeitfenster
 - Gewähltes Zeitfenster: $2s$
 - Eliminiert idle-Zyklen
 - Annahme
 - System verhält sich im kommenden Zeitfenster wie im letzten

Implementierung

- Auslastung bestimmen

- Running und Idle-Zyklen zählen

```
if (prev == rq->idle)
    idle_cycles += runtime;
else
    run_cycles += runtime;
```

- Prozentuale Auslastung: run_percent

- Excess-Zyklen

- Übrige Zyklen aus dem letzten Intervall, da die CPU zu langsam lief

- Ergebnis: neue Frequenz einstellen

```
if(excess_cycles > idle_cycles)
    speed = 100;
else if(run_percent > 70)
    speed += 20;
else if(run_percent < 50)
    speed = speed - (60 - run_percent);
```

Schwierigkeiten

- Scheduler komplex

- Hard-Idle und Soft-Idle Zyklen

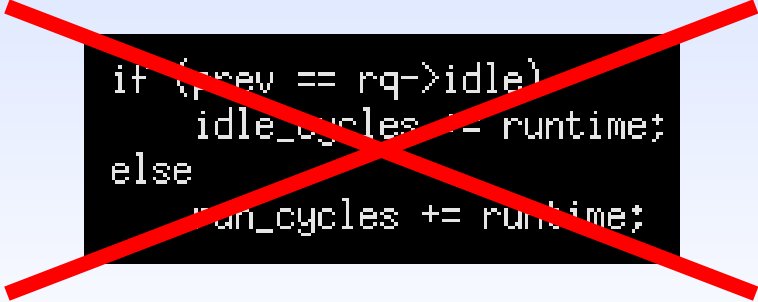
- Hard-Idle Zyklen

- Warten auf Festplatte

- Soft-Idle Zyklen

- Warten auf User Input

- Tatsächlich eliminierbare Zyklen



```
if (prev == rq->idle)
    idle_cycles += runtime;
else
    run_cycles += runtime;
```

- Hard und Soft-Idle Zyklen unterscheidbar

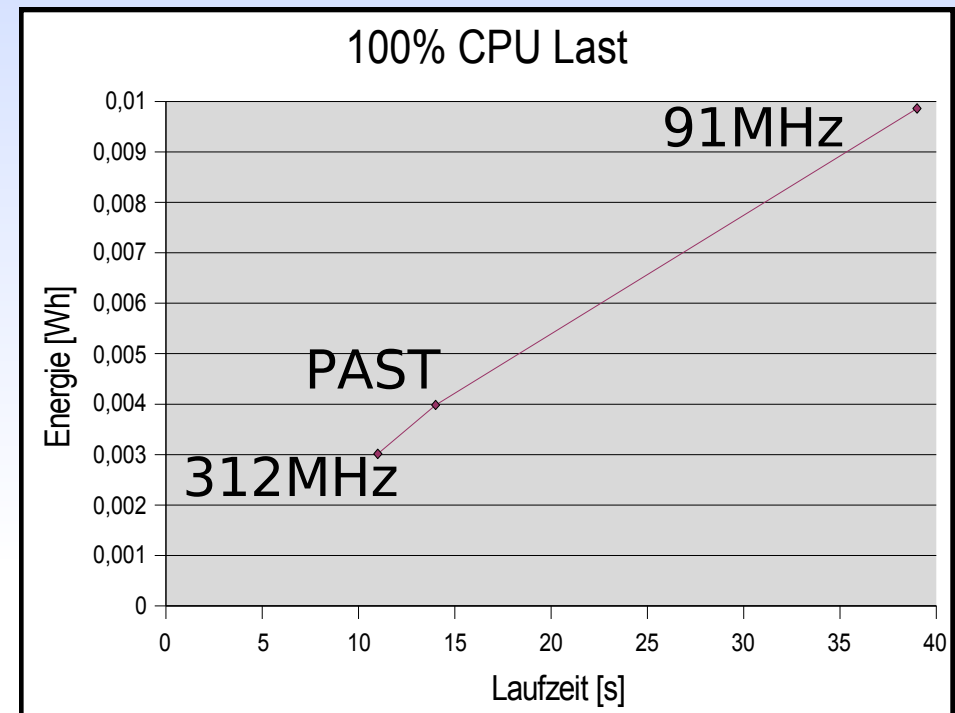
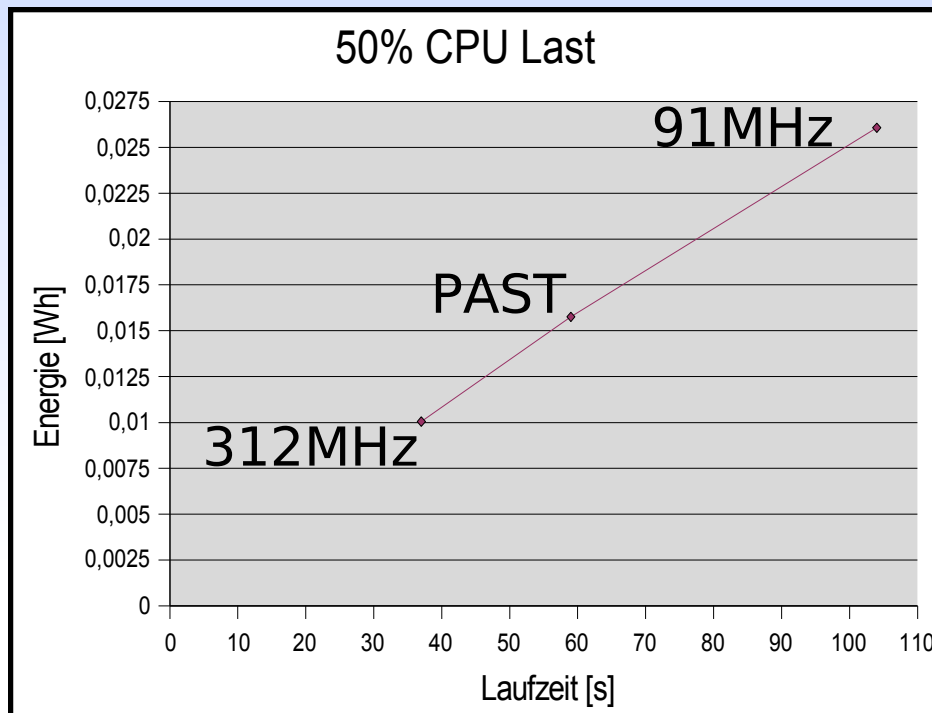
- Tasks INTERRUPTIBLE oder UNINTERRUPTIBLE

- Liste für UNINTERRUPTIBLE Tasks

- Laufzeit im UNINTERRUPTIBLE-Status bestimmen

Ergebnisse

- Energieverbrauch eines Testprogramms
 - Idle Phasen und Arbeitsphasen



- Stromverbrauch des Gesamtsystems, nicht nur CPU

Ergebnisse

- Energieverbrauch im Idle Zustand

