

# PAST-Algorithmus

PM1

Martin Riedlberger  
Tobias Modschiedler

# Aufgabe

- Frequency Scaling auf *Intel Atom N270*
  - 4 Taktraten zwischen 1600 und 800 MHz
- Algorithmus PAST von M. Weiser et al. [1]
  - Ziel: Eliminierung von Idlephasen
  - Unterscheidung der Idle-Art
    - Soft Idle : eliminierbar
    - Hard Idle (I/O): nicht eliminierbar
- Annahme: Auslastung im nächsten Intervall ähnlich zu letztem

# Implementierung

- als eigener CPUFreq-Governor
  - `drivers/cpufreq/cpufreq_past.c`
- darin: Timer (`struct timer_list`)
  - holt in festem Intervall Längen der aktiven und Idle-Phasen
  - setzt Frequenz abhängig vom Verhältnis aktiv/idle

# Daten

- Quelle: `struct cpu_usage_stat`  
`cpustat`
- Scheduler führt dort Statistik
  - Soft Idle: `cputime64_t idle`
  - Hard Idle: `cputime64_t iowait`
- Auflösung also  $\frac{1}{250}$  s (jiffies)
  - für unsere Zwecke ausreichende Genauigkeit

# Problem: Fließkomma

- keine Unterstützung für Fließkomma-Variablen im Linux-Kernel
- bräuchten wir für Anteil idle/aktiv
- Lösung: verwende Integer-Variablen mit Prozentwerten
  - und rechne entsprechend mit Faktor  $\frac{1}{100}$

# Problem: excess

- Teil des PAST-Algorithmus
  - $\text{next\_excess} = \text{run\_cycles} - \text{speed} * (\text{run\_cycles} + \text{soft\_idle})$
  - Übertrag in nächstes Intervall, weil CPU zu langsam eingestellt war, um Last zu bewältigen
- Problem: „[...] the actual version of PAST proposed by by Weiser et al. is not implementable [...]“ (aus [2])

# Problem: excess (2)

- Infos über Last benötigt
  - „Wieviel Arbeit hätte in den letzten Intervallen erledigt werden müssen?“
- bei Weiser et al. vorhanden
  - Tracing + Offline-Analyse
- hier: Scheduler kennt Menge „unfertiger“ Arbeit nicht

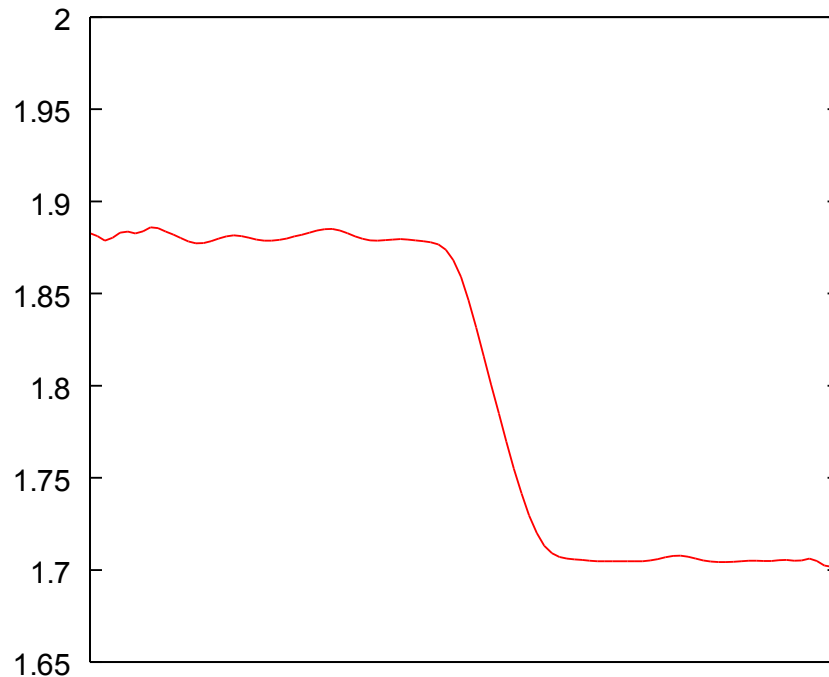
# Problem: excess (3)

- unser Workaround:
  - excess außer Acht lassen 😊
- Ergebnis: Frequenzanpassung funktioniert annehmbar



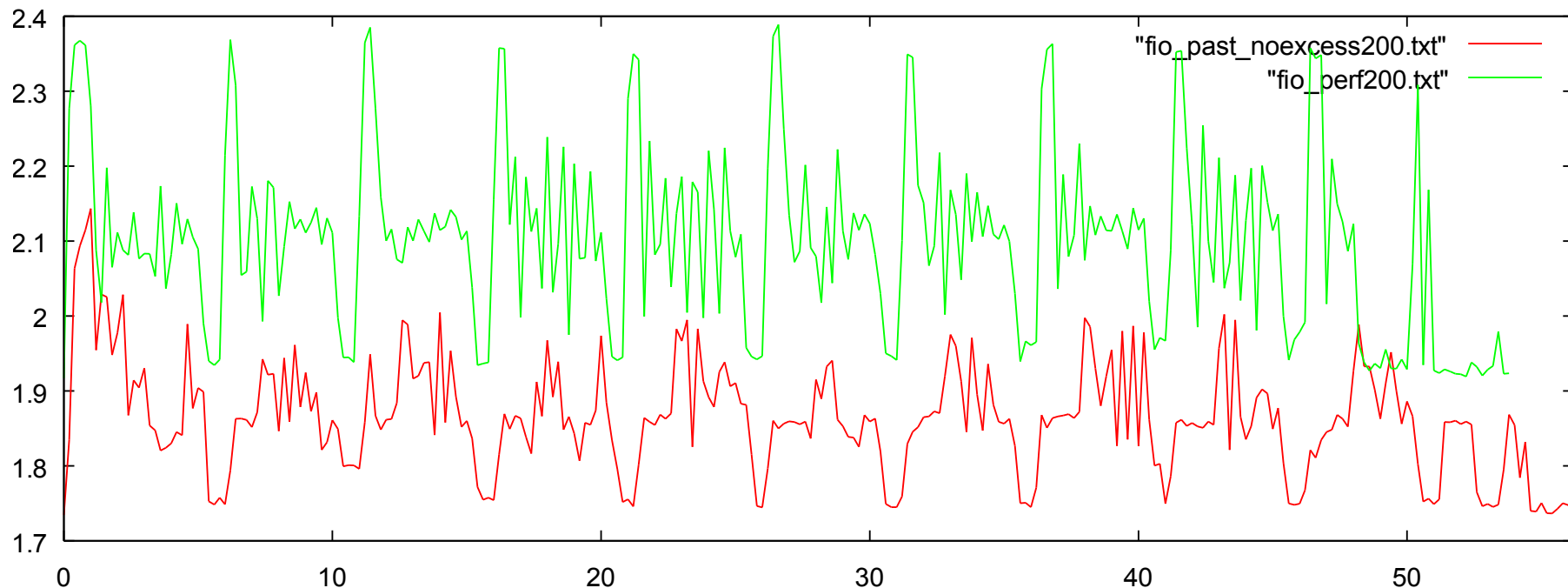
# Messergebnisse

- Energieunterschied beim Übergang von 1,6 GHz zu 800 MHz (CPU Idle)



# Messergebnisse (2)

- Benchmark: `fio`  
(„Intel IOMeter File Server Access Pattern“)
- viel I/O, aber auch CPU-Last



# Literatur

- [1] Scheduling for Reduced CPU Energy  
M. Weiser et al., OSDI 1994  
[http://www.usenix.org/publications/library/proceedings/osdi/full\\_papers/welch.ps](http://www.usenix.org/publications/library/proceedings/osdi/full_papers/welch.ps)
- [2] Policies for Dynamic Clock Scheduling  
D. Grunwald et al., OSDI 2000  
[http://www.usenix.org/events/osdi00/full\\_papers/grunwald/grunwald.pdf](http://www.usenix.org/events/osdi00/full_papers/grunwald/grunwald.pdf)