

AKBP/ES Team 1

17.02. - 01.03.2002

Thermo Throttling eines
PIII 1GHz auf 50° C

Überblick

1. Auslesen des Temperatursensors
2. Performance Counter des PIII
 1. TSC, Unhalted Cycles, DATA_MEM_REFS
 2. Virtuelle Performance Counter
3. Scheduler
 1. Temperaturgrenzen
 2. Throttling
4. Beispiele

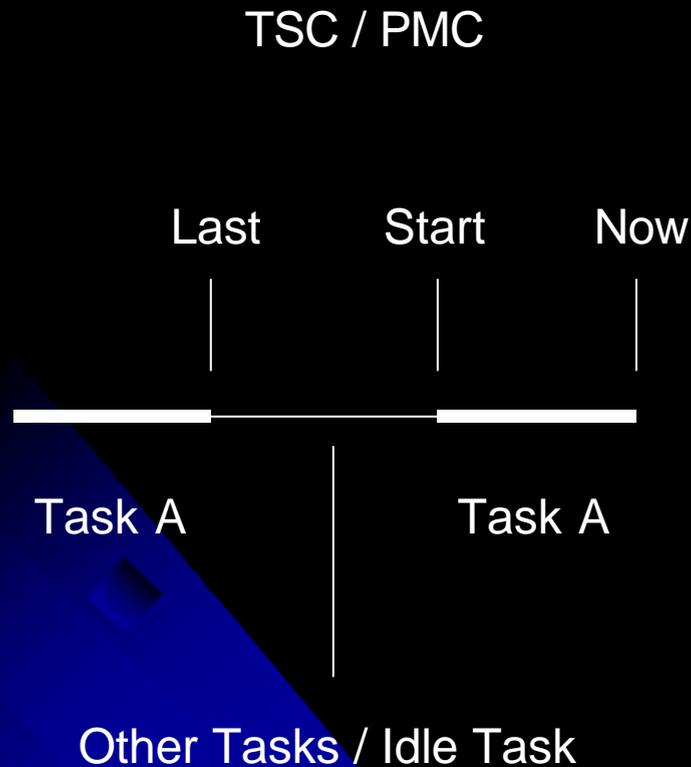
1. Temperatur

- Paket “Im_sensors-2.6.2”
- Direktes Auslesen des Registers
- Kein blockierender Aufruf
- Werte digital in “°C * 256”

2.1 Performance Counter

- Paket “perfcounter-2.4.3”
- PIII hat 2 PMC und 1 TSC
- Auf über 40 verschiedene Ereignisse programmierbar
- Grundsätzlich für alle Prozesse ein Gesamtergebnis
- Unhalted Cycles und DATA_MEM_REFS

2.2 Virtuelle Performance Counter



- Pro Task
- Selbständiges Auslesen
- Berechnen des Prozentualen Anteils
- Mitteln der Peaks

3.1 Scheduler – Temp. Grenzen



- Temp OK: kein Throttling
- Throttle 1: Throtten von Nice ≥ 0
- Throttle 2: Alle Prozesse Throtten
- Temp Crit.: Idle Task

3.2 Scheduler – Ticks

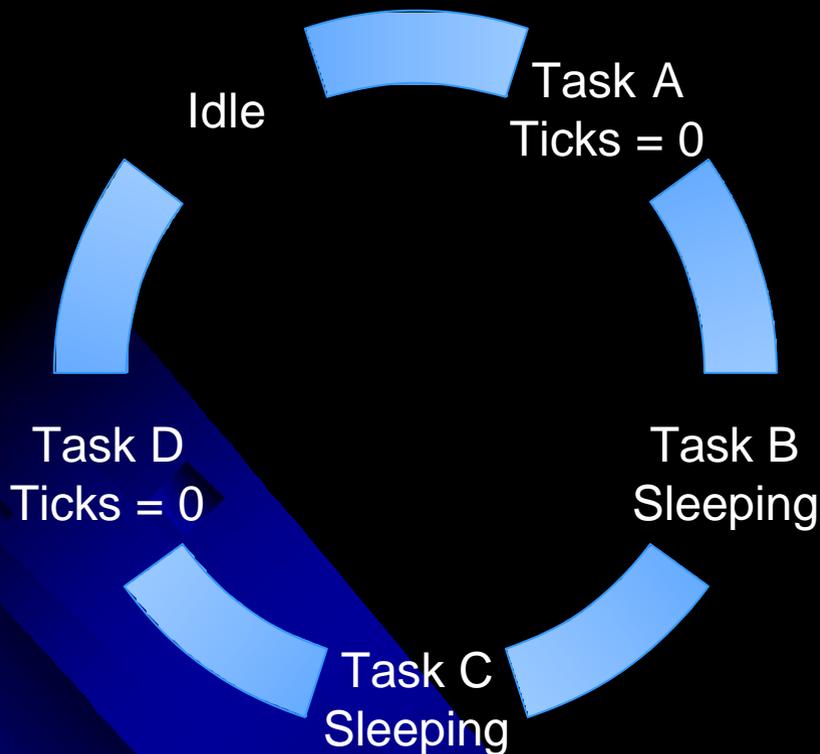
- `NICE_TO_TICKS(nice)`
- $\text{Ticks} = ((20 - \text{nice}) + 1)^2$
- Zwischen 0 und 1600 (entspricht einer Zeitscheibe von ca. 50ms bei 1 GHz)
- Durch teilen der Ticks bleibt aber die grundsätzliche Priorität innerhalb einer Gruppe erhalten

3.2 Scheduler – Throttling

PMC / TSC	Temp OK	Throttle 1	Throttle 2	Temp Crit
> 30% nice >=0	keine Aktion	1/2 Ticks	1/2 Ticks	Idle Task
> 50% nice >=0	keine Aktion	1/3 Ticks	1/3 Ticks	Idle Task
> 80% nice >=0	keine Aktion	0 Ticks	0 Ticks	Idle Task
> 30 % nice < 0	keine Aktion	keine Aktion	1/2 Ticks	Idle Task
> 50% nice < 0	keine Aktion	keine Aktion	1/3 Ticks	Idle Task
> 80% nice < 0	keine Aktion	keine Aktion	0 Ticks	Idle Task

3.2 Scheduler – Round Robin

- Wenn nur noch energiefressende Prozesse, die auf 0 gesetzt wurden, übrig sind, wird automatisch zum Idle Task gewechselt.



4. Beispiele - Quellcode

```
/* Drosselung fuer aktuellen Thread */
if (temp >= temp_hyst)
{
    if (curr_task->thread.perfctr != NULL){
        if ((curr_task->nice >= 0) || (temp >= temp_over)) {
            if (MITTELWERTE[2].mittelwert > 80 || MITTELWERTE[1].mittelwert > 80 ) /* ueber 80 % der Taktzyklen od DATA_MEM_REF verbraucht */
            {
                curr_task->counter = 0; /* diese Runde gar nicht mehr */
                keep=0;

#ifdef DEBUG
                printk("%s total ausgebremst\n",curr_task->comm);
#endif
            }
            else if (MITTELWERTE[2].mittelwert > 50 || MITTELWERTE[1].mittelwert > 50 ) /* ueber 50 % der Taktzyklen od DATA_MEM_REF verbraucht */
            {
                curr_task->counter /= 3; /* drittel Laufzeit */
                keep=0;

#ifdef DEBUG
                printk("%s 2/3 ausgebremst\n",curr_task->comm);
#endif
            }
            else if (MITTELWERTE[2].mittelwert > 30 || MITTELWERTE[1].mittelwert > 30 ) /* ueber 30 % der Taktzyklen od DATA_MEM_REF verbraucht */
            {
                curr_task->counter /= 2; /* halbe Laufzeit */
                keep=0;

#ifdef DEBUG
                printk("%s 1/2 ausgebremst\n",curr_task->comm);
#endif
            }
        }
    }
}
}
```

4. Beispiele - Quellcode

```
/* Pruefen, ob Nenner != 0 */
if ((PMCO_NOW - PMCO_LAST)!=0 && (PMC1_NOW - PMC1_LAST)!=0 && (TSC_NOW - TSC_LAST)!=0)
    { update_mittelwerte(curr_task, 0, 1, TSC_PERCENT, PMCO_PERCENT, PMC1_PERCENT); }
#ifdef DEBUG
    else
        { printk("%s New task! (nenner = 0)\n", curr_task->comm); }
#endif
```

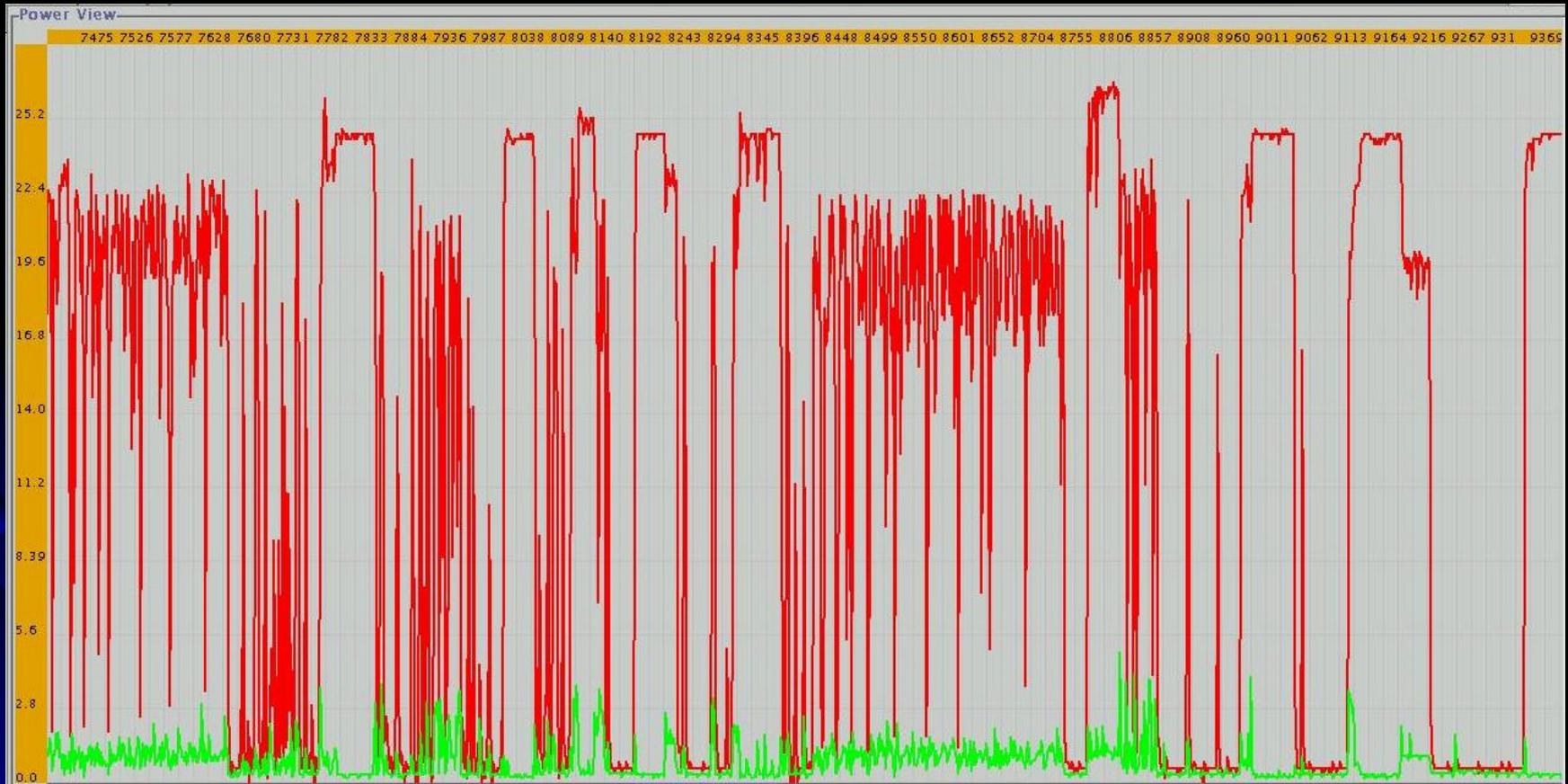
4. Beispiele – PowerView



4. Beispiele - Syslog

```
Mar 1 01:26:19 thermo kernel: Temperature phase change to Status OK
Mar 1 01:26:24 thermo kernel: Temperature phase change to Status THROTTLE 1
Mar 1 01:26:42 thermo kernel: scheduler module 'temp scheduler round robin team1' unloaded
Mar 1 01:26:51 thermo kernel: scheduler module 'temp scheduler round robin team1' loaded
Mar 1 01:26:51 thermo kernel: Temperature phase change to Status CRITICAL
Mar 1 01:26:56 thermo kernel: Temperature phase change to Status THROTTLE 1
Mar 1 01:27:16 thermo kernel: Temperature phase change to Status OK
Mar 1 01:27:21 thermo kernel: Temperature phase change to Status THROTTLE 1
Mar 1 01:27:26 thermo kernel: Temperature phase change to Status OK
```

4. Beispiele - PowerView

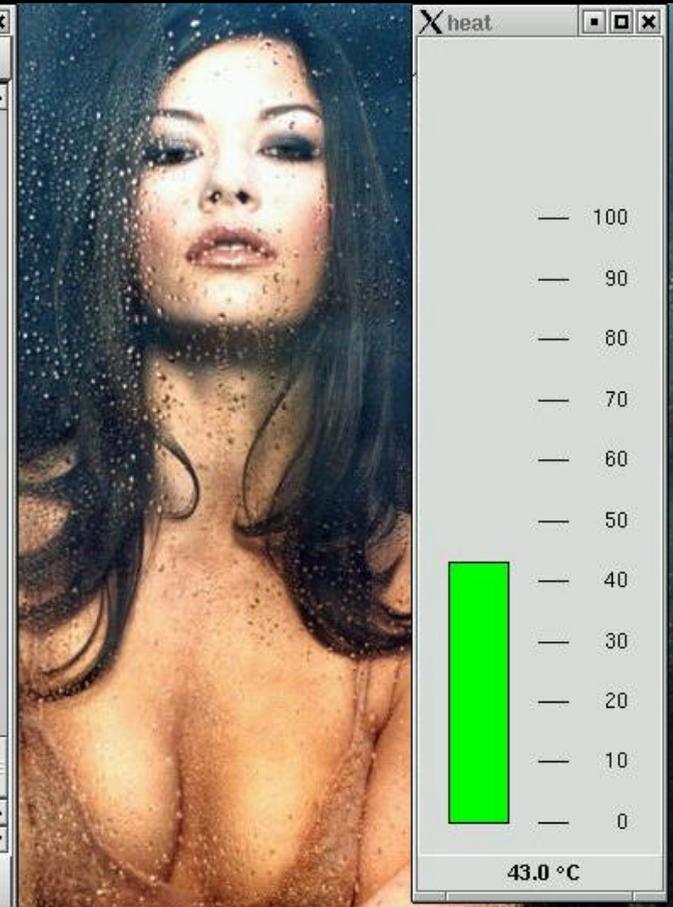


4. Beispiele – TOP + Heat

```
top - Konsole
File Sessions Settings Help

2:19pm up 3:45, 4 users, load average: 7.28, 4.86, 3.51
79 processes: 69 sleeping, 10 running, 0 zombie, 0 stopped
CPU states: 50.2% user, 5.3% system, 0.0% nice, 44.3% idle
Mem: 255928K av, 252016K used, 3912K free, OK shrd, 1932K buff
Swap: 393584K av, 1464K used, 392120K free

  PID USER      PRI  NI  SIZE  RSS  SHARE STAT %CPU %MEM    TIME COMMAND
 3240 root        10  -19  3708  3704  1920 S <    8.9  1.4   0:37 mplayer
 3433 root         0   0   304   304   248 R     7.8  0.1   0:13 x
 3394 root         0   0   304   304   248 R     6.8  0.1   0:27 x
 3393 root         0   0   304   304   248 R     6.4  0.1   0:29 x
 3247 root         0   0   304   304   248 R     6.1  0.1  13:17 x
 3434 root         0   0   304   304   248 R     6.1  0.1   0:12 x
 3430 root         0   0   304   304   248 R     5.5  0.1   0:15 x
 3431 root         0   0   304   304   248 R     5.5  0.1   0:15 x
 1209 root         0   0  9960  9600  7944 R     0.9  3.7   0:46 kdeinit
 1039 root        10  -19  57620 8000  1816 S <    0.5  3.1   3:23 X
 1400 root         0   0  1576  1576  1176 S     0.1  0.6   0:00 bash
 3349 root         0   0  1912  1892  1384 S     0.1  0.7   0:00 sshd
 3437 sithg:lan    14   0   992   992   776 R     0.1  0.3   0:00 top
   1 root         0   0   208   208   176 S     0.0  0.0   0:06 init
   2 root         0   0     0     0     0 SW    0.0  0.0   0:00 keventd
   3 root         0   0     0     0     0 SW    0.0  0.0   0:00 kapm-idled
   4 root        19  19     0     0     0 SWN   0.0  0.0   0:00 ksoftirqd_CPU0
   5 root         0   0     0     0     0 SW    0.0  0.0   0:00 kswapd
   6 root         9   0     0     0     0 SW    0.0  0.0   0:00 bdflush
   7 root         0   0     0     0     0 SW    0.0  0.0   0:00 kupdated
  247 root         0   0   484   484   416 S     0.0  0.1   0:00 dhcpcd
  367 root         0   0  1048  1008   904 S     0.0  0.3   0:00 sshd
  383 root         0   0   632   628   520 S     0.0  0.2   0:00 syslogd
  386 root         0   0   944   936   460 S     0.0  0.3   0:00 klogd
  410 bin          9   0   440   428   360 S     0.0  0.1   0:00 portmap
  436 root         9   0   708   700   604 S     0.0  0.2   0:00 rpc.statd
  438 root         0   0     0     0     0 SW    0.0  0.0   0:00 rpciod
```



FAZIT

Wer später bremst
rechnet länger schnell !!

;-)