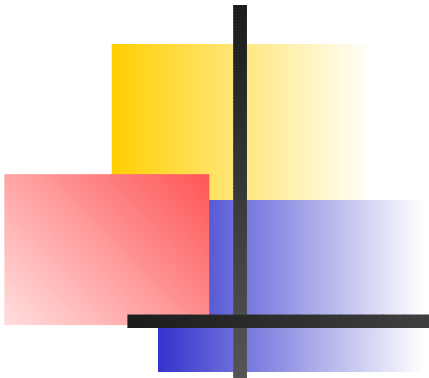

AKBP/ES II Team 2

Thermothrottling

Wolfgang Kroworsch

siwokrow@stud.uni-erlangen.de

Florian E.J. Fruth

siflfrut@stud.uni-erlangen.de

FAU Erlangen-Nürnberg

Aufgabe

Drosselung einer CPU

- CPU soll trotz fehlenden Lüfters eine festgelegte Temperatur nicht überschreiten.
- interaktive Prozesse sollen bevorzugt, rechenintensive Prozesse ausgebremst werden

Temperatur-Messung

- Temperatur-Messung mit Sensor an der CPU über I^2C-Bus
- Aktuelle Temperatur steht nur fünf bis zehn mal pro Sekunde zur Verfügung

$\Rightarrow$ Interpolation der Temperatur dazwischen durch Energiemessung

Energiemessung

Verwendung von Performance-Countern (z.B. Prozessor-Zyklen, Speicherzugriffe . . .) des Pentium III Prozessors zur Messung des Energieverbrauchs:

- global zur Verhinderung zu starker Erwärmung
- pro Prozess zur gerechten Verteilung der Energie

Energiemessung

Problem:

- Es gibt nur Prozessor-Globale Performance-Counter

Lösung:

- virtuelle Performance Counter pro Prozess (Patch zum Accounting der Prozesse bei jedem Re-Scheduling bzw. Timer-Interrupt)

Einbindung

Priorisierung der Tasks durch neuen Energie-bewußten Scheduler

- Dynamisches Accounting der Energie zu Prozessen bei jedem Timer-Interrupt
- Einführung von `halt()`-Zyklen bzw. idle-tasks zur Kühlung
- Einteilung in 3 Prioritäts-Queues damit möglichst nur unwichtige Prozesse gedrosselt werden

Prioritäten

Festlegen der Prioritäten

- durch SystemCalls
- Vererbung von Prioritäten
- oder: Standardmäßiges Starten von Tasks in niederpriorer Queue und Anheben durch Verwaltungs-Prozess
- Wrapper: `setqueue [queue] [pid]`

Energiekontingente

Zuweisen von Kontingenten zu Prioritäten

- jede Prioritätsklasse erhält ein maximales Energie-Kontingent für einen vorgegebenen Zeitraum
- innerhalb der Prioritätsklassen zählt das Prinzip first-come-first-serve
- wird die CPU zu heiß wird zuerst das Kontingent reduziert
- ab einer Notfallgrenze werden keine Prozesse mehr ausgeführt

Wärmeerzeugung

Einfaches Program nimmt sich nahezu die komplette CPU

```
int main(){
  register int i; register int j; register int k;

  for(j=1;1;j++) {
    for(i=1;i<250000000;i++)
      k*=3;
    printf("ping #%10.0ld\n",j);
    k=1;
  }
}
```

$\Rightarrow$ Temperaturanstieg von 37,5 Grad auf 52 Grad Celsius (bei normalem Linux-Scheduler)

Wärmevermeidung

Wenn die CPU zu warm wird muss Energie gespart werden

- möglich durch Einlegen von `halt()` Zyklen bzw. durch Idle-Task
- Überprüfen von prozessinternen maximalen PerformanceCountern pro Prozess
  $\Rightarrow$ wenn diese überschritten werden, wird die entsprechende Task nicht mehr ausgeführt
- Neuverteilung von Energie-Kontingenten erst nach Ablauf des Festgelegten Zeitraums

Zukunft

Erweiterungen

- höherpriore Prozesse können Energie aus niederprioren Klassen verbrauchen, falls diese dort nicht benötigt wird
- automatische Erkennung von interaktiven Prozessen und Zuweisung der Priorität 1
- most-energy-left first-serve strategie

3 Prozesse

```
 11:15am  up 56 min,  7 users,  load average: 4.91, 8.15, 8.54
63 processes: 54 sleeping, 6 running, 3 zombie, 0 stopped
CPU states: 20.4% user,  0.0% system,  0.0% nice, 79.6% idle
Mem:  1029656K av,  131060K used,  898596K free,       0K shrd,    4740K buff
Swap: 1028120K av,       0K used, 1028120K free                   96412K cached

  PID USER     PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM   TIME COMMAND
 1561 siwokrow  20   0   348  348   288 R    14.4  0.0   1:30 new_arith
  979 siwokrow  18   0   348  348   288 R     3.2  0.0   2:09 new_arith
 1037 root      18   0   348  348   288 R     2.6  0.0   2:07 new_arith
  980 root      18   0  1816 1816  1392 R     0.2  0.1   0:00 sshd
    1 root      17   0   208  208   176 S     0.0  0.0   0:06 init
    2 root       9   0     0    0     0 SW    0.0  0.0   0:00 keventd
    3 root      18  19     0    0     0 SWN   0.0  0.0   0:00 ksoftirqd_CPU0
    4 root       9   0     0    0     0 SW    0.0  0.0   0:00 kswapd
    5 root       9   0     0    0     0 SW    0.0  0.0   0:00 bdflush
    6 root      17   0     0    0     0 SW    0.0  0.0   0:00 kupdated
  225 root       9   0   484  484   416 S     0.0  0.0   0:00 dhcpcd
  357 root       9   0  1048 1048   920 S     0.0  0.1   0:00 sshd
  372 root      17   0   632  632   520 R     0.0  0.0   0:00 syslogd
  375 root      17   0  1064 1064   448 S     0.0  0.1   0:00 klogd
  411 bin        9   0   436  436   356 S     0.0  0.0   0:00 portmap
  434 root       9   0   712  712   604 S     0.0  0.0   0:00 rpc.statd
  436 root       9   0     0    0     0 SW    0.0  0.0   0:00 rpciod
```

9 Prozesse

```
 10:39am  up 20 min, 13 users,  load average: 9.37, 6.40, 3.66
79 processes: 66 sleeping, 12 running, 1 zombie, 0 stopped
CPU states: 23.7% user,  0.0% system,  0.0% nice, 76.2% idle
Mem:  1029656K av,  139188K used,  890468K free,       0K shrd,    4740K buff
Swap: 1028120K av,       0K used, 1028120K free                   96324K cached

  PID USER     PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM   TIME COMMAND
 1037 root      20   0   348  348   288 R     4.7  0.0   0:45 new_arith
 1597 siwokrow  20   0   348  348   288 R     3.3  0.0   0:23 new_arith
  979 siwokrow  20   0   348  348   288 R     3.1  0.0   0:48 new_arith
 1491 siwokrow  20   0   348  348   288 R     3.1  0.0   0:30 new_arith
 1561 siwokrow  18   0   348  348   288 R     2.5  0.0   0:23 new_arith
 1629 siwokrow  20   0   348  348   288 R     2.3  0.0   0:17 new_arith
 1663 siwokrow  20   0   348  348   288 R     2.3  0.0   0:14 new_arith
 1696 siwokrow  20   0   348  348   288 R     2.3  0.0   0:11 new_arith
 1733 siwokrow  20   0   348  348   288 R     1.7  0.0   0:08 new_arith
 1772 root      18   0   996  996   776 R     0.1  0.0   0:00 top
    1 root      17   0   208  208   176 S     0.0  0.0   0:06 init
    2 root       9   0     0    0     0 SW    0.0  0.0   0:00 keventd
    3 root      18  19     0    0     0 SWN   0.0  0.0   0:00 ksoftirqd_CPU0
    4 root       9   0     0    0     0 SW    0.0  0.0   0:00 kswapd
    5 root       9   0     0    0     0 SW    0.0  0.0   0:00 bdflush
    6 root      17   0     0    0     0 SW    0.0  0.0   0:00 kupdated
  225 root       9   0   484  484   416 S     0.0  0.0   0:00 dhcpcd
```

Ende

The End...

Wer später bremst rechnet länger schnell