

# Leistungsanalyse unter Linux

Einfache Werkzeuge zur Systemanalyse

von Norbert Schramm



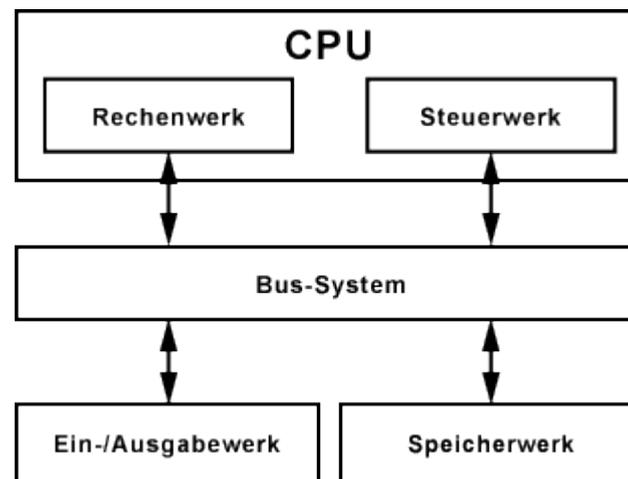
# Inhalt

- 1. Einleitung
- 2. Programme:
  - 2.1 vmstat
  - 2.2 top
  - 2.3 PowerTop
  - 2.4 vnstat
  - 2.5 iostat
  - 2.6 Gnome System Monitor
  - 2.7 andere Tools
- 3. Das /proc-Dateisystem
- 4. Zusammenfassung



# 1. Einleitung

- Grundlegende Funktion für Bewertung der Leistungsfähigkeit/Auslastung eines Systems
- Ziel: Überwachung auf Normalzustand, Auffinden von Fehlern oder Bottlenecks
- Unterschiedliche Ansätze/Schnittstellen zu beobachten



## 2.1 vmstat

- System Activity, Hardware and System Information
- Übersicht über Prozesse, Arbeitsspeicher, Swap, Festplatten- und Prozessoraktivitäten
- Bietet Durchschnitts- und Echtzeitwerte
- Verschiedene Ansichten

```
stoertebeker@kastenbier: ~  
stoertebeker@kastenbier:~$ vmstat 1  
procs -----memory----- ---swap-- -----io----- -system-- ----cpu----  
r  b   swpd  free  buff  cache   si   so    bi    bo    in   cs  us  sy  id  wa  
5  0   15984 483596 223188 2459996    0    0    13    22     1    0  92   7   1   0  
4  0   15984 483464 223188 2459996    0    0     0     0 2445 4195 96   4   0   0  
5  0   15984 483216 223188 2459996    0    0     0     0 2381 4080 93   7   0   0  
4  0   15984 483092 223188 2459996    0    0     0     0 2478 4192 92   8   0   0  
5  0   15984 483092 223188 2459996    0    0     0    284 2447 4079 91  10   0   0  
4  0   15984 483092 223188 2459996    0    0     0     0 2439 4191 92   9   0   0  
^C  
stoertebeker@kastenbier:~$
```



# 2.1 vmstat

- Livedemo:
  - Hohe CPU-Userlast
  - Hohe CPU-Systemlast
  - Hohe I/O-Schreiblast
  - CPU wartet auf I/O



## 2.2 Top

- Process Activity Command
- Überblick über laufende Prozesse, nach CPU-Last sortiert
- Seit 1984 open-source, weite Verbreitung
- Echtzeitausgabe in Zeitintervall
- Interaktion mit Prozessen möglich

```
stoertebeker@kastenbier: ~
Tasks: 141 total,  1 running, 140 sleeping,  0 stopped,  0 zombie
Cpu(s):  3.2%us,  7.2%sy, 88.8%ni,  0.7%id,  0.1%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   3797156k total,  3301112k used,   496044k free,   223188k buffers
Swap: 10485756k total,   15984k used, 10469772k free,   2460032k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
13411	fath	39	19	275m	76m	3048	S	387	2.1	2088:32	FahCore_a3.exe
19395	server	20	0	1324m	163m	9764	S	8	4.4	118:00.59	java
2305	root	20	0	0	0	0	S	2	0.0	12:20.30	kworker/3:1
3753	root	20	0	0	0	0	S	2	0.0	25:28.54	kworker/2:1
5852	server	20	0	198m	3624	1616	S	2	0.1	446:59.96	ts3server_linux
1	root	20	0	24116	1876	1048	S	0	0.0	0:23.58	init
2	root	20	0	0	0	0	S	0	0.0	0:00.30	kthreadd
3	root	20	0	0	0	0	S	0	0.0	0:13.60	ksoftirqd/0
6	root	RT	0	0	0	0	S	0	0.0	0:00.00	migration/0



## 2.2 Top

- Drei Ziele der Entwickler:
  - provide an accurate snapshot of the system and process state
  - not be one of the top processes itself
  - be as portable as possible
- Livedemo:
  - Übersicht
  - Zusätzliche Spalten
  - Filter, Prozessverwaltung



## 2.3 PowerTop

- Aktive Prozesse verbrauchen mehr Strom
  - Verschiedene Stromsparszustände in CPUs
- Bietet Überblick über Interrupts, die CPU im idle wecken
- Entwicklungsgrund: 1000 Hz-Problem des Kernel

```
PowerTOP version 1.13 (C) 2007 Intel Corporation

Cn          Avg residency      P-states (frequencies)
C0 (cpu running)  ( 0.2%)
polling      21.7ms ( 0.4%)
C1 mwait     108.4ms (99.3%)

Wakeups-from-idle per second : 9.4 interval: 20.0s
no ACPI power usage estimate available

Top causes for wakeups:
 14.7% ( 3.0) kworker/0:1
  9.8% ( 2.0) ksoftirqd/0
  9.6% ( 1.9) kworker/0:0
  8.1% ( 1.6) [eth0] <interrupt>
  8.1% ( 1.6) gdm-simple-gree
  5.4% ( 1.1) apache2

Suggestion: Enable SATA ALPM link power management via:
echo min_power > /sys/class/scsi_host/host0/link_power_management_policy
or press the S key.
Q - Quit R - Refresh S - SATA Link Power Management
```



## 2.3 PowerTop

- Listet Gründe für wakeups auf
- Übersicht über Häufigkeit der Nutzung der verschiedenen C-States
- Gibt Optimierungsvorschläge, welche direkt im Programm umgesetzt werden können
- Aber: Optimierungen nur bis zum nächsten Systemstart aktiv
- Benötigt root-Rechte für detailliertere Ausgabe
- Live-Demo



# 2.4 vnstat

- Konsolenbasierter Netzwerktrafficonitor
- Echtzeitauslastung
- Statistiken über verschiedene Zeiträume
- Mehrere Interfaces separat auswertbar

```
root@s15399019.onlinehome-server.info: ~  
Monitoring venet0... (press CTRL-C to stop)  
  
rx: 488 kbit/s 1470 p/s      tx: 32.93 Mbit/s 2910 p/s  
  
venet0 / traffic statistics  
  
              rx      |      tx  
-----  
bytes          1.53 MiB | 102.03 MiB  
-----  
  max          1.08 Mbit/s | 72.01 Mbit/s  
  average      543.30 kbit/s | 36.34 Mbit/s  
  min           4 kbit/s | 4 kbit/s  
-----  
packets        38106 | 73829  
-----  
  max          3298 p/s | 6355 p/s  
  average      1656 p/s | 3209 p/s  
  min           6 p/s | 6 p/s  
-----  
time                23 seconds  
root@s15399019:~#
```



## 2.4 vnstat

- Ausgabe in Stunden, Tagen, Wochen, Monate
- „Top10“ Liste
- Interfaces einzeln, kombiniert auswertbar
- Mit „vnstati“ grafische Ausgabe der Statistiken
- Live-Demo
  - Normalansicht
  - Echtzeitauswertung
  - Verschiedene Statistiken
  - Grafische Statistiken



## 2.5 iostat

- Speicherhierarchie im Computer
- iostat: ternäre Ebene → Festplatten
- Echtzeitüberwachung sowie Mittelwerte
- I/O für Input/Output

```
stoertebeker@kastenbier: ~  
stoertebeker@kastenbier:~$ iostat 1 2  
Linux 3.0.0-14-server (kastenbier)      25.03.2012      _x86_64_      (4 CPU)  
  
avg-cpu:  %user   %nice %system %iowait  %steal   %idle  
           3,18   88,87   7,17   0,11   0,00   0,67  
  
Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn  
sda                 4,07         51,09         88,21  308745608  533116304  
sdb                 0,00          0,01          0,00    47329         0  
  
avg-cpu:  %user   %nice %system %iowait  %steal   %idle  
           1,00   94,49   4,51   0,00   0,00   0,00  
  
Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn  
sda                 0,00          0,00          0,00         0         0  
sdb                 0,00          0,00          0,00         0         0  
stoertebeker@kastenbier:~$
```



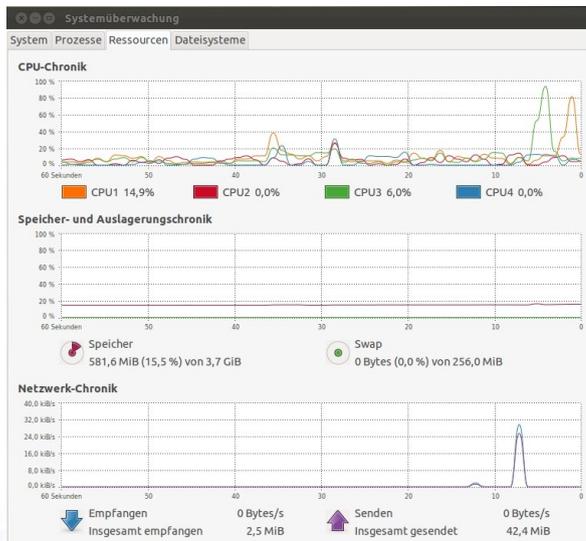
## 2.5 iostat

- Drei Schwerpunkte:
  - CPU-Auslastung
  - Festplattenauslastung
  - NFS-Auslastung
- Hat mit Parameter -x detaillierte Werte für eine sehr präzise Auswertung der Last
- Live-Demo:
  - Übersicht einfach/erweitert
  - Hohe Schreiblast



# 2.6 Gnome System Monitor

- X-Window Programm
- Ähnlich Taskmanager unter MS Windows
- Sehr einfache Übersicht für Endnutzer
- Echtzeitinformationen über CPU, RAM/Swap, Netzwerk und Dateisystem



The screenshot shows the 'Processes' tab of the Gnome System Monitor. It displays a table of running processes with columns for process name, status, CPU usage, nice value, PID, memory usage, and waiting channel. The table is sorted by CPU usage, with 'applet.py' at the top.

Prozessname	Status	% CPU	Nice-Wert	Kenntung	Speicher	Waiting Channel
applet.py	Schläft	0	0	2526	12,8 MiB	poll_schedule_tim
bamfdemon	Schläft	0	0	2449	3,4 MiB	poll_schedule_tim
bash	Schläft	0	0	2681	2,4 MiB	n_tty_read
bash	Schläft	0	0	1862	5,9 MiB	n_tty_read
bash	Schläft	0	0	1661	5,8 MiB	n_tty_read
bluetooth-applet	Schläft	0	0	2384	3,7 MiB	poll_schedule_tim
cat	Schläft	0	0	2517	96,0 kiB	unix_stream_data
compiz	Schläft	0	0	2360	86,3 MiB	poll_schedule_time
dbus-daemon	Schläft	0	0	2324	2,1 MiB	poll_schedule_time
dbus-launch	Schläft	0	0	2323	304,0 kiB	poll_schedule_time
dconf-service	Schläft	0	0	2377	672,0 kiB	poll_schedule_time
deja-dup-monitor	Schläft	0	0	2649	804,0 kiB	poll_schedule_time
gconfd-2	Schläft	0	0	2353	1,6 MiB	poll_schedule_time
gconf-helper	Schläft	0	0	2375	704,0 kiB	poll_schedule_time
gdu-notification-daemon	Schläft	0	0	2507	2,4 MiB	poll_schedule_time
geoclue-master	Schläft	0	0	2495	484,0 kiB	poll_schedule_time
gnome-fallback-mount-hel	Schläft	0	0	2381	2,3 MiB	poll_schedule_time
gnome-keyring-daemon	Schläft	0	0	2276	3,5 MiB	poll_schedule_time
gnome-pty-helper	Schläft	0	0	2680	824,0 kiB	unix_stream_data
gnome-screensaver	Schläft	0	0	2362	2,4 MiB	poll_schedule_time
gnome-session	Schläft	0	0	2285	2,7 MiB	poll_schedule_time
gnome-settings-daemon	Schläft	0	0	2344	9,3 MiB	poll_schedule_time



# 2.6 Gnome System Monitor

- Aufgeteilt in 4 Bereiche
  - System
  - Prozesse
  - Ressourcen
  - Dateisysteme
- Live-Demo
  - Übersicht
  - Hohe CPU-Userlast
  - Netzwerkverkehr



## 2.7 andere Tools

- W – Nutzerübersicht:
  - Übersicht über angemeldete User und was sie tun
- Uptime – Systemlaufzeit anzeigen:
  - Laufzeit seit dem Start des Systems, einfache Lastübersicht
- Free – Memory Usage:
  - Auslastung des Hauptspeichers und RAM
- Mpstat – Multiprocessor Usage:
  - Aufgeschlüsselte Übersicht über Auslastung der einzelnen Kerne



# 3. Das /proc-Dateisystem

- Datenquelle für viele Analysetools
- Kein richtiges Dateisystem mit festen Dateien
- Kernel erstellt gewünschte Daten dynamisch
- Lassen sich ohne root-Rechte auslesen
- Neben allgemeinen Systemwerten hat jeder Prozess eigenen Ordner

```
root@s15399019.onlinehome-server.info: /proc
root@s15399019: /proc# ls
1      1546  3676  3682   devices      meminfo      swaps        version
1146   1567  3677  4010   filesystems   modules      sys          vmstat
1191   3644  3678  5321   fs            mounts       sysrq-trigger  vz
1288   3655  3679  5380   kmsg          net          sysvipc
1289   3673  3680  cmdline loadavg       self         uptime
1333   3675  3681  cpuinfo locks         stat         user_beancounters
root@s15399019: /proc#
```



# 4. Zusammenfassung

Vorgestelltes Programm	Einsatzgebiet	GUI vorhanden, benötigt root-Rechte, Echtzeitfähig
vmstat	Allg. Systemüberwachung	Nein, nein, ja
top	Prozessüberwachung	Nein, nein, ja
powertop	Energieverbrauch-Analyse	Nein, ja, ja
vnstat	Netzwerktrafficonitor	Nein, nein, ja
iostat	Festplattenauslastung	Nein, nein, ja
Gnome System Monitor	Allg. Systemüberwachnug	Ja, nein, ja



# 4. Zusammenfassung

- Systemanalyse so alt wie Computersysteme selbst
- Nicht nur für Sys-Admins von Nutzen
- Viele Informationen über mehrere Wege beschaffbar
- Grafische Programme gut, aber konsolenbasierte Ausgabe meist schneller, Daten können ausgegeben und weiterverarbeitet werden
- Entwicklung weg von Einzelsystemanalyse hin zu verteilten Systemen (Cloud, HPC)
- Die Analyse geht mit der Entwicklung:



# Vielen Dank für die Aufmerksamkeit



- Fragen?

