

Universität Hamburg

Fakultät für Informatik

Seminar Leistungsanalyse unter Linux
im WS 11/12

Thema
einfache Tools zur Systemanalyse

Norbert Schramm

E-Mail: 9schramm@informatik.uni-hamburg.de

Matrikelnummer: 6146299

Studienfach und Semester: Informatik B.Sc., 5. Fachsemester

Inhaltsverzeichnis

1. Einleitung	
1.1 Motivation (warum ist das so wichtig, was ist das überhaupt).....	3
1.2 Arten der Systemanalyse.....	3
2. vmstat - System Activity, Hardware and System Information	
2.1 Funktion & Parameter.....	4
2.2 Möglichkeiten der Analyse.....	5
2.3 Lastbeispiele.....	5
3. top - Process Activity Command	
3.1 Übersicht.....	6
3.2 Funktionen.....	8
4. PowerTOP – power usage analysis	
4.1 Einsatzbereich.....	9
4.2 Programmübersicht.....	9
4.3 Beispiel.....	10
5. vnstat – network analysis and statistics	
5.1 Einsatzzweck	11
5.2 Programmübersicht.....	11
5.3 vnstati.....	14
6. iostat - Average CPU Load, Disk Activity	
6.1 Einsatzbereich.....	14
6.2 Programmübersicht.....	15
6.3 Beispiele.....	16
7. Gnome System Monitor – Real-time Systems Reporting and Graphing	
7.1 Einsatzbereich.....	16
7.2 Übersicht.....	17
8. andere Tools	
8.1 w - Find Out Who Is Logged on And What They Are Doing.....	19
8.2 uptime - Tell How Long The System Has Been Running.....	19
8.3 free - Memory Usage.....	19
8.4 mpstat - Multiprocessor Usage.....	20
9. Das /proc-Dateisystem.....	20
10. Stellenwert der Systemanalyse & Zusammenfassung.....	22
11. Quellen.....	23

1. Einleitung

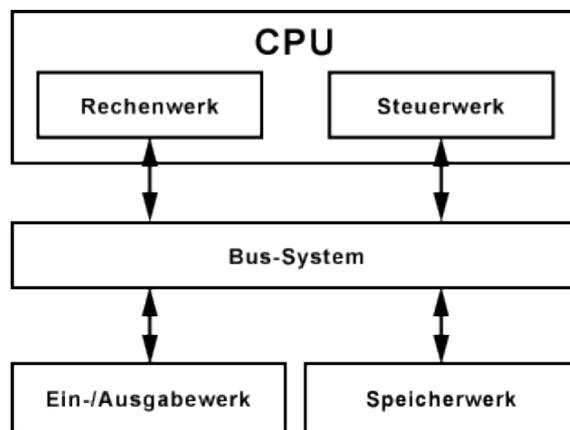
1.1 Motivation

Die Systemanalyse ist eine der grundlegenden Funktionen für die Bewertung der Leistungsfähigkeit und der Auslastung eines Systems. Sie spielt in allen Anwenderkreisen eine wichtige Rolle, egal ob im privaten Umfeld, in Netzwerksystemen oder im Highperformancebereich.

In der Informatik hat dieser Begriff noch eine weitere Bedeutung und bezieht sich auf die Spezifikation als erste Phase des Entwurfsprozesses. In dieser Seminararbeit geht es jedoch um das Auslesen, Analysieren und Verwerten von Systemparametern, meist mit dem Ziel der Leistungsoptimierung eines Systems und zur Feststellung von Bottlenecks oder überlasteter Komponenten.

1.2 Arten der Systemanalyse

Ein Computersystem besteht im einfachsten Fall wie in der von-Neumann-Architektur beschrieben aus einer CPU, einem Bussystem, einem Speicherwerk und einem Ein-/Ausgabewerk.



Alle diese Komponenten haben eine bestimmte Leistungsfähigkeit und können nur eine bestimmte Anzahl an Daten transportieren oder verarbeiten. In Folge dessen ist es auch sinnvoll, all diese Komponenten zu überwachen. Auf einer CPU müssen sich mehrere Prozesse gleichzeitig einen (oder mehrere) Rechenkern teilen, sind es zu viele auf einmal, wird die CPU zu einem Bottleneck, da hier mehr Zeit mit der Verwaltung der wartenden Prozesse verbraucht wird als mit der eigentlichen Bearbeitung. Um Prozesse auf einem Unixsystem zu überwachen eignet sich das Programm „top“ welches eine sich ständig aktualisierende Liste aller aktuell laufenden Prozesse anzeigt und so eine Übersicht über die benötigten Ressourcen auf CPU und RAM bietet. Eine Alternative zum Unix-Klassiker „top“ ist das Programm „vmstat“, welches sich nicht mit der Auswertung der einzelnen Prozesse auseinandersetzt sondern einen generellen Überblick über die Auslastung der Ressourcen bietet und in diesem Paper ebenfalls ausführlicher behandelt werden soll.

Prozesse werden für gewöhnlich von einem Speichermedium geladen und erzeugte Daten auch wieder auf dieses geschrieben. Im PCs sind dies normale Festplatten auf Magnetscheiben oder Flash-Bausteine wie SSDs, in größeren Stagesystemen kommen aber auch Bandlaufwerke zum Einsatz. Wichtig bei Speichermedien ist die Anzahl der Schreib- und Lesezugriffe und die Schreib- und Leserate. Eine Festplatte mit einer hohen Schreibrate kann trotzdem schnell an ihre Grenzen stoßen, wenn sehr viele parallele Zugriffe (wie in einem Datenbanksystem üblich) auf das Medium stattfinden. Um solche Größen zu messen bietet sich das hier vorgestellte Programm „iostat“ an.

Interagieren mehrere Systeme miteinander, sei es ein Rechnernetz oder ein verteiltes System, so muss ein Netzwerk zwischen den einzelnen Knoten verwendet werden. Zugriff darauf hat ein Computer über ein Netzwerkinterface. Für den Nutzer eines Netzes ist meist die Datenrate die entscheidende Größe und somit ein lohnendes Ziel der Überwachung. Für den Betreiber eines Netzes spielt eher die Menge der übertragenen Daten eine Rolle. Beides kann mit Hilfe des hier exemplarisch vorgestellten Tools „*vnstat*“ durch live-Monitoring und erstellen von Statistiken ermöglicht werden. Mit Hilfe der Erweiterung „*vnstati*“ wird eine grafische Auswertung der Schnittstellenauslastung in Form von png-Grafikdateien ermöglicht.

Da auch der Energieverbrauch (besonders in großen Systemen) eine nicht unerhebliche Rolle spielt und Green-IT längst kein Nischendasein mehr fristet, ist auch die möglichst effiziente Auslastung der Hardware wichtig. Um etwaige stromfressende Prozesse während der idle-Time des Systems zu entlarven und festzustellen, wie effizient die vorhandenen Stromsparmechanismen genutzt werden, empfiehlt sich das Linux-Programm „*PowerTop*“.

Zusätzlich zu diesen ausführlich behandelten Programmen zur Systemanalyse werde ich im 8. Kapitel noch einmal kurz auf weitere interessante Programme eingehen, die zum Beispiel die Systemlaufzeit ausgeben oder zur Analyse von Multiprozessorauslastung dienen.

2. *vmstat* – System Activity, Hardware and System Information

2.1 Funktionen & Parameter

Vmstat ist ein universelles Analyse-tool zur Auswertung leistungsrelevanter Systemkomponenten und gibt tabellarisch Auskunft über Prozesse, Arbeitsspeicher, Auslagerung sowie Festplatten- und Prozessor-Aktivitäten. Es kann wahlweise einmalig Daten ausgeben oder in festlegbaren Abständen ein Update ausführen. Für die Ausgabe werden keine root-Rechte benötigt. Ruft man das Tool mit „*vmstat*“ auf, so erhält man folgende Ausgabe mit Mittelwerten aller Parameter seit dem Start des Systems:

```
root@s15399019:~# vmstat
procs -----memory----- ---swap-- -----io---- -system-- -----cpu-----
 r b swpd free buff cache si so bi bo in cs us sy id wa
 0 0 0 1935764 0 0 0 0 14 1 0 1 0 1 98 0
```

Die Werte haben dabei folgende Bedeutung:

- **procs**
 - **r**: Anzahl der auf Prozessor-Zeit wartenden Prozesse
 - **b**: Anzahl der nicht wiederbelebbaren schlafenden Prozesse
- **memory**
 - **swpd**: Größe des genutzten virtuellen Speichers
 - **free**: Größe des ungenutzten Speichers
 - **buff**: Größe des als Festplatten-Puffer genutzten Teils des Speichers
 - **cache**: Größe des als Cache genutzten Speichers
folgende Werte sind nur mit dem angehängten Parameter -a (s.u.) zu sehen:
 - **inact**: Größe des inaktiven Speichers
 - **active**: Größe des aktiv genutzten Speichers
- **Swap (in Daten pro Sekunde)**
 - **si**: (swap-in) Menge des Speichers, der von der Festplatte eingelesen wird
 - **so**: (swap-out) Menge des Speichers, der auf die Festplatte ausgelagert wird

- io
 - bi: (block-in) eingehende Blöcke pro Sekunde von Blockmedien (z.B. Festplatte)
 - bo: (block-out) ausgehende Blöcke pro Sekunde an Blockmedien (z.B. Festplatte)
- System
 - in: (interrupts) Anzahl der Interrupts pro Sekunde (inkl. der internen Uhr)
 - cs: (context switches) Anzahl der Kontextwechsel pro Sekunde
- CPU (alle Werte in Prozent der gesamten CPU-Zeit)
 - us: (user) Zeit, die für User-Prozesse verbraucht wird
 - sy: (system) Zeit, die für Kernelprozesse verbrauchte Zeit
 - id: (idle) Leerlaufzeit
 - wa: (wait) Wartezeit der CPU auf I/O-Operationen (z.B. Festplatte, Netzwerk)
 - st: (nur bei aktiven virtuellen Maschinen auf dem Hostsystem) CPU-Zeit für VMs

Das oben beispielhaft abgebildete System befindet sich momentan im Leerlauf, es finden weder I/O-Operationen noch CPU-Aufgaben statt.

Wichtige Parameter für vmstat:

- -a wechselt die memory-Ansicht auf active/inactive
- ZAHL x wird hinter vmstat eine Zahl geschrieben, aktualisiert sich das Programm und gibt alle x Sekunden einen neuen Momentanzustand des Systems aus
- -d zeigt eine ausführliche Festplattenstatistik an, in der bisher gelesene/geschriebene Datenmenge, und die dafür benötigte Zeit für jede gemountete Festplatte angezeigt wird
- -p PARTITION gibt ähnliche Werte wie -d aus, jedoch speziell für eine Partition
- -s Ausgabe der Arbeitsspeichertabelle (Vgl. /proc/meminfo)
- -S nützlicher Parameter für bessere Lesbarkeit der Werte (z.B. K → Kilobytes)

2.2 Möglichkeiten der Analyse

Aus den gezeigten Werten lassen sich bereits schnell Schlussfolgerungen auf den Momentanzustand des Systems schließen. Ein ideales System sollte gar keine wartenden Prozesse aufweisen (Parameter r). Die dort ausgegebene Zahl sollte nicht höher sein, als die Anzahl der CPU-Kerne. Ein Ausweg zur Optimierung wäre eine schnellere CPU oder Kontrolle der aktuell laufenden Prozesse mit Hilfe von „top“.

Die Swap-Partition eines Systems wird genutzt, wenn der Hauptspeicher des Systems überfüllt ist und Daten ausgelagert werden müssen. Ist dies der Fall, wird unter „si“ und „so“ ein erhöhter Datenfluss festzustellen sein. Eine Lösung für dieses Problem ist die Vergrößerung des Hauptspeichers.

Wenn ein sehr datenintensiver Prozess auf der CPU läuft, muss diese häufig auf das entsprechende Medium (z.B. Festplatte) warten. Ist dies der Fall, steigt die „wa“-Zeit der CPU an und das System läuft langsamer.

2.3 Lastbeispiele

- *hohe CPU-Last*

Zur Generierung einer hohen Last der CPU nutze ich folgenden Befehl, welcher mittels „dd“ eine 200 MB große Datei mit zufälligem Inhalt schreibt:

```
dd if=/dev/urandom of=100MBfile bs=1M count=100
```

/dev/urandom liefert hierbei Zufallszahlen, die von der CPU berechnet werden müssen und so als eine Last unter dem Parameter „sy“ zu bemerken sind (der verwendete PC hat 4 Kerne, sodass durch Auslastung nur eines Kernes 25% Last generiert werden).

```
stoerte@TheVoid:~$ vmstat 1
procs-----memory----- --swap-- ----io---- -system-- ----cpu----
 r b swpd free buff  cache  si so bi bo in cs us sy id wa
0 0  12 30456 186352 3394648  0 0  0  0 672 121 1  0 98 0 ← Vor der Ausführung
1 0  12 30932 186360 3384380  0 0  0  0 133  34 0 25 75 0 ← CPU-Last
1 0  12 39240 186360 3376524  0 0  0  0 157  50 0 25 75 0
1 0  12 37152 186360 3378332  0 0  0  0 150  49 0 25 74 0
0 0  12 35920 186368 3380524  0 0  0  0  39  32 0  0 100 0 ← Nach der Ausführung
```

- *hohe Schreiblast*

Um eine hohe Schreiblast zu demonstrieren wird wieder mit „dd“ aus dem Ordner /dev/zero gelesen und direkt in eine Datei geschrieben:

```
dd if=/dev/zero of=100MBfile bs=1M count=100 oflag=dsync
```

Durch „oflag=dsync“ wird sichergestellt, dass die generierten Daten sofort auf die Festplatte geschrieben werden und nicht gecached.

```
stoerte@TheVoid:~$ vmstat 1
procs-----memory----- --swap-- ----io---- -system-- ----cpu----
 r b swpd free buff  cache  si so bi bo in cs us sy id wa
0 0  12 142136 187232 3281860  0 0  0  0  52  40  0  0 100 0 ← Vorher
0 1  12 60832 187856 3362588  0 0  0 8332 767 581 0 2 74 24 ← Last
0 1  12 53452 187920 3369860  0 0  0 7328 754 580 0 1 76 22
0 1  12 45168 187984 3378152  0 0  0 8352 830 621 0 3 71 26
0 1  12 38952 188032 3384252  0 0  0 6264 647 471 0 2 76 23
0 0  12 39728 188032 3384252  0 0  0  2  78  66 0 0 97 3 ← Nachher
0 0  12 39696 188032 3384260  0 0  0  0  37  32 0 0 100 0
```

- *CPU wartet auf IO*

Um einen Prozess aufzurufen, der viele Zugriffe auf das Speichermedium ausführt, nutzen wir „updatedb“. Dieser ist ein Teil von „mlocate“ und erstellt für die Suchfunktion einen Index der Festplatte und muss sie dafür einmal komplett scannen.

```
root@s15399019:~# vmstat 1
procs-----memory----- --swap-- ----io---- -system-- ----cpu----
 r b swpd free buff cache  si so bi bo in cs us sy id wa
0 0  0 1927856  0  0  0  0  0  0 12290 0 0 99  0 ← Vorher
0 1  0 1927704  0  0  0  0 956  0 0 10192 0 1 75 23 ← Last
0 1  0 1927676  0  0  0  0 1784  0 0 11358 0 1 75 24
0 1  0 1927676  0  0  0  0 1960  0 0 9483  0 2 75 24
1 1  0 1927780  0  0  0  0 1084  0 0 10350 0 2 75 23
0 0  0 1927856  0  0  0  0  12  0 0 12950 0 0 98  1 ← Nachher
```

3. Top – Process Activity Command

3.1 Übersicht

Für die Prozessanalyse und das Auffinden von CPU-Verbrauchern ist seit langem das Programm Top erste Wahl, da es einen einfachen und dynamischen Überblick über alle laufenden Prozesse gibt. Seine Wurzeln hat es in dem Betriebssystem VMS, welches mit seinen `monitor process/topcpu`-Kommandos bereits ähnliche Funktionen bot. Durch die Entscheidung, das Programm Open Source zu entwickeln, verbreitete es sich seit 1984 in diverse Distributionen und gehört heute zur Grundausstattung vieler Linuxsysteme.

Vorteilhaft gegenüber ähnlichen Programmen wie „ps“, dass nur einmalig alle Prozesse ausgeht, wird „Top“ periodisch aktualisiert. Die Entwickler selbst sehen drei wesentliche Ziele für ihr Tool:

- eine genaue Momentaufnahme des System- und Prozess-Status
- möglichst geringer Hardwareverbrauch, sodass es beim Ausführen des Programms nicht selbst in den nach Verbrauch sortierten Prozessen ganz oben steht
- so handlich und portabel wie nur möglich sein

Neben Zugriff auf den `/proc`-Ordner (s.u.) benötigt Top ebenfalls Zugriff auf die Dateien `/dev/kmem` und `/dev/mem`, um einen direkten Zugriff auf den Arbeitsspeicherplatz des Kernels zu haben.

Der typische Aufbau nach Aufruf des Befehls „Top“ sieht wie folgt aus (hier: Webserver):

```
top - 17:37:26 up 44 days, 5:17, 2 users, load average: 0.00, 0.00, 0.00
Tasks: 27 total, 1 running, 26 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.2% us, 0.2% sy, 0.0% ni, 99.7% id, 0.0% wa, 0.0% hi, 0.0% si, 0.0% st
Mem: 2097152k total, 169156k used, 1927996k free, 0k buffers
Swap: 0k total, 0k used, 0k free, 0k cached

  PID USER      PR  NI  VIRT  RES  SHR  S %CPU  %MEM  TIME+  COMMAND
1567 server    18   0 257m  11m 5520 S   1    0.6  1049:55 ts3server_linux
  1  root      15   0 23324 1548 1248 S   0    0.1    0:01.91 init
1146 root      18   0 21088 1024  776 S   0    0.0    0:28.96 cron
1191 syslog    18   0 12460  760  588 S   0    0.0    0:24.59 syslogd
1288 root      15   0 70036 2412  740 S   0    0.1    3:58.73 sendmail-mta
1289 root      18   0 49272 2540 1988 S   0    0.1    0:29.38 sshd
1333 root      15   0 115m  7672 2204 S   0    0.4   32:26.35 fail2ban-server
1546 root      15   0  6624   624  500 S   0    0.0    1:26.79 vnstatd
5321 bind      18   0 143m  15m 2088 S   0    0.8    0:00.46 named
5380 root      15   0 71276 2980 1564 S   0    0.1    2:15.71 apache2
5457 www-data  15   0 71276 2472  988 S   0    0.1    2:17.76 apache2
5462 www-data  18   0 71276 2444  972 S   0    0.1    0:54.34 apache2
5479 www-data  15   0 71276 2420  956 S   0    0.1    0:02.37 apache2
5526 www-data  15   0 71432 2544 1044 S   0    0.1    0:01.93 apache2
6051 www-data  15   0 71276 2420  956 S   0    0.1    2:09.80 apache2
7925 www-data  15   0 71276 2420  956 S   0    0.1    0:02.39 apache2
7931 root      15   0 70632 3304 2524 S   0    0.2    0:00.33 sshd
```

Die Ansicht ist in zwei Teile aufgeteilt: im Kopf stehen zusammengefasste Informationen über den Gesamtzustand des Systems. Im unteren Hauptteil sind die eigentlichen Dienste (standardmäßig sortiert nach der höchsten CPU-Last) aufgelistet.

Kopf:

- erste Zeile: aktuelle Uhrzeit, Laufzeit, Anzahl eingeloggter Benutzer, Systemlast der letzten 1, 5 und 15 Minuten
- zweite Zeile: Anzahl aller Prozesse, Aktive, Inaktive, Gestoppte und beendete, aber noch in der Prozesstabelle befindliche Prozesse
- dritte Zeile: Aufschlüsselung der CPU-Auslastung (s.u.)
- vierte Zeile: gesamter installierter Speicher, gesamt verwendeter, freier, gepufferter Speicher
- fünfte Zeile: wie vierte Zeile, nur für die Swappartition

Erläuterung der Abkürzungen der CPU-Zeile:

```
us = Percentage of CPU time used by User-mode processes
sy = Percentage of CPU time used by Kernel-mode processes
ni = Percentage of CPU time used by processes with a 'Nice' value greater than
zero
id = Percentage of CPU time spent Idle, i.e. not used elsewhere
wa = Percentage of CPU time used waiting on Input/Output devices
hi = Percentage of CPU time used processing Hardware Interrupts (IRQs)
si = Percentage of CPU time used processing Software Interrupts
st = (SMP kernels only) Percentage of CPU time spent in involuntary wait
(Steal Time)
```

Hauptteil von Top:

- PID: Prozess-ID der aktuell laufenden Prozesse
- USER: User, der den Prozess ausführt
- PR: Priorität des Prozesses
- NI: Nice-Wert der Prozesse (je niedriger desto höher die Priorität)
- VIRT: SWAP+RES (Summe aus im Swap und im RAM aktiv belegten Speichers eines Prozesses)
- RES: Summe des genutzten Speichers (CODE+DATA)
- SHR: Shared Memory Size, Speicher der mit anderen Prozessen geteilt und gemeinsam genutzt wird
- S: Prozessstatus (R = running, S = sleeping, T = traced/stopped, Z = zombie)
- %CPU: prozentuale CPU-Auslastung des Prozesses seit der letzten Top-Aktualisierung (bei Mehrkernsystemen sind 100% nur ein Kern → 4 Kerne, 400%)
- %MEM: prozentuale Nutzung des Speichers
- TIME+: gesamte CPU-Zeit, seit der Prozess gestartet wurde (+ bedeutet auf Hundertstel genau)
- COMMAND: Name/Pfad des Prozesses (mit „c“ wechselbar)

3.2 Funktionen

Neben den Standardmäßig angezeigten Spalten kann man mit Hilfe der Taste „f“ weitere Spalten an- oder abwählen. Hier findet man ebenfalls eine kurze Erläuterung der Abkürzungen. Andere Hotkeys lassen sich direkt im Programm durch drücken von „h“ anzeigen.

Top bietet auch die Möglichkeit, direkt im laufenden Programm Veränderungen am Aussehen und Übersicht vorzunehmen. Eine Möglichkeit dazu sind das über „f“ erreichbare Menü zur Änderung der angezeigten Spalten. Eine andere Möglichkeit ist es, direkt in der Freizeile zwischen dem Kopf und dem Hauptteil die in „h“ aufgeführten Befehle einzugeben. So ist es zum Beispiel möglich, die Aktualisierungsgeschwindigkeit mit Hilfe von „s“ anzupassen, wobei hier auch Updates unter 1/s möglich sind.

Hat man mit Hilfe von Top einen Prozess gefunden, der sich festgefahren hat und dauerhaft eine zu hohe CPU-Auslastung anzeigt, so kann man ihn direkt im Programm „killen“ und muss dazu nicht erst zurück in die Shell. Die dafür benötigte PID lässt sich ebenfalls schnell ablesen und der Prozess durch die Eingabe „k PID“ beenden. Das Ergebnis kann man ebenfalls sofort in Top sehen.

Hat ein Prozess zu wenig CPU-Zeit für seine Ausführung, so kann es helfen, den nice-Wert und somit die Priorität des Prozesses zu ändern. Auch dies ist einfach mit Hilfe der PID und der Eingabe „r PID neuer-Nicewert“ direkt im laufenden Programm machbar.

Es ist ebenfalls möglich, Filter über die Ausgabe zu legen und so gezielter zu beobachten. Möchte man in der Prozesstabelle, in der alle auf dem System laufenden Prozesse angezeigt werden, nur einen bestimmten User überwachen, so kann man dies mit „u“ und dem Usernamen erreichen. Ab sofort werden in der Tabelle nur noch seine aktiv laufenden Prozesse angezeigt.

Auch für den Kopfteil gibt es weitere Einstellmöglichkeiten, so lässt sich zum Beispiel mit „l“ die CPU-Ansicht wechseln zwischen einer Durchschnittszahl über alle Prozessoren und einer aufgeschlüsselten Ansicht.

Diese Parameter sind nur beispielhaft, verdeutlichen aber die Möglichkeiten des Programms. Es ist mit Top sowohl möglich, eine sehr allgemeine Übersicht über den Gesamtzustand des Systems bezüglich der CPU-Auslastung zu erhalten, genauso wie eine sehr detaillierte spezifische Ansicht über einzelne Schwerpunkte.

4. PowerTOP – power usage analysis

4.1 Einsatzbereich

Energieeffizienz ist ein wichtiges Thema in der IT-Branche. Bei jeder neuen Prozessorgeneration wird neben mehr Rechenleistung auch darauf geachtet, dass besonders mehr Rechenleistung pro Watt erreicht wird. Um dies zu erreichen, bieten inzwischen alle CPU-Hersteller Stromsparmechanismen wie ACPI an. Wird die CPU nicht benötigt, geht sie in einen Energiesparmodus, in dem sie weniger Strom verbraucht. Jedoch machen einige Programme dieses Feature zu Nichte, indem sie periodisch Aufgaben an die CPU geben, sodass diese nicht in den sleep-modus gehen kann. Einer der häufigsten Gründe hierfür ist der Kernel selbst. Bis zur Kernelversion 2.6.21 gab es das sog. 1000 Hz Problem, welches die CPU regelmäßig aufweckte. Seit dem neuen „tickless idle“-Zustand kann so deutlich mehr Energie gespart werden durch länger durchgehende Ruhezustände. Ursprünglich entwickelt wurde das Programm für Notebooks mit Intel-CPUs.

Doch trotz des neuen Kernels gibt es noch genug Stromfresser, welche mit PowerTOP entlarvt werden können.

4.2 Programmübersicht

```
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
```

Der hier gezeigte Konsolenausschnitt zeigt PowerTOP in der Version 1.13. Der obere Teil zeigt an, wie viel Prozent der Zeit sich die CPU in den einzelnen C-states befindet. Sie geben Aufschluss über den Grad der Energieeinsparung. Je höher die C-Zahl, desto sparsamer ist die CPU und desto mehr Zeit benötigt sie auch, um wieder aktiv zu werden und Befehle entgegenzunehmen. C0 stellt einen besonderen Status da, in dem der Prozessor aktiv ist und Berechnungen ausführt. Die auf dem Screenshot zu sehende CPU besitzt nur einen C1-Status, in dem sie sich den Großteil der Zeit auch befindet (99,3%). Andere Systeme können einen höheres C-State erreichen.

Die grün hinterlegte Zeile gibt an, wie oft die CPU im beobachteten Zeitintervall aus dem idle aufgeweckt wurde. Steigt die Zahl der wakeups an, so wechselt die Farbe von Grün über Gelb auf Rot und symbolisiert so die Ineffizienz des Systems, sofern es sich im idle-Zustand befindet. Die Tabelle im unteren Teil listet die häufigsten Ursachen für wakeups auf. Die häufigste Ursache in diesem Fall ist der Prozess kworker, welcher eine Bündelung von mehreren einzelnen Kernelprozessen darstellt. Des Weiteren ist die Ethernet-Schnittstelle ein weiterer kaum vermeidbarer Grund für wakeups.

4.3 Beispiel

Das folgende Bild zeigt eine ineffiziente Situation, in der der Prozessor durch Ethernet-Interrupts und kworker geweckt wird und häufiger im C0-Status arbeiten muss.

```

PowerTOP version 1.13          (C) 2007 Intel Corporation
Cn      Avg residency      P-states (frequencies)
C0 (cpu running)      ( 7.6%)
polling      97.0ms ( 0.9%)
C1 mwait      17.6ms (91.5%)

Wakeups-from-idle per second : 52.2      interval: 3.0s
no ACPI power usage estimate available

Top causes for wakeups:
 0.5% ( 1.0)D getmail
 0.8% ( 1.7)D imap
22.1% ( 43.7) kworker/0:1
15.9% ( 31.3) [eth0] <interrupt>
13.0% ( 25.7) [uhci_hcd:usb3, ahci] <interrupt>
10.3% ( 20.3) [TLB shootdowns] <kernel IPI>

The program 'getmail' is writing to file 'opBuKC' on /dev/sda1.
This prevents the disk from going to powersave mode.

Q - Quit  R - Refresh

```

Meist sind die Probleme auf vielen Systemen die selben, sodass für Standardprobleme direkt in PowerTOP Lösungen vorgeschlagen werden. Dies geschieht im unteren Teil des Fensters. Im aktuellen Beispiel wird darauf hingewiesen, dass das Programm „getmail“ auf die Festplatte zugreift. Einige Vorschläge beziehen sich auf Änderung der Systemparameter, was direkt im Programm mit drücken der dafür vorgegebenen Taste möglich ist. Möchte man diese Änderungen auch permanent behalten (alle mit PowerTOP vorgenommenen Einstellungen sind nach einem Neustart zurückgesetzt), muss man die Befehle aufschreiben und in das Startscript von Linux einfügen.

5. vnstat – network analysis and statistics

5.1 Einsatzzweck

Vnstat ist ein konsolenbasierter Netzwerktrafficmonitor für Linux, der sowohl die Anzeige der Echtzeitauslastung bestimmter Interfaces als auch die Speicherung und Auswertung der Daten beherrscht. Durch das reine abtasten der Datenrate, ohne direkt in die Pakete zu schauen und sie nach Art und Größe zu klassifizieren (sniffing), hat vnstat eine nur sehr geringe Systemauslastung. Um funktionsfähig zu sein, muss das System, auf dem es laufen soll, mindestens den Linuxkernel 2.2 besitzen.

Im Gegensatz zu anderen Programmen wie „Top“, welches eine Interaktion mit den ausgegebenen Daten ermöglicht, ist vnstat ein reines Analysetool ohne direkte Interaktion wie Traffic-limits.

5.2 Programmübersicht

Installiert man vnstat neu auf einem System, kann es anfangs nicht viele Daten ausgeben, da es diese erst sammeln muss. Zu diesem Zweck startet man es mit folgenden Parametern

„Vnstat -u -i eth0“ um das Programm anzuweisen, die Schnittstelle eth0 zu protokollieren. Es ist auch möglich, mehrere Schnittstellen parallel zu überwachen.

Hat das Programm ein paar Stunden oder Tage lang Daten über das Interface gesammelt, so erhält man beim Aufruf folgende Ansicht:

```
root@s15399019:~# vnstat
Database updated: Wed Mar 21 18:06:15 2012

venet0 since 03/05/12

    rx: 6.94 GiB   tx: 316.90 GiB   total: 323.84 GiB

monthly
-----+-----+-----+-----
      rx |      tx |      total | avg. rate
-----+-----+-----+-----
Mar '12  6.94 GiB | 316.90 GiB | 323.84 GiB | 1.51 Mbit/s
-----+-----+-----+-----
estimated 10.36 GiB | 473.35 GiB | 483.71 GiB |

daily
-----+-----+-----+-----
      rx |      tx |      total | avg. rate
-----+-----+-----+-----
yesterday 453.43 MiB | 20.22 GiB | 20.67 GiB | 2.01 Mbit/s
today     146.05 MiB | 10.28 GiB | 10.42 GiB | 1.34 Mbit/s
-----+-----+-----+-----
estimated  193 MiB | 13.63 GiB | 13.82 GiB |
```

Es wird eine Übersicht über den Traffic seit dem Start des Programms ausgegeben, was in diesem Beispiel der 5. März 2012 am Interface venet0 ist.

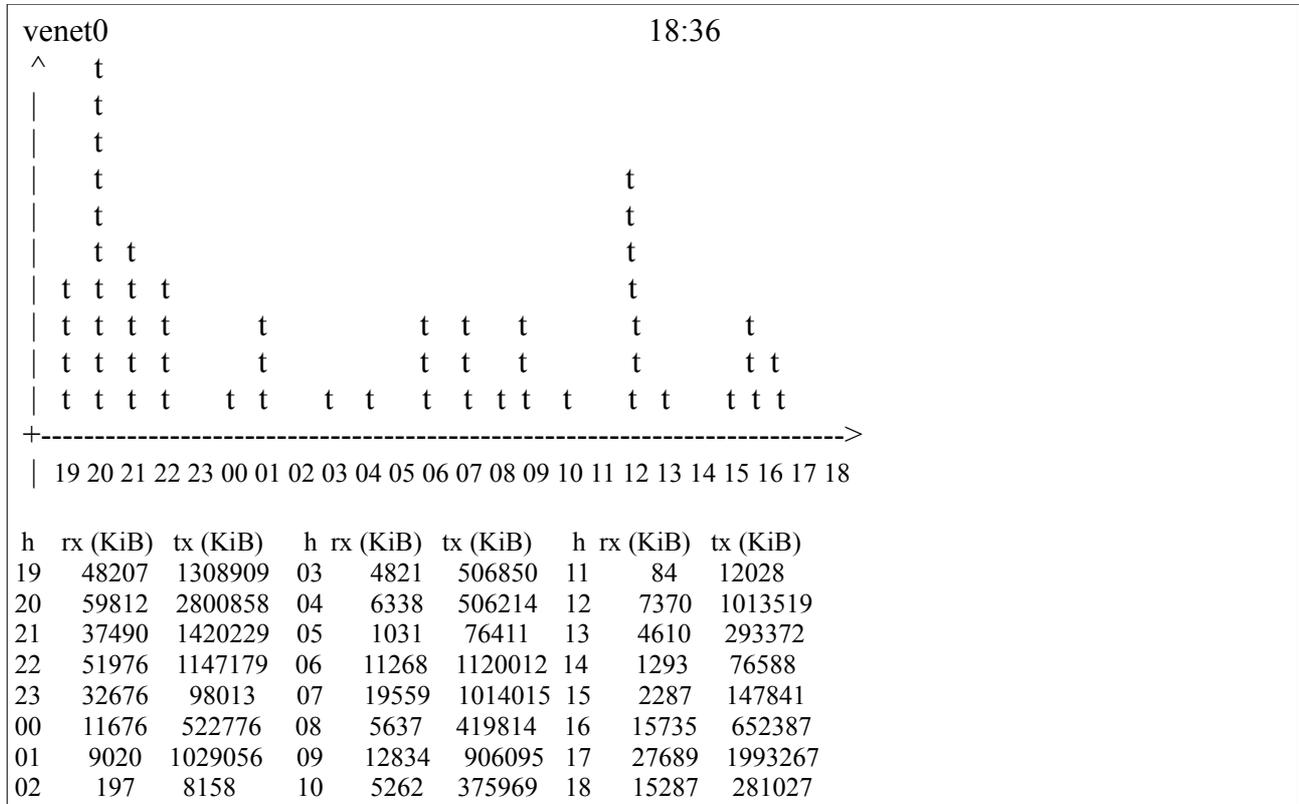
Darunter findet sich die Übersicht über den gesamten Datenverkehr, wobei hier rx für received und tx für transmitted steht.

Darunter wird eine monatliche Auswertung der Daten mit empfangenen, gesendeten und gesamten Daten angegeben, sowie die daraus resultierende durchschnittliche Datenrate. Unter „estimated“ sind die jeweils erwarteten Werte für den Monat, basierend auf Hochrechnungen vorhandener Daten angegeben.

Unter den monatlichen Statistiken finden wir die täglichen Statistiken für den aktuellen Tag und den Vortag. Das erwartete Datenvolumen bezieht sich auf den laufenden Tag.

Dies ist jedoch nur eine zusammengefasste Übersicht. Meist interessieren eher einzelne Tage, Monate oder eine Echtzeitanzeige.

Der Befehl „vnstat -h“ gib ein simples Diagramm der letzten 24 Stunden mit Tabelle für jede einzelne Stunde aus:



So können sowohl grob im Diagramm als auch sehr speziell in der darunter befindlichen Tabelle Aussagen zu Spitzenlastzeiten gemacht werden.

Der nächst größere Zeitraum sind Tage, welche mit „vnstat -d“ ausgegeben werden:

```

venet0 / daily

```

day	rx	tx	total	avg. rate
03/05/12	393.65 MiB	10.37 GiB	10.76 GiB	1.04 Mbit/s
03/06/12	357.39 MiB	14.44 GiB	14.79 GiB	1.44 Mbit/s
03/07/12	393.81 MiB	13.15 GiB	13.53 GiB	1.31 Mbit/s
03/08/12	410.09 MiB	19.22 GiB	19.62 GiB	1.90 Mbit/s
03/09/12	318.02 MiB	16.23 GiB	16.54 GiB	1.61 Mbit/s
03/10/12	307.18 MiB	21.88 GiB	22.18 GiB	2.15 Mbit/s
03/11/12	470.17 MiB	29.26 GiB	29.72 GiB	2.89 Mbit/s
03/12/12	437.50 MiB	15.55 GiB	15.98 GiB	1.55 Mbit/s
03/13/12	410.25 MiB	13.38 GiB	13.78 GiB	1.34 Mbit/s
03/14/12	516.45 MiB	14.54 GiB	15.04 GiB	1.46 Mbit/s
03/15/12	353.80 MiB	14.79 GiB	15.13 GiB	1.47 Mbit/s
03/16/12	421.39 MiB	19.71 GiB	20.12 GiB	1.95 Mbit/s
03/17/12	402.13 MiB	24.67 GiB	25.06 GiB	2.43 Mbit/s
03/18/12	803.94 MiB	36.07 GiB	36.85 GiB	3.58 Mbit/s
03/19/12	509.22 MiB	23.15 GiB	23.65 GiB	2.30 Mbit/s
estimated	204 MiB	13.42 GiB	13.62 GiB	

Hier erfolgt ähnlich der Standardausgabe eine tägliche Unterteilung in eingehenden, ausgehenden, gesamten Datenverkehr und die durchschnittliche Datenrate des entsprechenden Tages.

Analog dazu gibt es noch eine wöchentliche und monatliche Ausgabe, welche den selben Aufbau besitzen.

Interessant ist noch die Ausgabe von „vnstat -t“, welche die Top10 der trafficintensivsten Tage auflistet:

```
venet0 / top 10
```

#	day	rx	tx	total	avg. rate
1	03/18/12	803.94 MiB	36.07 GiB	36.85 GiB	3.58 Mbit/s
2	03/11/12	470.17 MiB	29.26 GiB	29.72 GiB	2.89 Mbit/s
3	03/17/12	402.13 MiB	24.67 GiB	25.06 GiB	2.43 Mbit/s
4	03/19/12	509.22 MiB	23.15 GiB	23.65 GiB	2.30 Mbit/s
5	03/10/12	307.18 MiB	21.88 GiB	22.18 GiB	2.15 Mbit/s
6	03/20/12	453.43 MiB	20.22 GiB	20.67 GiB	2.01 Mbit/s
7	03/16/12	421.39 MiB	19.71 GiB	20.12 GiB	1.95 Mbit/s
8	03/08/12	410.09 MiB	19.22 GiB	19.62 GiB	1.90 Mbit/s
9	03/09/12	318.02 MiB	16.23 GiB	16.54 GiB	1.61 Mbit/s
10	03/12/12	437.50 MiB	15.55 GiB	15.98 GiB	1.55 Mbit/s

Manchmal benötigt man jedoch auch die aktuelle Bandbreite des Interfaces in Echtzeit. Dies geschieht mit dem Parameter „vnstat -l“

```
Monitoring venet0... (press CTRL-C to stop)
```

```
rx: 32 kbit/s 37 p/s tx: 32 kbit/s 37 p/s
```

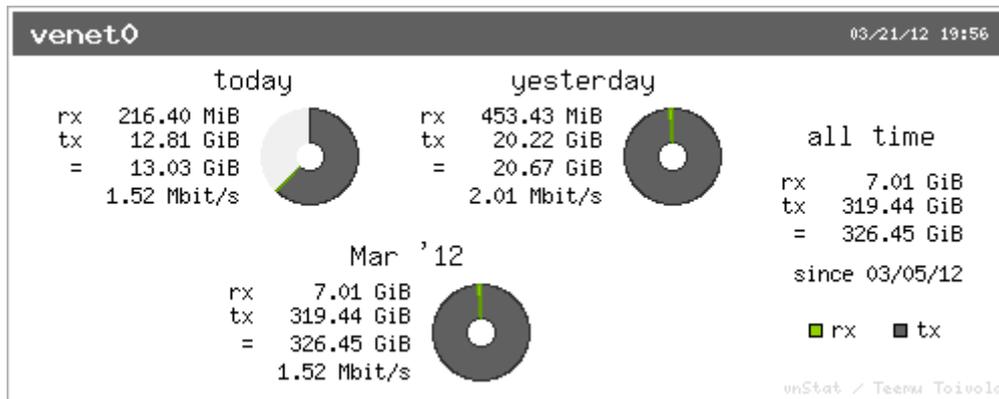
```
venet0 / traffic statistics
```

	rx	tx
bytes	539 KiB	847 KiB
max	180 kbit/s	296 kbit/s
average	70.69 kbit/s	111.08 kbit/s
min	4 kbit/s	4 kbit/s
packets	4231	6249
max	161 p/s	257 p/s
average	69 p/s	102 p/s
min	14 p/s	14 p/s
time	1.02 minutes	

Wird der Befehl aufgerufen, so sieht man nur die obere Zeile und es wird in Echtzeit sowohl der eingehende als auch der ausgehende Datenverkehr im kb/s und p/s angegeben. Die angezeigte Datengröße kb/s ändert sich dynamisch zu Mbit/s, sollte der Verkehr zu stark ansteigen, um eine bessere Übersicht zu gewähren.

5.3 Vnstat

Das Programm „vnstati“ ist eine Erweiterung von vnstat, wobei i für Image steht. Die gesammelten Daten sind für gewöhnlich als logs auf dem System gespeichert und können nur mit Hilfe von vnstat in der Konsole ausgegeben werden, sofern man einen Account und Zugang zum Server besitzt. Durch das Erzeugen von Grafiken ist es möglich, die erhobenen Daten auch in einer Grafikdatei auszugeben. Hierfür steht aktuell nur das Format .png zur Verfügung. Die so entstandenen Dateien können zum Beispiel auf einem Webserver gespeichert und so jedem Besucher zugänglich gemacht werden.



Analog zu der tabellenbasierten Ausgabe (s.o.) gibt es die gleichen Werte auch als Bild. Die Erzeugung der Grafiken erfolgt genau wie mit vnstat. Es gibt die selben Parameter für Tage, Wochen etc. wobei zusätzlich noch der Speicherpfad und der Name der Ausgabedatei angegeben werden muss.

Zur bequemen Aktualisierung kann man Linux bequem mit einem cron-job anweisen, regelmäßig den gewünschten Befehl auszuführen, um so immer aktuelle Werte in eine Grafik zu rendern.

6. Iostat

6.1 Einsatzbereich

Die Speicherhierarchie eines Computers kann man pyramidenförmig darstellen. An der Spitze stehen sehr kleine und schnelle Speicher, wie die Register und der Cache der CPU, darunter kommen die etwas größeren aber langsameren Speichermedien wie der Arbeitsspeicher. Auf der darunter befindlichen Ebene sind die Massenspeicher zu finden, typischerweise Festplatten. Da diese die ersten Medien in der Hierarchie sind, die auch nach abschalten des Systems ihre Daten noch behalten und sehr voluminös sind, erfolgen hier beim starten von Programmen oder laden von Daten häufig Wartezeiten. Diese können sich noch potenzieren, greifen mehrere Programme parallel auf das Medium zu. Da dies gerade bei Systemen mit komplexen Datenbanken der Fall ist, bei denen sekundlich viele Anfragen parallel gestellt werden, ist hier der Wunsch nach höchstmöglicher Performance und Minimierung von störenden Zugriffen besonders hoch.

An dieser Stelle setzt das Programm „iostat“ an. Io (oder auch I/O) steht für Input/Output und beschreibt den eingehenden bzw. ausgehenden Datenverkehr an Speichermedien, in diesem Fall der Festplatten. Iostat überwacht also als Live-monitoring-tool die Betriebssystemspeichermedien und gibt auch Auskunft über die Auslastung der CPU und ihrer entsprechenden Wartezeit wegen Festplattenzugriffen.

6.2 Programmübersicht

Wird der Befehl ohne Parameter ausgeführt, so gibt er (ähnlich wie vmstat) den Durchschnittswert der Auslastung seit dem Systemstart an (hier ein frisch gestartetes System):

```
stoertebeker@ubuntu:~$ iostat
Linux 3.0.0-12-generic (ubuntu)      22.03.2012   _x86_64_   (4 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           1,92    0,00    2,73   42,46    0,00   52,89

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sda                 125,77      6328,02       41,81      290013     1916
scd0                 0,26         1,05         0,00         48         0
sdb                  3,67         15,27         0,00         700         0
```

Insgesamt generiert iostat für drei verschiedene Komponenten Meldungen: CPU-Auslastung, Medienauslastung und Netzwerkdateisystemauslastung. Im gezeigten Beispiel sind jedoch nur die ersten Beiden zu sehen.

Die CPU-Auslastung ähnelt stark der von top:

- %user: CPU-Zeit für User-Programme
- %nice: CPU-Zeit für Prozesse mit hoher nice-Priorität (→ niedriger nice-zahl)
- %system: CPU-Zeit für Kernelprozesse
- %iowait: Zeit, die die CPU auf I/O-Operationen warten musste
- %steal: CPU-Zeit, die für anderen virtuellen Prozessor „gestohlen“ wurde
- %idle: Leerlaufzeit der CPU

Darunter befindet sich der Teil über die Bock-Medien (Festplatten):

- Device: alle in /dev gelisteten Geräte/Partitionen
- tps: „transfers per second“, eine I/O-Anfrage ist ein Transfer, mehrere logische Anfragen können zu einer Anfrage an ein Gerät kombiniert werden
- kB_read/s: Lese-Datenrate in Kilobyte pro Sekunde
- kB_wrtn/s: Schreib-Datenrate in Kilobyte pro Sekunde
- kB_read: gesamte Menge an gelesenen Daten
- kB_wrtn: gesamte Menge an geschriebenen Daten

Wird iostat mit dem Parameter -x (extended) aufgerufen, so gibt es noch folgende zusätzliche Werte über Block-Medien:

- rrqm/s: „read requests queued merge“ gesammelte Leseanfragen an das Device
- wrqm/s: „write requests queued merge“ gesammelte Schreibanfragen an das Device
- r/s: Anzahl der Leseanforderungen, die an das Gerät pro Sekunde ausgegeben werden
- w/s: Anzahl der Schreib-Zugriffe, die auf das Gerät pro Sekunde ausgegeben werden
- avgrq-sz: durchschnittliche Länge (in Sektoren) der Anfragen, die an dem Gerät ausgegeben werden
- avgqu-sz: durchschnittliche Länge der Warteschlange der Anträge, die an das Gerät ausgegeben werden
- await (r/w): durchschnittliche Zeit in ms für die Bearbeitung von I/O-Anfragen
- svctm: „service time“ durchschnittliche Zeit in ms für I/O-Anfragen, bis sie an das Gerät ausgegeben werden
- %util: CPU-Zeit, während I/O-Zugriffe an das Gerät ausgegeben werden

```

stoertebecker@ubuntu:~$ iostat -m -x
Linux 3.0.0-12-generic (ubuntu)      22.03.2012   _x86_64_   (4 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0,11    0,00    0,12    0,61    0,00   99,17

Device:            rrqm/s  wrqm/s   r/s    w/s  rMB/s  wMB/s avgrq-sz  avgqu-sz   await  r_await w_await svctm  %util
sda                 61,21   0,58    3,71   0,35  0,23   0,00   117,79    0,02     5,75   4,65   17,57   3,32  1,35
scd0                 0,01    0,00    0,01   0,00  0,00   0,00     8,00    0,00    30,33  30,33    0,00  30,33  0,03

```

6.3 Beispiele

Mit Hilfe dieser doch sehr detaillierten Ansicht der einzelnen Werte kann man nicht nur die Momentanauslastung erkennen, sondern mit Hilfe von künstlich erzeugter Last die Maximalwerte ermitteln oder mit anderen Festplatten vergleichen bezüglich ihrer Leistungsfähigkeit für I/O-Operationen oder der Datenrate.

Im praktischen Fall muss hier noch unterschieden werden zwischen herkömmlichen Festplatten mit Magnetspeicherscheiben und modernen Solid State Disks (SSD), welche durch Flashbausteine eine um den Faktor 3 höhere I/O-Rate haben als herkömmliche Festplatten (einige hundert gegen einige hunderttausend I/O pro Sekunde).

Die folgende Ausgabe von iostat mit den Parametern -m -x (Angaben in Megabyte, ausführliche Ausgabe) wurde erstellt, während auf dem System einer hohen Schreibrate belastet wurde (Vgl. 2.3)

```

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           5,82    0,00    1,05   52,91    0,00   40,22

Device:            rrqm/s  wrqm/s   r/s    w/s  rMB/s  wMB/s avgrq-sz  avgqu-sz   await  r_await w_await svctm  %util
sda                 31,40  5374,80  5,60  122,40  0,14   32,65   524,71   118,92  1003,39  40,14  1047,46  7,49  95,84
sdb                  0,00    0,00    0,00   0,00  0,00   0,00     0,00    0,00    0,00   0,00    0,00   0,00  0,00

```

Die Zeile über die CPU-Auslastung zeigt, dass mit einem %iowait-Wert von fast 53% die CPU die Hälfte der Zeit auf das Schreiben der von ihr erzeugten Daten warten muss.

In der Device-Übersicht sieht man sehr deutlich, dass nur das Gerät sda aktiv war und sdb im idle blieb. Ebenfalls sehr deutlich zu sehen ist, dass eine hohe Schreibrate (wMB/s) von 32,65 MB/s erzeugt wird und die hohe Wartezeit (await), besonders beim schreiben (w_await) durch den konstant erzeugten Datenstrom, vorhanden ist. Bei %util ist der Wert mit 95% ebenfalls sehr hoch, sodass das Gerät an seiner Sättigungsgrenze arbeitet und keine freien Kapazitäten für andere Aufgaben mehr hat.

7. Gnome System Monitor – Real-time Systems Reporting and Graphing

7.1 Einsatzbereich

Längst vorbei sind die Zeiten, als Computer lediglich der Wissenschaft dienen. Ein PC gehört heutzutage ebenso wie ein Telefon zu einem normalen Haushalt, entsprechend entwickelte sich auch die Benutzerfreundlichkeit der Interaktion. Reichte es früher noch aus, über ein Terminal Daten einzugeben und Ergebnisse auf selbem Wege zu erhalten, ist heutzutage eine GUI das Mindeste im Endanwenderbereich. So verwundert es auch nicht, dass neben den ganzen konsolenbasierten Programmen zu Leistungsanalyse auch einige populäre Versionen mit umfassender grafischer Oberfläche entstanden sind. Ähnlich dem Taskmanager in Windows-Betriebssystemen gibt es unter Linux den Gnome System Monitor.

Dieser ist ebenfalls eine echtzeitfähige Anwendung für die Desktopoberfläche Gnome und bietet mit Hilfe von Diagrammen eine schnelle Übersicht über wesentliche Bestandteile des Systems. Eine Interaktion mit den Prozessen ist ähnlich wie mit Top direkt im Programm möglich.

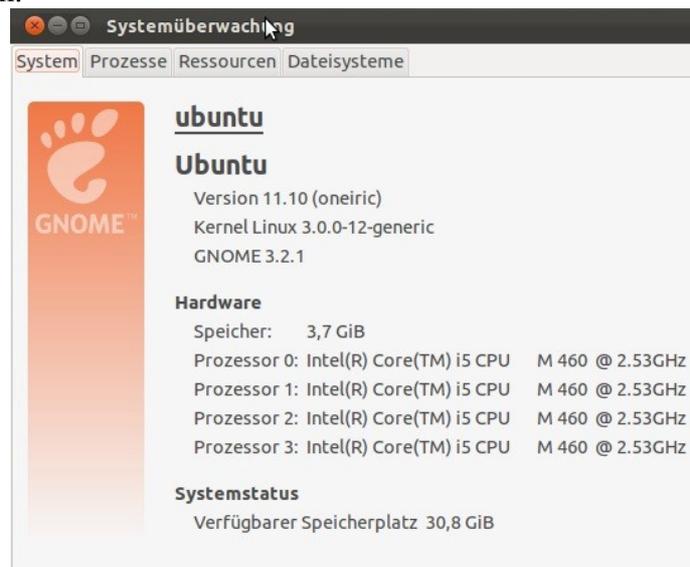
7.2 Übersicht

Die Systemüberwachung wird in 4 Bereiche unterteilt: System, Prozesse; Ressourcen, Dateisysteme.

In der Systemübersicht bekommt man allgemeine Information über Das Betriebssystem und die Hardware. Neben der eigentlichen Version von Ubuntu ist hier auch die Kernelversion und die der Oberfläche GNOME zu sehen.

Die Hardware beschränkt sich auf Größe des installierten Arbeitsspeichers und Anzahl und Modell der einzelnen Kerne. Ob es sich um einen echten 4-Kerner handelt oder nur um einen (wie hier der Fall) 2-Kerner mit 2 virtualisierten Kernen, ist nicht direkt sichtbar.

Der Systemstatus gibt Auskunft über das auf dem Hostmedium noch frei verfügbare Speichervolumen.



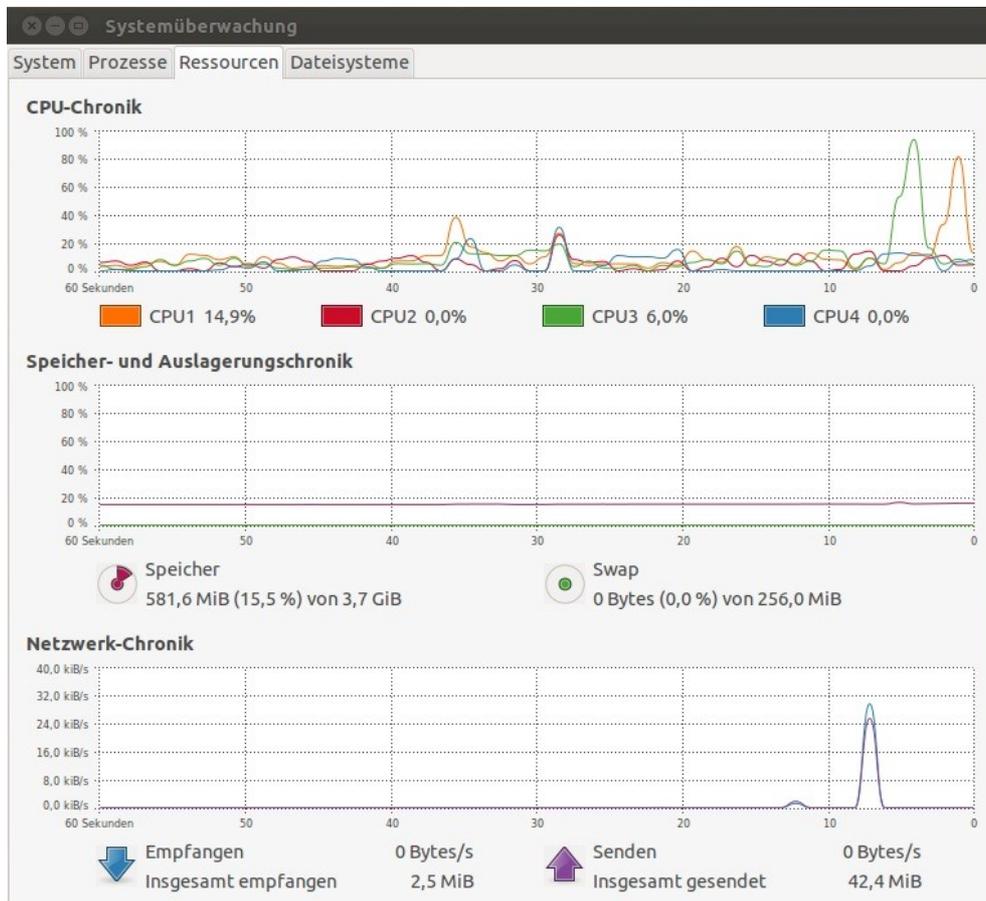
Die zweite Karteikarte Prozesse verhält sich ähnlich wie die Prozessausgabe des Programms „Top“ und gibt eine Auflistung aller gestarteten Prozesse aus. Durch die GUI kann hier bequemer die Sortierung gewählt werden (hier: nach CPU-Auslastung). Über der Prozesstabelle wird wie schon aus anderen Programmen bekannt die Auslastung des Systems der letzten 1, 5 und 15 Minuten angezeigt.

The screenshot shows the 'Prozesse' tab of the System Monitoring application. It displays the following information:

Durchschnittslast der letzten 1, 5 und 15 Minuten: 0,25 - 0,11 - 0,07

Prozessname	Status	% CPU	Nice-Wert	Kennung	Speicher	Waiting Channel
applet.py	Schläft	0	0	2526	12,8 MiB	poll_schedule_time
bamfd daemon	Schläft	0	0	2449	3,4 MiB	poll_schedule_time
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_time
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

Durch einen Rechtsklick auf die jeweiligen Prozesse gibt es die Möglichkeit, die Priorität der Prozesse zu ändern, seine offenen Fenster anzeigen zu lassen oder sie zu „killen“. Überfährt man einen Prozess nur mit der Maus, wird sein Speicherpfad angezeigt, in dem er ausgeführt wird. Die dritte Karteikarte ermöglicht mit Hilfe von fortlaufenden Diagrammen eine Echtzeitsicht auf die CPU-Auslastung (inkl. Unterteilung auf die einzelnen Kerne), des Speichers und der Auslagerungsdatei (Swap) sowie dem Netzwerk für senden und empfangen. Für das Netzwerk werden neben Momentanauslastung auch Statistiken für den Gesamtverkehr seit dem Start der Netzwerkverbindung angezeigt.



Ein Nachteil dieser einfachen und unkomplizierten Übersicht ist das Problem, dass sich in der Netzwerkansicht nicht unterscheiden lässt, über welches Interface Daten aus- oder eingehen. Ist der PC über zwei Schnittstellen mit einem Netzwerk verbunden, so wird trotzdem nur eine Gesamtansicht gegeben.

Die letzte Kartei Dateisysteme beinhaltet eine triviale Übersicht über die angeschlossenen Medien (Festplatten, optische Laufwerke, Speichersticks etc.). Neben Informationen über Gesamtgröße und freien Speicher auf den Medien findet man unter „Gerät“ auch ihren Platz im System und unter „Ordner“ den Pfad, in dem sie eingehängt sind. Die Spalte „Typ“ gibt Auskunft über das Dateiformat der Medien.

The screenshot shows the 'Systemüberwachung' window with the 'Dateisysteme' tab selected. It displays a table of mounted filesystems:

Gerät	Ordner	Typ	Gesamt	Frei	Verfügbar	Belegt		
/dev/loop0	/	ext3	11,1 GiB	7,5 GiB	7,5 GiB	3,6 GiB		32 %
/dev/sda2	/host	fuseblk	116,4 GiB	23,3 GiB	23,3 GiB	93,1 GiB		79 %

Diese Übersicht ist ebenfalls nur eine sehr einfache Sicht auf das Gesamtsystem der Speichermedien. Die einzige Interaktion ist ein Doppelklick auf die Medien, mit denen sie im Dateibrowser geöffnet werden.

8. andere Tools

8.1 w - Find Out Who Is Logged on And What They Are Doing

„w“ ist ein Programm, das nichts anderes tut, als nach Aufruf eine Übersicht über alle aktuell angemeldeten Benutzer zu geben und was sie gerade tun:

```
stoertebeker@ubuntu:~$ w
16:34:10 up 3:52, 2 users, load average: 0,01, 0,06, 0,06
USER  TTY  FROM          LOGIN@  IDLE   JCPU   PCPU   WHAT
stoerteb pts/0 192.168.2.148 12:41   0.00s  0.28s  0.00s  paper -s
stoerteb pts/1 192.168.2.148 12:42   3:39m  0.29s  0.29s  -bash
stoerteb pts/2  :0                12:41   2:00m  0,14s  0,00s  w
```

Die Kopfzeile gibt wieder eine Gesamtübersicht der Systemlaufzeit und -last an.

Die Spalten der Benutzerübersicht haben folgende Bedeutung:

- USER: Benutzername (ggf. abgekürzt bei zu langen Namen)
- tty: „teletypewriter“ tty-Name
- FROM: der entfernte Host (bei lokalem login → :0)
- LOGIN@: Login-Zeit des Users
- IDLE: Leerlaufzeit seit letzter Interaktion
- JCPU: Zeit, die von allen Prozessen genutzt wurde, die am jeweiligen Terminal angemeldet sind
- PCPU: Zeit, die vom laufenden Prozess bisher genutzt wurde
- WHAT: Kommandozeile des aktuell laufenden Prozesses

Eine ähnliche Übersicht bietet das fast gleichnamige Tool „who“

8.2 uptime - Tell How Long The System Has Been Running

Manchmal genügt die einfache Ausgabe der Laufzeit des Systems. Für diese Fälle gibt es Das Tool „uptime“ was ähnlich einer aus anderen Programmen wie „top“ oder „w“ bekannten Infozeile ist.

```
stoertebeker@kastenbier:~$ uptime
16:56:23 up 68 days, 33 min, 1 user, load average: 4.02, 4.07, 4.12
```

Hier werden ebenso noch ein paar zusätzliche Infos, wie eingeloggter User oder durchschnittliche Systemauslastung angezeigt.

8.3 free – Memory Usage

Der Befehl „free“ zeigt eine einfache und schnelle Übersicht über die Auslastung des Hauptspeichers und der dazugehörigen swap-Datei an. Zur besseren Lesbarkeit können mit den entsprechenden Parametern (-k → Kilobyte etc.) die Ausgabewerte angepasst werden:

```
stoertebeker@kastenbier:~$ free -m
              total    used    free   shared  buffers   cached
Mem:          3708    3237    470      0     216    2399
-/+ buffers/cache:    621    3086
Swap:        10239     16    10223
```

Das Programm unterteilt dabei in drei Bereiche: den installierten Arbeitsspeicher (Mem), den vom Kernel genutzten Puffer (buffers/cache) und die Auslagerungsdatei (swap). In jedem werden die entsprechend möglichen Werte für die Spalten angezeigt. Das der Arbeitsspeicher dabei zum Beispiel der einzige Puffer ist, versteht sich von selbst.

8.4 mpstat – Multiprocessor Usage

Alle bisher vorgestellten Programme beschränken sich bei der Auslastung des Prozessors lediglich auf einen Gesamtwert bzw. Durchschnittswert der Auslastung. Mpstat gibt eine ausführlichere Ansicht aller im System aktiven Prozessoren und deren Auslastung. Als Standardbefehl „mpstat“ wird lediglich die Zeile „all“ im folgenden Beispiel ausgegeben und ist somit auch nur eine Zusammenfassung. Durch den Parameter „-P“ für Processor kann man sich die Auslastung der einzelnen Kerne anzeigen lassen (beginnend mit 0) oder mit „ALL“ eine Tabelle für alle Kerne. Bei einmaligem Ausführen des Befehl ohne Intervallparameter wird ähnlich „vmstat“ die gemittelte Auslastung seit Systemstart ausgegeben:

```
stoertebeker@kastenbier:~$ mpstat -P ALL
Linux 3.0.0-14-server (kastenbier)  24.03.2012  _x86_64_  (4 CPU)
```

12:43:30	CPU	%usr	%nice	%sys	%iowait	%irq	%soft	%steal	%guest	%idle
12:43:30	all	3,20	88,84	7,16	0,11	0,00	0,02	0,00	0,00	0,68
12:43:30	0	4,77	88,51	5,78	0,37	0,00	0,07	0,00	0,00	0,50
12:43:30	1	2,72	88,92	7,61	0,04	0,00	0,00	0,00	0,00	0,72
12:43:30	2	2,65	88,98	7,60	0,01	0,00	0,00	0,00	0,00	0,74
12:43:30	3	2,67	88,93	7,65	0,01	0,00	0,00	0,00	0,00	0,74

Auf dem beispielhaft gezeigten System gibt es durch Folding@home eine sehr hohe Last auf allen Kernen an. Die Spaltenbezeichnungen sind ähnlich den anderen vorgestellten Programmen und werden hier nicht weiter näher erklärt.

Auch die Anzahl der Interrupts pro Prozessor lässt sich mit mpstat anzeigen:

```
stoertebeker@kastenbier:~$ mpstat -I SUM -P ALL
Linux 3.0.0-14-server (kastenbier)  24.03.2012  _x86_64_  (4 CPU)
```

12:50:19	CPU	intr/s
12:50:19	all	1791,35
12:50:19	0	69,47
12:50:19	1	0,00
12:50:19	2	0,01
12:50:19	3	15,65

9. Das /proc-Dateisystem

Die oben bereits ausführlich behandelten Programme haben alle eines gemeinsam: sie greifen auf das /proc-Dateisystem zu, um ihre benötigten Daten zu ermitteln.

In jedem Linux-system herrscht eine ähnliche Dateistruktur vor: als oberste Wurzel im Dateisystem gilt das „/“. Unter ihm sind meist Ordner wie /dev, /home, /root oder /var vorhanden. Doch nicht alle davon sind Ordner im eigentlichen Sinne, wie der Ordner /proc beweist. Er ist eine Art virtuelles Dateisystem, in dem aus Pseudodateien Statusinformationen des Kernels generiert und ausgelesen werden können. Wird eine dieser Dateien geöffnet, so gibt der Kernel die gewünschten Daten dynamisch zum Anforderungszeitpunkt aus, ein Schreibbefehl auf die Festplatte findet dabei nicht statt.

Ein typischer /proc-Ordner kann wie folgt aussehen:

```
root@s15399019:/proc# ls
1      1567 6051 8018    cpuinfo  meminfo  sys          vz
1146   5321 7925 8113    devices  modules  sysrq-trigger
1191   5380 7931 8124    filesystems  mounts  sysvipc
1288   5457 7942 8130    fs        net       uptime
1289   5462 8015 8141    kmsg     self      user_beancounters
1333   5479 8016 9393    loadavg  stat      version
1546   5526 8017 cmdline locks     swaps     vmstat
```

Die Zahlen-Ordner sind einzelne Prozesse, die restlichen erklären sich meist von selbst. Die Datei „meminfo“ zum Beispiel gibt Informationen über die aktuelle Speicherauslastung an:

```
root@s15399019:/proc# cat meminfo
MemTotal: 2097152 kB
MemFree: 1927488 kB
Buffers: 0 kB
Cached: 0 kB
SwapCached: 0 kB
Active: 0 kB
Inactive: 0 kB
HighTotal: 0 kB
HighFree: 0 kB
LowTotal: 2097152 kB
LowFree: 1927488 kB
SwapTotal: 0 kB
SwapFree: 0 kB
Dirty: 0 kB
Writeback: 0 kB
AnonPages: 0 kB
Mapped: 0 kB
Slab: 0 kB
PageTables: 0 kB
NFS_Unstable: 0 kB
Bounce: 0 kB
CommitLimit: 0 kB
Committed_AS: 0 kB
VmallocTotal: 0 kB
VmallocUsed: 0 kB
VmallocChunk: 0 kB
HugePages_Total: 0
HugePages_Free: 0
HugePages_Rsvd: 0
Hugepagesize: 2048 kB
```

Programme wie „vmstat“ greifen auf diese Dateien zu und verwerten die Daten entsprechend ihrer Funktion. Das Programm „free“ oder „vmstat -s“ geben zum Beispiel die selben Werte aus wie „cat meminfo“, nur in einer leicht anderen Formatierung.

Die einzelnen Prozess-Ordner haben in sich wiederum eine eigene Struktur (wie hier die Instanz eines Apache-Servers) aus der man weitere Details wie Laufzeit oder das aktuelle Arbeitsverzeichnis erfahren kann:

```
root@s15399019:/proc/5479# ls
auxv      cwd      fdinfo   maps     numa_maps  schedstat  statm
cmdline   environ io        mem      oom_adj    smaps      status
coredump_filter  exe      limits   mounts   oom_score  stack      task
cpuset    fd       loginuid mountstats root       stat       wchan
```

Diese Informationen sind für Programme wie „top“ von Nutzen, um detailliertere Informationen über die einzelnen Prozesse anzeigen zu können.

10. Stellenwert der Systemanalyse & Zusammenfassung

Die Entwicklung von Systemanalyseprogrammen bzw. der Analyse generell ist höchstwahrscheinlich genauso alt, wie Computer selbst. Oft kommt es in einem Entwicklungszyklus zu Fehlern oder unvorhergesehenen Problemen, welche analysiert werden müssen. Heute ist diese Art der Analyse eher in der Hintergrund gerückt für den Nutzer und die Überwachung des laufenden Systems wichtig. Auch hierbei kann es zu unvorhergesehenen Problemen kommen (wie Zombieprozesse, die trotz Beendung das System blockieren). Für Systemadministratoren, aber auch Endanwender mit vertieften Kenntnissen sind die oben genannten Werkzeuge Standardprogramme für die alltägliche Arbeit und nehmen mindestens bei Ersteren einen hohen Stellenwert ein.

In dieser Arbeit habe ich eine kleine Übersicht über einige gängige Programme gegeben, welche stellvertretend für jeweilige Einsatzgebiete genutzt werden. Die Einführung gibt eine kleine Übersicht über die Gründe und Arten der Systemanalyse und welcher Sinn dahinter steckt. Die Punkte 2. bis 7. stellen einige gängige Programme dar, von Prozessorüberwachung über Speichermedien bis hin zur Netzwerkanalyse. Im Punkt 8 wurden noch einige andere Programme kurz angerissen, welche sich zum Teil auch mit Funktionen der vorher genannten überdecken. Zum Schluss wurde noch das /proc-Dateisystem etwas näher betrachtet, welches unter anderem als Grundlage für die Daten der hier vorgestellten Programme dient.

Wie bereits mehrfach erwähnt, gibt es oft Überschneidungen oder Gemeinsamkeiten in Programmen unterschiedlicher Einsatzzwecke. So hat zum Beispiel „iostat“ eine Übersicht über die Prozessorauslastung, da eine stark ausgelastete CPU auch die Leistung einer Festplatte ausbremsen kann und somit ein wichtiger Parameter ist.

Da die Themen verteiltes Rechnen und Cloud einen immer höheren Stellenwert einnehmen, könnten sich hier in Zukunft neue Probleme ergeben wie die Übersicht der Gesamtauslastung eines verteilten Systems und somit einige neue interessante und nützliche Programme hervorbringen.

11. Quellen

Quellen

<http://www.cyberciti.biz/tips/top-linux-monitoring-tools.html>

vmstat

http://linuxcommand.org/man_pages/vmstat8.html

<http://wiki.ubuntuusers.de/vmstat>

http://www.thomas-krenn.com/de/wiki/Linux_Performance_Messungen_mit_vmstat

(alle am 19.3.2012 abgerufen)

Top

<http://www.unixtop.org/>

http://de.wikipedia.org/wiki/Top_%28Unix%29

(19.03.2012)

<http://unixhelp.ed.ac.uk/CGI/man-cgi?top>

(20.03.2012)

powertop

<http://www.lesswatts.org/projects/powertop>

<http://wiki.ubuntuusers.de/PowerTOP>

(20.03.2012)

vnstat

<http://wiki.ubuntuusers.de/vnStat>

<http://www.cyberciti.biz/tips/keeping-a-log-of-daily-network-traffic-for-adsl-or-dedicated-remote-linux-box.html>

<http://humdi.net/vnstat/man/vnstat1.html>

<http://humdi.net/vnstat/>

(21.03.2012)

powerTOP

<http://www.heise.de/open/artikel/Powertop-2-0-Strom-sparen-unter-Linux-1167455.html>

iostat

<http://blog.netways.de/2008/11/20/linux-monitoring-iostat/> →

<http://en.wikipedia.org/wiki/Iostat>

<http://linux.die.net/man/1/iostat>

(22.03.2012)

<http://www.tomshardware.de/ssd-raid-io-performance.testberichte-240714-2.html>

(23.03.2012)

/proc

<http://www.oreilly.de/german/freebooks/rlinux3ger/ch057.html>

(19.03.2012)

Ende

<http://www.maximumpc.com/files/u112496/funny-pictures-history-honestly-frank-kb-ram-is-more-than-anyone-will-ever-need.jpg>

(25.03.2012)