DE PRESTATIES UIT HET VERLEDEN

Rakel je performance-gegevens op met atop

Er zijn veel programma's beschikbaar waarmee je de belasting van je systeem kunt monitoren. De meeste van deze programma's geven de systeembelasting van dit moment weer, maar hebben niet de mogelijkheid om de systeembelasting uit het (recente) verleden te tonen. Denk daarbij aan programma's als 'top' of 'vmstat'. Vaak hoor of besef je pas achteraf dat er 'iets' gaande is geweest, dat je nu niet meer kunt analyseren. In deze workshop bekijken we de mogelijkheden om met atop in het verleden te duiken. **Gerlof Langeveld**

Atop is een interactief programma, waarmee je kunt bepalen waarom de prestaties van je systeem (gevoelsmatig) onder de maat zijn. De systeemprestaties worden doorgaans negatief beïnvloed, doordat één van de kritieke hardware-componenten overbelast wordt, zoals de processors, het RAM-geheugen (swapping), de disks of het netwerk. Waar een programma als 'top' alleen de belasting van de processors en het geheugen toont, laat atop ook de belasting van de disks en de netwerk-interfaces zien. Als één van deze componenten zwaar belast blijkt, kun je vervolgens ook zien welke processen deze belasting veroorzaken. Daarbij worden zelfs de gegevens getoond van de processen, die gedurende de interval geëindigd zijn.

De processorbelasting wordt getoond in de regel met het label CPU. De waarden 'sys', 'user' en 'irq' geven gezamenlijk het percentage gebruikte processortijd weer. In dit geval is dat 198% van de 400% in totaal (capaciteit van 4 processors), dus pakweg de helft van de processorcapaciteit was in gebruik. In de proceslijst zie je vervolgens in de kolommen SYSCPU en USRCPU hoeveel processortijd is geconsumeerd door deze processen. Het proces 'find' vind je daar tussen '<' en '>' tekens, die aangeven dat dit proces gedurende de interval geëindigd is, maar nog wel 39% van de processorcapaciteit heeft verbruikt.

```
File  Edit  View  Search  Terminal  Help
ATOP - robin        2018/05/29  17:07:44        --------------        10s elapsed
PRC | sys   11.17s | user   8.98s | #proc    323 | #zombie    0 | #exit      9 |
CPU | sys     104% | user     90% | irq       4% | idle    136% | wait     67% |
CPL | avg1    4.04 | avg5    3.06 | avg15  1.93 | csw  2237514 | intr  123230 |
MEM | tot     7.6G | free  141.0M | cache  4.6G | buff    3.6M | slab    1.1G |
SWP | tot    10.0G | free    9.7G |             | vmcom   6.0G | vmlim  13.8G |
PAG | scan   64078 | steal  62332 | stall      0 | swin       0 | swout     13 |
DSK |          sda | busy     97% | read    4185 | write    381 | avio 2.09 ms |
DSK |          sdb | busy      0% | read      14 | write     26 | avio 0.23 ms |
NET | transport    | tcpi    6732 | tcpo    3264 | udpi       0 | udpo       0 |
NET | network      | ipi     6732 | ipo     3264 | ipfrw      0 | deliv   6732 |
NET | enp2s0    0% | pcki    6732 | pcko    3264 | si  948 Kbps | so  175 Kbps |

  PID SYSCPU USRCPU  VGROW  RGROW  RDDSK  WRDSK  RNET  SNET  CPU  CMD          1/4
 4454  3.28s  4.78s     0K     0K     0K    24K     0     0  82%  gnome-terminal
18636  2.57s  1.21s     0K     0K      -      -     0     0  39%  <find>
18654  1.40s  0.58s 120.7M  4708K     0K     0K     0     0  20%  find
 3231  0.22s  0.69s   -32K   -16K     0K     0K     0     0   9%  gnome-shell
 2142  0.29s  0.48s   616K  -440K     0K     0K     0     0   8%  X
17914  0.45s  0.18s     0K     0K 248.8M     0K     0     0   6%  grep
15519  0.54s  0.00s     0K     0K     0K     0K     0     0   6%  kworker/2:1
16624  0.49s  0.00s     0K     0K     0K     0K     0     0   5%  kworker/3:1
17561  0.48s  0.00s     0K     0K     0K     0K     0     0   5%  kworker/1:1
 3778  0.09s  0.35s     0K   204K    16K    16K     0     0   5%  firefox
16371  0.06s  0.38s   464K   248K    40K     8K    12     8   5%  thunderbird
18190  0.27s  0.13s     0K     0K     0K     0K  6753  3270   4%  ssh
16836  0.40s  0.00s     0K     0K     0K     0K     0     0   4%  kworker
18189  0.19s  0.15s     0K     0K     0K     0K     0     0   3%  ssh
   41  0.14s  0.00s     0K     0K     0K     0K     0     0   1%  kswapd0
```

1. Generieke uitvoer van atop

De geheugenbezetting wordt getoond in de regel met het label MEM, maar belangrijker nog zijn de activiteitentellers bij het label PAG. In laatstgenoemde regel staat de teller 'swout', die de swapout-frequentie aangeeft. Als met name deze teller voortdurend een hoge waarde heeft, zal het systeem zeker traag aanvoelen. In Figuur 1 geeft de kleur cyaan aan dat deze waarde wel te hoog is, maar nog niet kritiek. In de proceslijst zie je de geheugengroei van elk proces in de afgelopen interval, virtueel (VGROW) en resident (RGROW). Voor het bepalen van fysiek geheugengebruik is RGROW belangrijk, maar een continue VGROW duidt mogelijk op een geheugenlek in een proces. De toetsaanslag 'm' geeft meer geheugenwaarden per proces, waaronder de *totale* virtuele en resident grootte.

De diskbelasting (uitgevoerde opdrachten per tijdseenheid) zie je in de regels met het label DSK. Het belangrijkste criterium voor overbelasting is natuurlijk het percentage 'busy'. De waarden van disk 'sda' zijn rood gekleurd in **Figuur 1** om aan te geven dat de belasting van deze disk kritiek is vanuit performance-oogpunt. In de proceslijst geven de kolommen RDDSK (read) en WRDSK (write) aan welk proces de grootste hoeveelheid data heeft gelezen en geschreven.

Tenslotte tonen de regels met het label NET de netwerkactiviteiten, waarbij de onderste regel het netwerk-interface toont (in dit geval

'enp2s0') en de regels daarboven respectievelijk de tellers van de IP- en TCP/UDP-laag. De invoer-snelheid 'si' en uitvoer-snelheid 'so' van het netwerk-interface geven het effectieve gebruik van de verbinding weer. Op basis van deze waarden wordt het busy-percentage van de interface bepaald. De kolommen RNET en SNET in de proceslijst tonen het aantal netwerkpakketten dat ontvangen en verzonden zijn per proces. Voor deze netwerktellers per proces moet je wel de 'netatop' kernel module installeren (zie de website van atop).

Je kunt de uitvoer van atop met allerlei toetsaanslagen wijzigen. De toetsaanslag 'h' (help) toont beknopt alle mogelijkheden. Daarnaast wordt ook uitgebreide documentatie meegeleverd in de vorm van een online manual. Hierin worden ook alle getoonde waarden verklaard.

SESSIES OPNEMEN

Als je atop start zonder parameters krijg je een interactieve meetsessie met een (default) interval van 10 seconden, analoog aan een meting met 'top'. Je kunt met de optie '-w' (gevolgd door een bestandsnaam) ook een meetsessie 'opnemen' met atop om die later op je gemak te analyseren. Daarbij kun je ook de intervaltijd wijzigen en het aantal intervallen aangeven. Voorbeeld:

```
atop -w /tmp/take2 60 10
```

In dit geval worden 10 metingen gedaan met een interval van 60 seconden. De resultaten worden weggeschreven naar het bestand '/tmp/take2' in binaire vorm en gecomprimeerd om het gebruik van diskruimte te beperken.

Naast dit soort ad-hoc metingen, wordt ieder etmaal een standaardmeting gestart. Zo'n meting gebruikt een intervaltijd van 10 minuten en wordt 28 dagen bewaard. Op deze manier kun je altijd vier weken terugblikken op het wel en wee van je systeem. De bestanden per etmaal vind je onder de directory **/var/log/atop** (zie **Listing 2**).

In de bestandsnaam is de datum van het etmaal opgenomen. Een meting van een volledige dag hoeft niet meer te kosten dan enkele MB's.

LISTING 2

```
ls -l /var/log/atop
...
-rw-r--r--  1 root root 2415334 May 30 00:00 atop_20180529
-rw-r--r--  1 root root 2358240 May 30 16:42 atop_20180530
```

SESSIES INTERACTIEF AFSPELEN

Je kunt een opgenomen sessie 'afspelen' met de optie '-r' (gevolgd door een bestandsnaam) van atop:

```
atop -r /tmp/take2
```

Atop toont nu de gegevens van het eerste interval uit het bestand. Met de toetsaanslag 't' kun je telkens om een volgende interval vragen en met toetsaanslag 'T' weer om de voorgaande. De toets 'r' doet een rewind naar het begin van de opgenomen sessie en met toets 'b' kun je naar een bepaald tijdstip springen.

Op soortgelijke wijze kun je ook de standaard meetsessie van een bepaald etmaal bekijken, bijvoorbeeld de sessie van vandaag (tot nu toe):

```
atop -r
```

Of de sessie van eergisteren (iedere 'y' geeft een extra dag terug aan):

```
atop -r yy
```

Of de sessie van een specifieke datum (in formaat YYYYMMDD):

```
atop -r 20180529
```

Je kunt dan dezelfde toetsen 't', 'T', 'r' en 'b' gebruiken om te manoeuvreren binnen de intervallen van het etmaal.

SESSIES BATCH-MATIG AFSPELEN

Als je alle gegevens uit een sessie versneld wilt doorzoeken om bepaalde gegevens te vinden, dan kun je de uitvoer van atop naar een bestand of een pipe sturen. Als atop 'merkt' dat zijn uitvoer niet naar een scherm gaat, wordt alle uitvoer als één stroom gegenereerd en wordt schermopmaak achterwege gelaten. Zo kun je bijvoorbeeld het verloop van de geheugenbezetting van gisteren zichtbaar maken (zie **Listing 7**).

Het nadeel van deze werkwijze is dat je geen tijdstippen ziet bij de regels, aangezien dat tijdstip immers staat in de kopregel van iedere interval en die wordt hier niet getoond. Daarom kun je het verloop van de belasting op systeemniveau beter bekijken met het programma **atopsar** dat ook in het atop package wordt meegeleverd. Atopsar kent vele opties, zoals de optie '-m' voor geheugenbezetting (zie **Listing 8**).

SESSIEGEGEVENS NABEWERKEN

Als je bepaalde waarden uit een meting wilt ontleden en nabewerken, kan het lastig zijn

LISTING 7

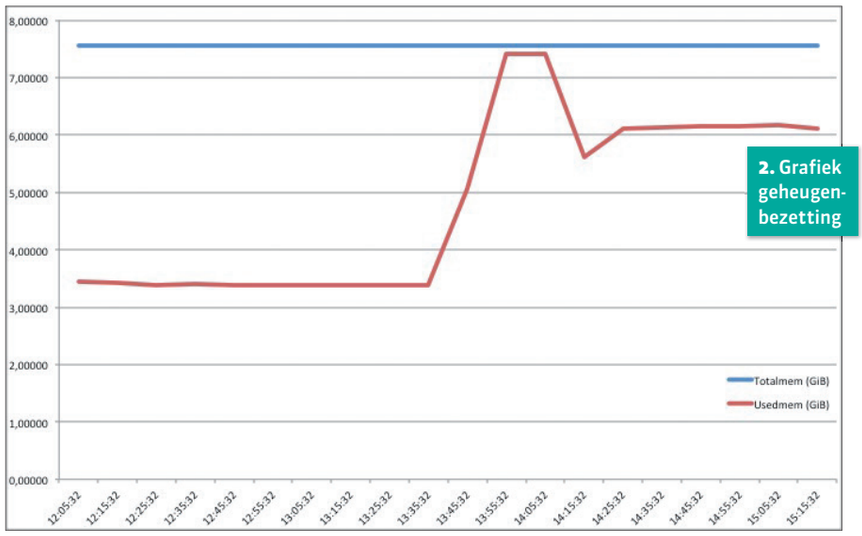
```
atop -r y | grep 'MEM'
MEM | tot 7.6G | free 794.3M | cache 4.4G | buff 7.5M | slab 661.9M |
MEM | tot 7.6G | free 123.5M | cache 4.4G | buff 4.5M | slab 425.2M |
MEM | tot 7.6G | free 137.9M | cache 4.4G | buff 3.7M | slab   1.2G |
...
```

LISTING 8

```
atopsar -m  -r y
16:35:32  memtotal memfree buffers cached dirty slabmem  ... _mem_
16:45:32     7738M    794M      7M  4544M    0M    661M  ...
16:55:32     7738M    123M      4M  4523M    0M    425M  ...
17:05:32     7738M    137M      3M  4478M    0M   1192M  ...
...
```

LISTING 9

```
atop -PMEM  -r y
MEM robin 1527588332 2018/05/29 12:05:32 600 4096 1980934 1077968 299014 4 132372 0 107978 0
105604 11825 0 2097152 0 0
SEP
MEM robin 1527588932 2018/05/29 12:15:32 600 4096 1980934 1081906 305660 4 132364 10 108002 0
106872 11825 0 2097152 0 0
SEP
...
```



2. Grafiek geheugen-bezetting

LISTING 10

```
atop -PMEM -r y | awk -f mem.awk > memory.csv
```

LISTING 11

```
cat mem.awk
BEGIN{OFS=",”; print “Tijd”, “Totalmem (GiB)”, “Usedmem (GiB)”}
/^MEM/{print $5, $8*$7/1024/1024/1024, ($8-$9)*$7/1024/1024/1024}
```

als atop of atopsar al ‘voorgekookte’ waarden (zoals percentages) weergeeft in plaats van de ‘rauwe’ getallen. Daarnaast hebben atop en atopsar de neiging om waarden zo precies mogelijk te tonen binnen een bepaalde kolombreedte. Zo kan het vrije geheugen in de ene interval in Mbytes getoond worden en in de volgende interval in Gbytes, omdat die waarde anders te groot wordt voor de betreffende kolom. Om waarden goed te kunnen nabewerken, biedt atop de mogelijkheid om ‘parseable’ uitvoer te genereren. Stel dat je voor een bepaalde periode een grafiek wilt maken van het totale geheugen versus het geheugen dat in gebruik is. Je kunt dan parseable uitvoer van de geheugengetallen laten genereren met de optie ‘-P’ gevolgd door het label MEM (hetzelfde label dat gebruikt wordt in de schermuitvoer van atop), zie **Listing 9** op de vorige pagina.

In de manual page van atop vind je de beschrijving van de verschillende waarden in een regel. Voor de gewenste grafiek is het vijfde veld van belang (tijdstip), het zevende veld (grootte van een geheugenpagina), het achtste veld (totaal geheugen in pagina’s) en het negende veld (vrij geheugen in pagina’s). De nabewerking kunnen we bijvoorbeeld regelen met awk, waarbij we de invoer voor de grafiek in dit voorbeeld naar het bestand memory.csv laten schrijven (zie **Listing 10**).

Het bestand mem.awk bevat de opdrachten voor awk (zie **Listing 11**).

De BEGIN opdracht wordt eenmalig uitgevoerd en genereert de eerste regel uitvoer met de koppen voor de kolommen. De tweede opdracht werkt op alle data regels die met MEM beginnen en genereert drie kolommen per regel met respectievelijk het tijdstip ($5 refereert aan het vijfde veld), het totale geheugen in Gbytes en het gebruikte geheugen in Gbytes (voor die laatste waarde wordt het vrije geheugen afgetrokken van het totale geheugen). De uitvoer is dan als volgt (**Figuur 2** toont de bijbehorende grafiek).

```
cat memory.csv
Tijd,Totalmem (GiB),Usedmem (GiB)
12:05:32,7.55666,3.44454
12:15:32,7.55666,3.42952
12:25:32,7.55666,3.38194
...
15:05:32,7.55666,6.17453
15:15:32,7.55666,6.11981
```

Als je voor de optie ‘-P’ de labels PRC, PRM, PRD en PRN gebruikt, kun je respectievelijk de CPU-, geheugen-, disk- en netwerk-gebruikstellers per proces nabewerken.

OVERBELASTING ONTDEKKEN

Als je zelf ervaren hebt dat je systeem een bepaalde periode stroperig ‘aanvoelde’ of als je gebruikers daarover geklaagd hebben, dan vind je in atopsar en atop een ideale combinatie om te ontdekken welke hardwarecomponent overbelast is geweest én welk proces dat op z’n geweten heeft. Eerst vraag je met atopsar de belasting van je kritieke componenten op systeemniveau op. Als je het exacte tijdstip van de overbelasting kunt duiden, kun je diezelfde interval met atop bekijken om de belasting te zien van elk proces dat op dat tijdstip draaide.

Stel dat er klachten zijn over de systeemprestaties in de periode rond 17:00 uur gisteren. Dan kunnen we eerst met atopsar de belasting van de vier kritieke componenten bekijken met de opties ‘-c’ (CPU), ‘-s’ (swapping in geval van geheugenoverloop), ‘-d’ (disk) en ‘-i’ (netwerk-interfaces) van gisteren van begintijd 16:45 (optie ‘-b’) tot eindtijd 17:25 (optie ‘-e’). De uitvoer in **Figuur 3** laat zien dat de netwerk-interface

File Edit View Search Terminal Help

```
[gerlof@robin ~]$ atopsar -csdi  -r y  -b 16:45  -e 17:25

------------------------- analysis date: 2018/05/29 ----------

16:45:32  cpu  %usr %nice %sys %irq %softirq  %steal %guest  %wai
16:55:32  all     7    11   11    0        1       0       0   10
17:05:32  all    36    16   29    0        2       0       0   71   247
17:15:32  all    68    33   83    0        2       0       0   23   191
17:25:32  all    12     0    4    0        0       0       0    0   384

16:45:32  pagescan/s   swapin/s  swapout/s        commitspc  commitlim   _swap_
16:55:32     2788.16       0.01      91.34             8092M     14109M
17:05:32     4628.34       8.53      11.84             6159M     14109M
17:15:32     3959.77       8.67       3.95             5539M     14109M
17:25:32        0.00       0.01       0.00             5538M     14109M

16:45:32  disk           busy read/s KB/read  writ/s KB/writ avque avserv _dsk_
16:55:32  sda               1%    0.0      8.0     3.1   128.3   2.8    2.65 ms
17:05:32  sda              54%  283.0     55.7    13.4    15.8  16.5    1.81 ms
17:15:32  sda              37%  337.6     48.3    11.1    15.5   6.9    1.06 ms
17:25:32  sda               1%    0.0      8.0     0.1     5.0   1.8   31.96 ms

16:45:32  interf busy ipack/s opack/s iKbyte/s oKbyte/s imbps ombps maxmbps_if_
16:55:32  enp2s0  86%  7255.6  3645.5    10596      276    86     2      100 f
17:05:32  enp2s0  27%  2505.5  1239.4     3375       92    27     0      100 f
17:15:32  enp2s0   0%   175.4    86.5       35        6     0     0      100 f
17:25:32  enp2s0   0%     3.2     3.0        0        0     0     0      100 f
[gerlof@robin ~]$
```

3. Uitvoer van atopsar voor traag systeem

'enp2s0' weliswaar een hoge belasting heeft, maar dat het aantal swapouts per seconde waarschijnlijk een nog zwaardere impact had gedurende de 10-minuten interval, die eindigt om 16:55 uur.

We kunnen diezelfde interval met atop verder analyseren en vragen daarbij om het geheugengebruik per proces te tonen (optie '-m', analoog aan de 'm' toets binnen atop) voor het specifieke tijdstip 16:55:

```
atop -m  -r y  -b 16:55
```

De uitvoer in **Figuur 4** toont in de MEM regel, dat dit systeem 7.6 GiB geheugen heeft ('tot'), waarvan 4.4 GB gebruikt wordt als page cache ('cache'+'buff') en 425 MB als dynamisch kernelgeheugen ('slab'). In de proceslijst zien we dat het commando grep 1 GB fysiek geheugen gebruikt ('RSIZE') en de daaropvolgende vier processen samen meer dan 1 GB fysiek geheugen gebruiken. Daarmee hebben we al een groot deel van het geheugengebruik in kaart gebracht. Opvallend is natuurlijk dat het commando grep bij het doorspitten van grote hoeveelheden data zelf een behoorlijke omvang krijgt (1 GB). Daarnaast zorgt het doorzoeken van vele bestanden voor een forse groei van de page cache. In dit geval veroorzaken deze factoren dat delen van andere processen uitgeswapt worden, waardoor de gebruikers van die processen het systeem als traag ervaren.

```
File  Edit  View  Search  Terminal  Help
ATOP - robin        2018/05/29  16:55:32        -------------        10m0s elapsed
PRC | sys   95.59s | user   1m45s | #proc    552 | #zombie    0 | #exit    246 |
CPU | sys      11% | user     18% | irq        1% | idle    269% | wait    100% |
CPL | avg1    1.06 | avg5    1.13 | avg15   0.98 | csw  9569310 | intr 6449651 |
MEM | tot     7.6G | free  123.5M | cache   4.4G | buff    4.5M | slab  425.2M |
SWP | tot    10.0G | free    9.7G |              | vmcom   7.9G | vmlim  13.8G |
PAG | scan 1672893 | steal 1562e3 | stall      7 | swin       8 | swout  54803 |
LVM | tos_ssd-home | busy      0% | read   13337 | write    268 | avio 0.19 ms |
LVM | ntos_hdd-var | busy      0% | read      12 | write    147 | avio 12.8 ms |
DSK |          sda | busy      1% | read      13 | write   1715 | avio 2.65 ms |
NFM |  /nfs/Public | srv      nasi | read    5.8G | write     0K | nread   5.0G |
NFC | rpc   206048 | read  192952 | write      0 | retxmit    0 | autref 206e3 |
NET | transport    | tcpi 4109180 | tcpo 2187246 | udpi      43 | udpo      43 |
NET | network      | ipi  4109399 | ipo  2187315 | ipfrw      0 | deliv 4109e3 |
NET | enp2s0   86% | pcki 4353364 | pcko 2187328 | si   86 Mbps | so 2264 Kbps |

  PID VSTACK   VSIZE   RSIZE  PSIZE   VGROW   RGROW SWAPSZ   MEM  CMD        1/111
16606   132K    2.1G    1.0G   1.0G    2.0G    1.0G     0K   13%  grep
 3778   144K    3.0G  463.9M 440.2M  -8196K   -960K     0K    6%
16371   140K    2.7G  266.0M 250.7M   -12K    -800K     0K    3%
 9977   132K    2.0G  235.7M 214.7M     0K    -384K     0K    3%
 3231   140K    2.0G  187.1M 170.5M  -5832K   -9.8M 10620K    2%  gnome-shell
```

4. Uitvoer van atop voor traag systeem

LISTING 15

```
atop -r yyyyy | egrep '^ATOP|chronyd'
....
ATOP - robin    2018/05/25  15:49:01 ...    10m0s elapsed
  1124  0.00s  0.01s  0K  0K  0K  0K  2   3    0% chronyd
ATOP - robin    2018/05/25  15:59:01 ...    10m0s elapsed
  1124  0.00s  0.00s  0K  0K   -   -  1   3    0% <chronyd>
ATOP - robin    2018/05/25  16:09:01 ...    10m0s elapsed
ATOP - robin    2018/05/25  16:19:01 ...    10m0s elapsed
```

LISTING 17

```
  PID ...   ENDDATE     ENDTIME   ST EXC S  CPU CMD
 1124 ... 2018/05/25  15:56:41  -S   9 E   0% <chronyd>
```

VERDWENEN PROCESSEN TRACEREN

Atop is niet alleen nuttig voor performance-analyse, maar ook voor troubleshooting. Stel dat je ontdekt dat de systeemtijd niet meer accuraat is, omdat het daemon-proces 'chronyd' verdwenen is. Dan wil je kunnen achterhalen wanneer dat proces geëindigd is en op welke wijze (uit vrije wil of door ontvangst van een dodelijk signal). Je kunt eerst kijken of het proces eerder vandaag is geëindigd:

```
atop -r | egrep '^ATOP|chronyd'
```

De optie '-r' zorgt voor het lezen van de gegevens sinds middernacht. Als het gezochte proces al eerder dan vandaag is geëindigd, kun je een of meer 'y'-tekens meegeven bij de optie '-r' om telkens een dag verder terug te doorzoeken. De zoekstring voor egrep zorgt dat we niet alleen de regel vinden met het gezochte proces 'chronyd', maar ook de kopregel van iedere interval (regels die beginnen met ATOP) om het tijdstip te kunnen bepalen. Na enige zoeken zou je de volgende uitvoer kunnen krijgen (zie **Listing 15**).

In een interactieve atop sessie kun je vervolgens bepalen waarom chronyd er de brui aan gegeven heeft:

```
atop -v  -r yyyyy  -b 15:59
```

De optie '-v' (of toetsaanslag 'v') zorgt dat atop per proces de start- en eindtijd, én de manier van eindigen toont. Als je binnen atop zoekt naar proces 'chonyd' (/chronyd), dan zou je de volgende uitvoer kunnen krijgen (zie **Listing 17**).

We zien dat chronyd om 15:56:41 is geëindigd. Als reden zien we de letter 'S' als tweede teken in de kolom ST: getermineerd door een *signal* en wel signal 9 (kolom EXC). Dat tweede teken zou ook een 'E' kunnen zijn, bij een vrijwillig einde met de exitcode in de kolom EXC. Maar de cliffhanger na deze workshop blijft: wie vermoordde chronyd?

Het package 'atop' vind je voor de meeste distributies in de standaard repository. In dat geval kun je 'atop' installeren via commando's als yum of apt-get (afhankelijk van de distro). De meest-courante versie vind je altijd op de website www.atoptool.nl als RPM en als tar-ball.

OVER DE AUTEUR

Gerlof Langeveld is de maker van atop, atopsar en kernel module netatop. Hij werkt inmiddels 21 jaar als docent/consultant bij AT Computing. Hij doceert o.a. de vijfdaagse master class "Linux performance analysis and tuning" voor systeembeheerders, waarin ruim aandacht wordt geschonken aan de werking van de kernel en de interpretatie van performance-gegevens, die door tools als atop worden getoond.