How to use SAR to Monitor System Performance in Red Hat Enterprise Linux

SOLUTION VERIFIED - Updated June 22 2016 at 11:27 PM - English ()

Environment

- Red Hat Enterprise Linux 4
- Red Hat Enterprise Linux 5
- Red Hat Enterprise Linux 6
- Red Hat Enterprise Linux 7

Issue

- What tool can I use to monitor the health of my system?
- How to use SAR (System Activity Reporter) from the sysstat package to Monitor System Performance in Red Hat Enterprise Linux OS

- How to view average and historical CPU and memory usage on a system
- How to track system performance
- How to collect system performance data with SAR
- What monitoring tool can check for high load and other causes of system slow down or hangs?

Resolution

What is SAR?

- SAR is a utility used to collect and report system activity. It collects data relating to most core system functions, and writes those metrics to binary data files.
- SAR is also a binary (/usr/bin/sar) which can be used to specifically query an sa## file (eg: sa01) or to request current running statistics from a system.

  Information and statistics collected include

  - CPU / IO / System / Nice / Idle percentages
  - Network Traffic / Network Errors
  - Load Average and Run queue
  - Interrupts
  - Memory Free / Cached / Buffered / Swapped
  - Device usage per Major/Minor number
  - And many others
- SAR is provided by the sysstat package, which also provides other statistical reporting tools, such as iostat. Note that the sysstat package is not installed by default.
  - In RHEL-4 use `up2date` command to install `sysstat` package.

```
[root@example ~]# up2date -i sysstat
```

- In RHEL-5, RHEL-6 and RHEL-7 use `yum` to install `sysstat` package.

```
[root@example ~]# yum install sysstat
```

How does SAR work ?

- SAR writes to log files in `/var/log/sa` . This directory holds two types of files - `sa\#\#` files (binary) and `sar\#\#` files (text).
- The number at the end of the file corresponds to the day of the month that file was recording.
- For example, an sa03 file refers to the 03 day of the month.
- When the sysstat package is installed it places a file into /etc/cron.d/sysstat.
- This sets up two cron jobs.
  - 1 job to record statistics every 10 minutes.
  - 2 job to write the binary `sa\#\#` file to a text `sar\#\#` file once a day (typically right before midnight).
- Additionally, it places a configuration file in `/etc/sysconfig/sysstat` .

SAR cron jobs

- There are two cron jobs in `/etc/cron.d/sysstat`

```
# run system activity accounting tool every 10 minutes
*/10 * * * * root /usr/lib/sa/sa1 1 1

# generate a daily summary of process accounting at 23:53
53 23 * * * root /usr/lib/sa/sa2 -A
```

- If it is desired for SAR to collect data more frequently, simply change " `*/10` " to a new interval.
- For example, if to make SAR to track every 5 minutes, simply change to " `*/5` ".
- **NOTE:** SAR does not add significant load to a server. It safely can be tuned down to 2 minute intervals without seeing a significant problem. SAR also does not grab individual block data.

SAR configuration file

- To capture disk activity, consult: Why does SAR -d result in "Requested activities not available in file" error? (https://access.redhat.com/knowledge/solutions/23173)
- To make SAR track data for more than 7 days, simply change the configuration file:

```
[root@example ~]# vim /etc/sysconfig/sysstat
# How long to keep log files (days), maximum is a month
HISTORY=7     <===
```

Note that RHEL 4/5 sysstat does not support keeping more than 1 month of data; however, in RHEL6 if a `HISTORY` value greater than 28 is declared, SAR log files are automatically split up into separate directories.

How is SAR useful?

SAR is useful in many ways, both directly and indirectly.

- Overall barometer of system performance. When working with a system and not knowing what the "normal" state is, looking at SAR data over the last several production days is useful to establish a baseline of standard activity.

- To get a feel for CPU load, load average, memory usage, etc.
- Detecting system activity leading up to a crash or hang. Again, you can watch system statistics leading up to a fatal event.
    - Did memory usage creep up?
    - Did the IO-wait climb to 100%?
    - Did the devices stop writing to disk? etc.
- Useful for tuning Hangwatch. Since Hangwatch triggers on load average, we need to know what is "normal" and what is "high" load average. Otherwise Hangwatch will fire sysrq-triggers too much, or too little.
- Deep dive into subsystems useful for cross-referencing events with time-stamps.
- For example, "when I start application, I see memory usage spike and IOWait spike, but all writes to the network stop".

Examples

1. Basic Usage

- Print all CPU statistics for today:

```
# sar -P ALL
```

- Select all network statistics from file sa13:

```
# sar -n ALL -f /var/log/sa/sa13
```

- Select all Memory statistics between 10AM and 2 PM from file sa07 and output to file mem.txt

```
# sar -r -s 10:00:00 -e 14:00:00 -f /var/log/sa07 -o /tmp/mem.txt
```

2. Advanced Usage

Is my system leaking memory ?

- If leaking memory is suspected on a system, taking a look at the memory portion of SAR (sar -r) can be very illuminative. In this contrived example, memory usage increase to nearly 100% can be observed, and then swap usage increase to 100% until the box hangs. This would be strong evidence of a memory leak.

NOTE: The time was tuned down to 1 minute intervals. If the default 10 minute intervals aren't giving the resolution needed, remember that SAR's time interval can be tuned so that is appropriate for the problem.

```
time         kbmemfree kbmemused  %memused kbbuffers  kbcached kbswpfree kbswpused  %swpused  kbswpcad
02:10:09 PM    444736     64312     12.63       584     20696    926960     88840      8.75      8984
02:11:01 PM    436160     72888     14.32      1032     29164    927036     88764      8.74      9424
02:12:01 PM    436160     72888     14.32      1048     29164    927036     88764      8.74      9424
02:13:02 PM    435456     73592     14.46      1108     29524    927036     88764      8.74      9648
02:14:01 PM    409440     99608     19.57      1172     31592    927040     88760      8.74      9688
02:15:01 PM    348640    160408     31.51      1200     31616    927040     88760      8.74      9720
02:16:01 PM    286816    222232     43.66      1216     31620    927040     88760      8.74      9720
02:17:01 PM    224992    284056     55.80      1232     31620    927040     88760      8.74      9720
02:18:01 PM    161056    347992     68.36      1260     31860    927040     88760      8.74     11536
02:19:01 PM    100192    408856     80.32      1276     31860    927040     88760      8.74     11568
02:20:01 PM     38176    470872     92.50      1296     31960    927040     88760      8.74     11612
02:21:01 PM     10720    498328     97.89       196     11032    930172     85628      8.43      3176
02:22:01 PM     10848    498200     97.87       200     10432    870320    145480     14.32      1740
02:23:01 PM     12064    496984     97.63       248      9176    806724    209076     20.58      4612
02:24:01 PM     12000    497048     97.64       264      9068    747032    268768     26.46      2576
02:25:01 PM     12064    496984     97.63       284      9052    684732    331068     32.59      2940
02:26:01 PM     10976    498072     97.84       280      9004    626108    389692     38.36      2084
02:27:01 PM     10976    498072     97.84       256      8972    564280    451520     44.45      2080
02:28:01 PM     10976    498072     97.84       320      9112    501764    514036     50.60      2784
02:29:02 PM     12000    497048     97.64       284      9052    440668    575132     56.62      2236
02:30:01 PM     12064    496984     97.63       388     12840    375168    640632     63.07      2920
02:31:01 PM     12192    496856     97.60       404     12648    311024    704776     69.38      5320
02:32:01 PM     10016    499032     98.03       376     12644    252712    763088     75.12      5132
02:33:01 PM     12320    496728     97.58       360      9608    193176    822624     80.98      3588
02:34:01 PM     12064    496984     97.63       532     12592    128540    887260     87.35      3484
02:35:01 PM     10848    498200     97.87       516     12592     68852    946948     93.22      3648
02:36:01 PM     10144    498904     98.01       472     11916      6036   1009764     99.41      6084`
```

2. How should I tune Hangwatch to catch a time when the server is hung ?

- Here is SAR data on load average:

```
16:30:01      runq-sz  plist-sz   ldavg-1   ldavg-5  ldavg-15
16:40:01            0      1358      0.34      0.32      0.23
16:50:01            0      1375      1.27      0.75      0.40
17:00:01            0      1353      1.49      1.32      0.83
17:10:01            2      1357      1.19      1.20      1.00
17:20:01            0      1368      1.25      1.23      1.10
17:30:01            1      1346      1.23      1.30      1.18
17:40:01            5      1357      1.38      1.30      1.22
17:50:01            0      1367     11.32      6.20      3.41
18:00:01            0      1346      7.02      5.42      4.15
18:10:01            0      1356     13.88      9.10      6.41
18:20:01            2      1378      8.21      9.62      7.62
18:30:01            8      1346     19.93     14.77     11.47
18:40:01            1      1355     22.05     25.36     18.83
18:50:02            0      1366     13.88     20.24     20.77
19:00:01           62      1346     46.47     46.68     32.89`
```

- Note that 0.3-1.3 seems to be a normal range for load average. However, during this period, load average climbs to 46.47.
- An educated deduction can be made, based upon the above information, and determine that any load over 15 could be considered abnormal.
- This number can be arrived at in the following way:
  - The server has 8 cores. So a load average of 8 or below will not stress the box.
  - Based upon the above information, under load the box will float to around the 12 level. There are values at 11, 13, and 13 again.
  - There are higher than average values of 19, 22, and 46.
- As such, if hangwatch was tuned to 15, we could capture sysrq data from 18:30 - 18:50, and then again at 19:00.
- Tuning hangwatch is a judgement call - sometimes the values chosen need to be adjusted based upon other

evidence.

For detail information on each switch, Refer man page of sar

```
# man sar
```

Product(s) Red Hat Enterprise Linux (/taxonomy/products/red-hat-enterprise-linux) **Component** sysstat (/components/sysstat)

Category Performance tune (/category/performance-tune)

Tags performance (/tags/performance) performance_tools (/tags/performancetools) performance_tuning (/tags/performancetuning) rhel (/tags/rhel) sar (/tags/sar) tuning (/tags/tuning)

This solution is part of Red Hat's fast-track publication program, providing a huge library of solutions that Red Hat engineers have created while supporting our customers. To give you the knowledge you need the instant it becomes available, these articles may be presented in a raw and unedited form.

---

1 Comments


(/user

Dale Stubblefield (/user/1543903) 27 May 2014 3:09 PM (https://access.redhat.com/solutions/276533#comment-762293)

The SAR data is much easier to interpret with kSar than viewing on the command line.

Reply (/Ajax_comments/reply/276533/762293)

/1543903)

ACTIVE CONTRIBUTOR

135 Points