Linux performance monitoring tools

For Linux I like using the [sysstat](http://pagesperso-orange.fr/sebastien.godard/" \t "_blank) utilities, which are a collection of performance monitoring tools for Linux, including: [sar](http://linux.die.net/man/1/sar" \t "_blank), [sadf](http://linux.die.net/man/1/sadf" \t "_blank), [mpstat](https://en.wikipedia.org/wiki/Mpstat" \t "_blank), iostat, [pidstat](http://linux.die.net/man/1/pidstat" \t "_blank) and related sa tools. I've always had a preference here for mpstat. Note that some of the Linux distributions also offer their own tools for performance management and capacity planning, many of which are RPMs and also available on other distributions. Red Hat Enterprise Linux 5 offers the following tools out of the box: [SystemTap](http://sourceware.org/systemtap/" \t "_blank), [frisk](http://sources.redhat.com/frysk/) and [Oprofile](http://people.redhat.com/wcohen/Oprofile.pdf" \t "_blank).

What about Unix/Linux generic tools that are available on all Unix flavors and Linux distributions? Is there such a thing? Absolutely! If you've decided to go this method for consistency and cost drivers, I would use the [system activity reporter (sar)](http://www.softpanorama.org/Admin/Monitoring/sar.shtml), quite possibly the oldest Unix monitoring utility there is. I really like sar and with its more powerful colleague sadf, you can do reporting from the data that sar retrieves. You can just leave it running in cron. Sar is available on every version of Unix or Linux, though you may have to download the sysstat utilities for your Linux version if you don't see it installed.

**New data center system sizing guidelines**  
Another side of capacity planning relates to systems sizing. When you are building a system, you don't usually have the luxury of referencing historic data.

On the application side, there are several tools that will help you size the type of system that you will need. [HP LoadRunner](https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-126-17%5e8_4000_100__) is probably the most well-known tool. It actually simulates user-workloads that allow the architect or capacity planner to test various back-end configurations to determine the right sizing. Open source equivalents of this system; include products like [JMeter](http://jakarta.apache.org/jmeter/" \t "_blank) and [The Grinder](http://grinder.sourceforge.net/). [Oracle's Hyperion](http://www.oracle.com/appserver/business-intelligence/hyperion-financial-performance-management/hyperion-financial-performance-management.html) is another tool worth looking at.

It needs to be mentioned how important it is when sizing boxes for future growth, one must fully engage the business and the application to get a better understanding of the client's plans. For example, while your statistics may show a 25% increased utilization, you may not know that the banking division is going to be sold, which means that when purchasing your next systems, you should not go with the statistic alone. You need to know if the client will be rightsizing, downsizing or perhaps brining in new business that may mean that you might need 400% more iron than you think. Perhaps the most compelling reason for doing capacity planning is that if done effectively, it can help avoid costly hardware upgrades through your trending and performance data. Another reason, is that if done properly, you can save your organization big money and be the superstar you *know* you are.

**How to Produce and Deliver System Activity Reports Using Linux Toolsets – Part 3**

Besides the well-known native Linux tools that are used to check disk, memory, and CPU usage – to name a few examples, Red Hat Enterprise Linux 7 provides two additional toolsets to enhance the data you can collect for your reports: **sysstat** and **dstat**.

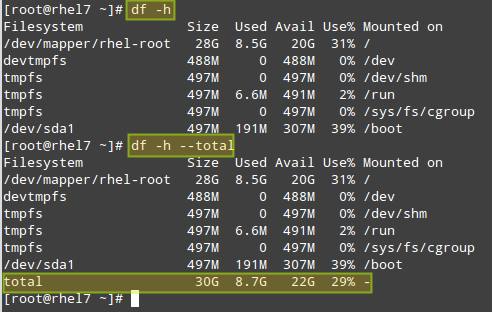
In this article we will describe both, but let’s first start by reviewing the usage of the classic tools.

### Native Linux Tools

With **df**, you will be able to report disk space and inode usage of by filesystem. You need to monitor both because a lack of space will prevent you from being able to save further files (and may even cause the system to crash), just like running out of inodes will mean you can’t link further files with their corresponding data structures, thus producing the same effect: you won’t be able to save those files to disk.

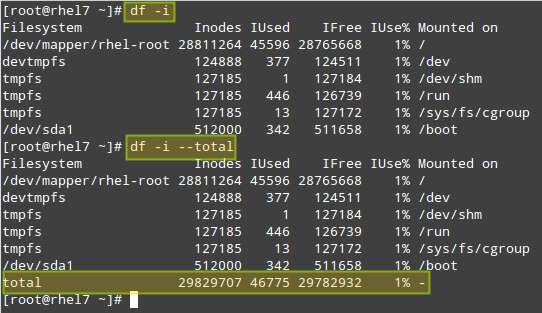
# df -h [Display output in human-readable form]

# df -h --total [Produce a grand total]

[](https://www.tecmint.com/wp-content/uploads/2015/08/Check-Linux-Total-Disk-Usage.png)Check Linux Total Disk Usage

# df -i [Show inode count by filesystem]

# df -i --total [Produce a grand total]

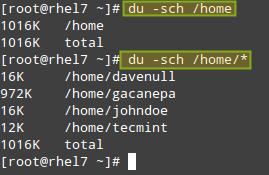
[](https://www.tecmint.com/wp-content/uploads/2015/08/Check-Linux-Total-inode-Numbers.png)Check Linux Total inode Numbers

With **du**, you can estimate file space usage by either file, directory, or filesystem.

For example, let’s see how much space is used by the **/home** directory, which includes all of the user’s personal files. The first command will return the overall space currently used by the entire **/home** directory, whereas the second will also display a disaggregated list by sub-directory as well:

# du -sch /home

# du -sch /home/\*

[](https://www.tecmint.com/wp-content/uploads/2015/08/Check-Linux-Directory-Disk-Size.png)Check Linux Directory Disk Size

**Don’t Miss**:

1. [12 ‘df’ Command Examples to Check Linux Disk Space Usage](https://www.tecmint.com/how-to-check-disk-space-in-linux/)
2. [10 ‘du’ Command Examples to Find Disk Usage of Files/Directories](https://www.tecmint.com/check-linux-disk-usage-of-files-and-directories/)

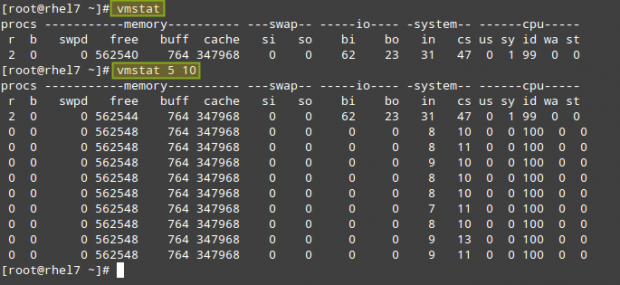
Another utility that can’t be missing from your toolset is **vmstat**. It will allow you to see at a quick glance information about processes, CPU and memory usage, disk activity, and more.

If run without arguments, **vmstat** will return averages since the last reboot. While you may use this form of the command once in a while, it will be more helpful to take a certain amount of system utilization samples, one after another, with a defined time separation between samples.

For example,

# vmstat 5 10

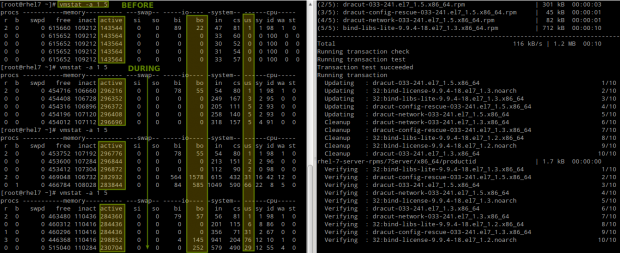
will return 10 samples taken every 5 seconds:

[](https://www.tecmint.com/wp-content/uploads/2015/08/Check-Linux-Systerm-Performance.png)Check Linux System Performance

As you can see in the above picture, the output of vmstat is divided by columns: **procs** (processes), **memory**, **swap**, **io**, **system**, and **cpu**. The meaning of each field can be found in the **FIELD DESCRIPTION** sections in the man page of **vmstat**.

Where can **vmstat** come in handy? Let’s examine the behavior of the system before and during a **yum update**:

# vmstat -a 1 5

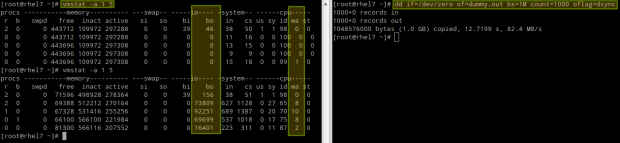
[](https://www.tecmint.com/wp-content/uploads/2015/08/Vmstat-Linux-Peformance-Monitoring.png)Vmstat Linux Performance Monitoring

Please note that as files are being modified on disk, the amount of **active** memory increases and so does the number of blocks written to disk **(bo)** and the CPU time that is dedicated to user processes **(us)**.

Or during the saving process of a large file directly to disk (caused by **dsync**):

# vmstat -a 1 5

# dd if=/dev/zero of=dummy.out bs=1M count=1000 oflag=dsync

[](https://www.tecmint.com/wp-content/uploads/2015/08/VmStat-Linux-Disk-Performance-Monitoring.png)VmStat Linux Disk Performance Monitoring

In this case, we can see a yet larger number of blocks being written to disk **(bo)**, which was to be expected, but also an increase of the amount of CPU time that it has to wait for I/O operations to complete before processing tasks **(wa)**.

**Don’t Miss**: [Vmstat – Linux Performance Monitoring](https://www.tecmint.com/linux-performance-monitoring-with-vmstat-and-iostat-commands/" \t "_blank)

### Other Linux Tools

As mentioned in the introduction of this chapter, there are other tools that you can use to check the system status and utilization (they are not only provided by **Red Hat** but also by other major distributions from their officially supported repositories).

The **sysstat** package contains the following utilities:

1. **sar** (collect, report, or save system activity information).
2. **sadf** (display data collected by sar in multiple formats).
3. **mpstat** (report processors related statistics).
4. **iostat** (report CPU statistics and I/O statistics for devices and partitions).
5. pidstat (report statistics for Linux tasks).
6. **nfsiostat** (report input/output statistics for NFS).
7. **cifsiostat** (report CIFS statistics) and
8. **sa1** (collect and store binary data in the system activity daily data file.
9. **sa2** (write a daily report in the **/var/log/sa** directory) tools.

whereas **dstat** adds some extra features to the functionality provided by those tools, along with more counters and flexibility. You can find an overall description of each tool by running **yum info sysstat** or **yum info dstat**, respectively, or checking the individual man pages after installation.

To install both packages:

# yum update && yum install sysstat dstat

The main configuration file for **sysstat** is **/etc/sysconfig/sysstat**. You will find the following parameters in that file:

# How long to keep log files (in days).

# If value is greater than 28, then log files are kept in

# multiple directories, one for each month.

**HISTORY=28**

# Compress (using gzip or bzip2) sa and sar files older than (in days):

**COMPRESSAFTER=31**

# Parameters for the system activity data collector (see sadc manual page)

# which are used for the generation of log files.

**SADC\_OPTIONS="-S DISK"**

# Compression program to use.

**ZIP="bzip2"**

When **sysstat** is installed, two cron jobs are added and enabled in **/etc/cron.d/sysstat**. The first job runs the system activity accounting tool every **10 minutes** and stores the reports in **/var/log/sa/saXX** where **XX** is the day of the month.

Thus, **/var/log/sa/sa05** will contain all the system activity reports from the 5th of the month. This assumes that we are using the default value in the **HISTORY** variable in the configuration file above:

\*/10 \* \* \* \* root /usr/lib64/sa/sa1 1 1

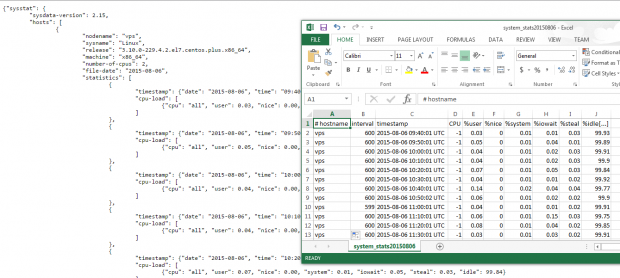
The second job generates a daily summary of process accounting at **11:53** pm every day and stores it in **/var/log/sa/sarXX** files, where **XX** has the same meaning as in the previous example:

53 23 \* \* \* root /usr/lib64/sa/sa2 -A

For example, you may want to output system statistics from **9:30 am** through **5:30 pm** of the sixth of the month to a **.csv** file that can easily be viewed using **LibreOffice Calc** or **Microsoft Excel** (this approach will also allow you to create charts or graphs):

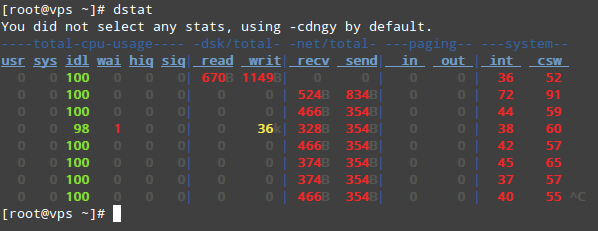
# sadf -s 09:30:00 -e 17:30:00 -dh /var/log/sa/sa06 -- | sed 's/;/,/g' > system\_stats20150806.csv

You could alternatively use the **-j** flag instead of **-d** in the **sadf** command above to output the system stats in **JSON** format, which could be useful if you need to consume the data in a web application, for example.

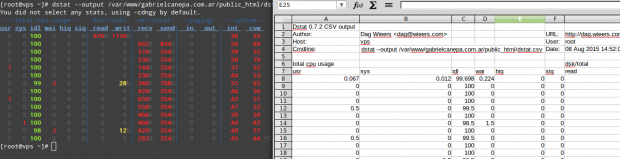
[](https://www.tecmint.com/wp-content/uploads/2015/08/Linux-System-Statistics.png)Linux System Statistics

Finally, let’s see what **dstat** has to offer. Please note that if run without arguments, **dstat** assumes **-cdngy** by default (short for CPU, disk, network, memory pages, and system stats, respectively), and adds one line every second (execution can be interrupted anytime with **Ctrl + C**):

# dstat

[](https://www.tecmint.com/wp-content/uploads/2015/08/dstat-command.png)Linux Disk Statistics Monitoring

To output the stats to a **.csv** file, use the **–output** flag followed by a file name. Let’s see how this looks on LibreOffice Calc:

[](https://www.tecmint.com/wp-content/uploads/2015/08/Monitor-Linux-Statistics-Output.png)Monitor Linux Statistics Output

I strongly advise you to check out the man page of [dstat](https://www.tecmint.com/wp-content/pdf/dstat.pdf" \t "_blank) along with the man page of [sysstat](https://www.tecmint.com/wp-content/pdf/sysstat.pdf" \t "_b) in PDF format for your reading convenience. You will find several other options that will help you create custom and detailed system activity reports.

# Collect and report Linux System Activity Information with sar

[Main Page](https://www.thomas-krenn.com/en/wiki/Main%20Page) > [Server Software](https://www.thomas-krenn.com/en/wiki/Category:Server_Software) > [Linux](https://www.thomas-krenn.com/en/wiki/Category:Linux)  
[Main Page](https://www.thomas-krenn.com/en/wiki/Main%20Page) > [Server Software](https://www.thomas-krenn.com/en/wiki/Category:Server_Software) > [Linux](https://www.thomas-krenn.com/en/wiki/Category:Linux) > [Linux Performance](https://www.thomas-krenn.com/en/wiki/Category:Linux_Performance)

﻿Under Linux, **sar** serves to log and evaluate a variety of information regarding system activity. With performance problems, sar also permits retroactive analysis of the load values for various sub-systems (CPUs, memory, disks, interrupts, network interfaces and so forth) and limitation of problems in this manner.

## Contents

 [hide]

* [1 Construction](https://www.thomas-krenn.com/en/wiki/Collect_and_report_Linux_System_Activity_Information_with_sar#Construction)
* [2 Versions](https://www.thomas-krenn.com/en/wiki/Collect_and_report_Linux_System_Activity_Information_with_sar#Versions)
* [3 Installation](https://www.thomas-krenn.com/en/wiki/Collect_and_report_Linux_System_Activity_Information_with_sar#Installation)
* [4 Functional Approach of Logging](https://www.thomas-krenn.com/en/wiki/Collect_and_report_Linux_System_Activity_Information_with_sar#Functional_Approach_of_Logging)
* [5 Displaying the recorded data using sar](https://www.thomas-krenn.com/en/wiki/Collect_and_report_Linux_System_Activity_Information_with_sar#Displaying_the_recorded_data_using_sar)
  + [5.1 Limiting the Display Period](https://www.thomas-krenn.com/en/wiki/Collect_and_report_Linux_System_Activity_Information_with_sar#Limiting_the_Display_Period)
  + [5.2 Parameters for Various Sub-systems](https://www.thomas-krenn.com/en/wiki/Collect_and_report_Linux_System_Activity_Information_with_sar#Parameters_for_Various_Sub-systems)
    - [5.2.1 Displaying the CPU Load: sar -P ALL](https://www.thomas-krenn.com/en/wiki/Collect_and_report_Linux_System_Activity_Information_with_sar#Displaying_the_CPU_Load:_sar_-P_ALL)
    - [5.2.2 Displaying the RAM Load: sar -r](https://www.thomas-krenn.com/en/wiki/Collect_and_report_Linux_System_Activity_Information_with_sar#Displaying_the_RAM_Load:_sar_-r)
    - [5.2.3 Displaying the Load Average: sar -q](https://www.thomas-krenn.com/en/wiki/Collect_and_report_Linux_System_Activity_Information_with_sar#Displaying_the_Load_Average:_sar_-q)
    - [5.2.4 Displaying the I/O Transfer Rate: sar -b](https://www.thomas-krenn.com/en/wiki/Collect_and_report_Linux_System_Activity_Information_with_sar#Displaying_the_I.2FO_Transfer_Rate:_sar_-b)
    - [5.2.5 Displaying the Network Statistics: sar -n DEV](https://www.thomas-krenn.com/en/wiki/Collect_and_report_Linux_System_Activity_Information_with_sar#Displaying_the_Network_Statistics:_sar_-n_DEV)
    - [5.2.6 Displaying All Values](https://www.thomas-krenn.com/en/wiki/Collect_and_report_Linux_System_Activity_Information_with_sar#Displaying_All_Values)
* [6 Tools for Visual Assessment](https://www.thomas-krenn.com/en/wiki/Collect_and_report_Linux_System_Activity_Information_with_sar#Tools_for_Visual_Assessment)
* [7 References](https://www.thomas-krenn.com/en/wiki/Collect_and_report_Linux_System_Activity_Information_with_sar#References)

## Construction

sar is a component of sysstat.[[1]](https://www.thomas-krenn.com/en/wiki/Collect_and_report_Linux_System_Activity_Information_with_sar#cite_note-1) sar itself consists of the following components:

* [sar(1)](http://pagesperso-orange.fr/sebastien.godard/man_sar.html) – displays the recorded data.
* [sadc(8)](http://pagesperso-orange.fr/sebastien.godard/man_sadc.html) - system activity data collector, which records the data regarding system activity in a binary format.
* [sa1(8)](http://pagesperso-orange.fr/sebastien.godard/man_sa1.html) – a BASH script used by sadc in background; called every ten minutes by cron (from /etc/cron.d/sysstat).
* [sa2(8)](http://pagesperso-orange.fr/sebastien.godard/man_sa2.html) – a BASH script used for writing the daily reports and called by cron once per day (from /etc/cron.d/sysstat).
* [sadf(1)](http://pagesperso-orange.fr/sebastien.godard/man_sadf.html) – displays the recorded data; in contrast with sar provides various formats (CSV, XML and so forth).

## Versions

Development of the sysstat package (which contains sar among others) is ongoing. The following schema is used for the version numbers:[[2]](https://www.thomas-krenn.com/en/wiki/Collect_and_report_Linux_System_Activity_Information_with_sar" \l "cite_note-2)

* Version numbers with x.0 each indicate a stable version
* Version numbers with x.1 each indicate a version in development

Newer versions generally provide additional features. The features of sar -x and sar -X were exported to a separate command as of version 7.1.6 ([pidstat](http://pagesperso-orange.fr/sebastien.godard/man_pidstat.html)).[[3]](https://www.thomas-krenn.com/en/wiki/Collect_and_report_Linux_System_Activity_Information_with_sar#cite_note-3)

## Installation

As noted, sar is a component of the sysstat package, which is already contained in numerous distribution files and can be easily installed, depending on the distribution file, either by yum install sysstat or apt-get install sysstat. Generally at the same time, the cron jobs will be created in /etc/cron.d/sysstat.

## Functional Approach of Logging

sa1 and sa2 will store the recorded data in the directories, /var/log/sa/ (RHEL/CentOS) or /var/log/sysstat/ (Debian). The data will be retained for a week retroactively in the default configuration. This duration can be increased to 28 days (in /etc/sysconfig/sysstat under RHEL/CentOS or in /etc/sysstat/config under Debian 4.0 and /etc/sysstat/sysstat under Debian 5.0).

**Activating sar Logging in Debian:** To activate automatic logging under Debian, the "ENABLED" parameters must be set to "true" in the /etc/default/sysstat file. This parameter has been set to "false" by default.

Here is an example of this:

[root@testserver ~]# cat /etc/sysconfig/sysstat

# How long to keep log files (days), maximum is a month

HISTORY=7

[root@testserver ~]# cat /etc/cron.d/sysstat

# run system activity accounting tool every 10 minutes

\*/10 \* \* \* \* root /usr/lib64/sa/sa1 1 1

# generate a daily summary of process accounting at 23:53

53 23 \* \* \* root /usr/lib64/sa/sa2 -A

[root@testserver ~]# ls -l /var/log/sa/

total 8952

-rw-r--r-- 1 root root 410352 May 18 23:50 sa18

-rw-r--r-- 1 root root 410352 May 19 23:50 sa19

-rw-r--r-- 1 root root 410352 May 20 23:50 sa20

-rw-r--r-- 1 root root 410352 May 21 23:50 sa21

-rw-r--r-- 1 root root 410352 May 22 23:50 sa22

-rw-r--r-- 1 root root 410352 May 23 23:50 sa23

-rw-r--r-- 1 root root 410352 May 24 23:50 sa24

-rw-r--r-- 1 root root 410352 May 25 23:50 sa25

-rw-r--r-- 1 root root 253712 May 26 14:40 sa26

-rw-r--r-- 1 root root 611085 May 17 23:53 sar17

-rw-r--r-- 1 root root 611129 May 18 23:53 sar18

-rw-r--r-- 1 root root 611127 May 19 23:53 sar19

-rw-r--r-- 1 root root 611127 May 20 23:53 sar20

-rw-r--r-- 1 root root 611085 May 21 23:53 sar21

-rw-r--r-- 1 root root 611100 May 22 23:53 sar22

-rw-r--r-- 1 root root 611085 May 23 23:53 sar23

-rw-r--r-- 1 root root 611085 May 24 23:53 sar24

-rw-r--r-- 1 root root 611125 May 25 23:53 sar25

[root@testserver ~]#

## Displaying the recorded data using sar

### Limiting the Display Period

sar can display either the data for the current day (which does not require any special parameters) or the data for a specific day (such as with the -f /var/log/sa/sa21 parameter).

The query period can be further limited by the following parameters:

* -s [ hh:mm:ss ] - sets the starting time (such as -s 07:30:00); if a specific time is not specified, 08:00:00 will be used as the default starting time.
* -s [ hh:mm:ss ] - sets the ending time (such as -s 19:30:00); if a specific time is not specified, 18:00:00 will be used as the default ending time.

### Parameters for Various Sub-systems

Here are several examples for displaying various values (the values originated on a system using CentOS 4.7 with sysstat version 5.0.5).

#### Displaying the CPU Load: sar -P ALL

The values for all CPUs (-P ALL) will be displayed starting from the 25th of the current month (-f /var/log/sa/sa25).

[root@testserver ~]# sar -P ALL -f /var/log/sa/sa25

Linux 2.6.9-023stab048.6-smp (testserver.thomas-krenn.com) 05/25/09

00:00:01 CPU  %user  %nice  %system  %iowait  %idle

00:10:01 all 9.63 0.00 3.81 6.20 80.36

00:10:01 0 8.69 0.00 3.63 7.18 80.50

00:10:01 1 9.28 0.00 3.96 4.47 82.29

00:10:01 2 10.22 0.00 3.86 6.31 79.62

00:10:01 3 9.84 0.00 3.66 6.91 79.59

00:10:01 4 9.86 0.00 3.94 5.28 80.92

00:10:01 5 9.97 0.00 3.84 6.37 79.81

00:10:01 6 9.90 0.00 4.04 7.21 78.85

00:10:01 7 9.32 0.00 3.54 5.83 81.32

[...]

Average: all 9.21 0.00 1.68 0.27 88.84

Average: 0 9.03 0.00 1.70 0.31 88.97

Average: 1 9.20 0.00 1.67 0.25 88.88

Average: 2 9.45 0.00 1.70 0.28 88.56

Average: 3 9.84 0.00 1.68 0.26 88.22

Average: 4 9.23 0.00 1.68 0.26 88.84

Average: 5 9.05 0.00 1.68 0.27 89.00

Average: 6 8.90 0.00 1.68 0.29 89.14

Average: 7 8.97 0.00 1.66 0.28 89.10

[root@testserver ~]#

#### Displaying the RAM Load: sar -r

The RAM values (-r) for the current day will be displayed here. A value of approximately 100 for the percentage of memory used is thoroughly positive (if swapping is not required). See also the [Operating System Caches section of the RAID Controller and Hard Disk Cache Settings article](https://www.thomas-krenn.com/en/wiki/RAID_Controller_and_Hard_Disk_Cache_Settings#Operating_System_Cache).

[root@testserver ~]# sar -r

Linux 2.6.9-023stab048.6-smp (testserver.thomas-krenn.com) 05/26/09

00:00:01 kbmemfree kbmemused  %memused kbbuffers kbcached kbswpfree kbswpused  %swpused kbswpcad

00:10:01 67248 16292680 99.59 42320 12502420 2047648 628 0.03 0

00:20:01 129776 16230152 99.21 47020 12527100 2047648 628 0.03 0

[...]

14:50:01 638680 15721248 96.10 158392 6953128 2047648 628 0.03 0

15:00:01 646272 15713656 96.05 159880 6966980 2047648 628 0.03 0

15:10:01 59784 16300144 99.63 132044 6565036 2047648 628 0.03 0

Average: 554290 15805638 96.61 124889 10992802 2047648 628 0.03 0

[root@testserver ~]#

#### Displaying the Load Average: sar -q

The average load values (-q) for the current day will be displayed here.

[root@testserver ~]# sar -q

Linux 2.6.9-023stab048.6-smp (testserver.thomas-krenn.com) 05/26/09

00:00:01 runq-sz plist-sz ldavg-1 ldavg-5 ldavg-15

00:10:01 0 590 1.62 1.89 1.21

00:20:01 11 574 0.24 0.63 0.86

[...]

15:10:01 7 671 2.36 2.26 2.03

15:20:01 9 651 1.85 1.96 1.99

Average: 7 600 1.02 1.04 1.00

[root@testserver ~]#

#### Displaying the I/O Transfer Rate: sar -b

The I/O transfer rates (-r) for the current day will be displayed here.

[root@testserver ~]# sar -b

Linux 2.6.9-023stab048.6-smp (testserver.thomas-krenn.com) 05/26/09

00:00:01 tps rtps wtps bread/s bwrtn/s

00:10:01 402.21 197.20 205.00 35133.80 27616.04

00:20:01 66.61 56.36 10.24 2034.96 288.93

[...]

15:20:01 72.43 6.54 65.89 176.29 2720.35

15:30:01 73.28 6.08 67.20 399.04 3516.21

Average: 40.21 5.57 34.64 585.00 1561.78

[root@testserver ~]#

**tps** - Total number of transfers per second that were issued to physical devices. A transfer is an I/O request to a physical device. Multiple logical requests can be combined into a single I/O request to the device. A transfer is of indeterminate size

**rtps** - Total number of read requests per second issued to physical devices

**wtps** - Total number of write requests per second issued to physical devices

**bread/s** - Total amount of data read from the devices in blocks per second. Blocks are equivalent to sectors with 2.4 kernels and newer and therefore have a size of 512 bytes. With older kernels, a block is of indeterminate size.

**bwrtn/s** - Total amount of data written to devices in blocks per second.

#### Displaying the Network Statistics: sar -n DEV

The statistical network values (-n) for the current day will be displayed here.

[root@testserver ~]# sar -n DEV

Linux 2.6.9-023stab048.6-smp (testserver.thomas-krenn.com) 05/26/09

00:00:01 IFACE rxpck/s txpck/s rxbyt/s txbyt/s rxcmp/s txcmp/s rxmcst/s

00:10:01 lo 0.82 0.82 177.69 177.69 0.00 0.00 0.00

00:10:01 eth0 31.56 40.21 4537.41 35891.07 0.00 0.00 0.00

00:10:01 eth1 4876.29 9919.22 451754.61 14927283.09 0.00 0.00 0.00

00:10:01 venet0 52.88 44.19 46547.68 15279.15 0.00 0.00 0.00

[...]

15:30:01 lo 0.82 0.82 176.68 176.68 0.00 0.00 0.00

15:30:01 eth0 113.08 155.70 17825.90 188643.77 0.00 0.00 0.00

15:30:01 eth1 634.80 1268.92 58495.76 1888004.02 0.00 0.00 0.00

15:30:01 venet0 189.86 147.08 213164.21 42532.10 0.00 0.00 0.00

Average: lo 0.82 0.82 177.10 177.10 0.00 0.00 0.00

Average: eth0 56.42 78.90 9104.66 94428.28 0.00 0.00 0.00

Average: eth1 281.59 559.55 25919.74 830592.96 0.00 0.00 0.00

Average: venet0 87.26 64.75 99974.13 14933.05 0.00 0.00 0.00

[root@testserver ~]#

#### Displaying All Values

The collective values for all devices (CPU, RAM, etc.) can be displayed using sar -A.

## Tools for Visual Assessment

The following tools represent the values recorded by sar visually.

* [sarface](http://sourceforge.net/projects/sarface/)
* [Linux Performance Analysis using kSar](https://www.thomas-krenn.com/en/wiki/Linux_Performance_Analysis_using_kSar)
* [sar2png](https://github.com/sqrt529/sar2png) (A Perl Script for creating PNG files from sar data, tested under GNU/Linux and Solaris 10)
* For additional tools, see [Sysstat FAQs](http://pagesperso-orange.fr/sebastien.godard/faq.html) Question 2.11. *Do you know a tool which can graphically plot the data collected by sar?*

## References

1. [Jump up↑](https://www.thomas-krenn.com/en/wiki/Collect_and_report_Linux_System_Activity_Information_with_sar#cite_ref-1) [Sysstat Documentation](http://pagesperso-orange.fr/sebastien.godard/documentation.html)
2. [Jump up↑](https://www.thomas-krenn.com/en/wiki/Collect_and_report_Linux_System_Activity_Information_with_sar#cite_ref-2) [Sysstat Download Page](http://pagesperso-orange.fr/sebastien.godard/download.html)
3. [Jump up↑](https://www.thomas-krenn.com/en/wiki/Collect_and_report_Linux_System_Activity_Information_with_sar#cite_ref-3) [Sysstat Change log up to Version 8.0.4](http://pagesperso-orange.fr/sebastien.godard/old_changelog.html), [Sysstat Change log from Version 8.1.1](http://pagesperso-orange.fr/sebastien.godard/changelog.html)