|  |
| --- |
|  |
| SAR Command line Utility |
| KSAR – Graphical SAR analysis tool |
|  |
| **Shalaj Shukla** |
| **11/3/2016** |

|  |
| --- |
|  |

Contents

[SAR Command 2](#_Toc466016239)

[Install and Configure Sysstat 2](#_Toc466016240)

[CPU Usage of ALL CPUs (sar -u) 3](#_Toc466016241)

[CPU Usage of Individual CPU or Core (sar -P) 4](#_Toc466016242)

[Memory Free and Used (sar -r) 5](#_Toc466016243)

[Swap Space Used (sar -S) 5](#_Toc466016244)

[Overall I/O Activities (sar -b) 6](#_Toc466016245)

[Individual Block Device I/O Activities (sar -d) 6](#_Toc466016246)

[Display context switch per second (sar -w) 7](#_Toc466016247)

[Reports run queue and load average (sar -q) 7](#_Toc466016248)

[Report network statistics (sar -n) 8](#_Toc466016249)

[Report Sar Data Using Start Time (sar -s) 9](#_Toc466016250)

[Enable sar Logging 10](#_Toc466016251)

[KSAR Graphical SAR Analysis Tool 12](#_Toc466016252)

[Download And Install kSar 12](#_Toc466016253)

[Run KSAR 12](#_Toc466016254)

[Check Statistics for particular day 17](#_Toc466016255)

# SAR Command

SAR stands for System Activity Report, as its name suggest sar command is used to collect, report & save CPU, Memory, I/O usage in Unix like operating system. SAR command produces the reports on the fly and can also save the reports in the log files as well.

SAR is one of the utility inside sysstat. You can easily download and install it in your machine very easily through YUM, rpm or apt-get according to your OS.

## Install and Configure Sysstat

First, make sure the latest version of sar is available on your system. Install it using any one of the following methods depending on your distribution.

sudo apt-get install sysstat

(or)

yum install sysstat

(or)

rpm -ivh sysstat-10.0.0-1.i586.rpm

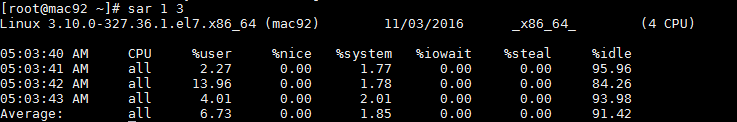
To check the version of SAR use

|  |
| --- |
| sar -V |



Finally, make sure sar works. For example, the following gives the system CPU statistics 3 times (with 1 second interval).

|  |
| --- |
| sar 1 3 |



All the sar command will have the following as the 1st line in its output.

|  |
| --- |
| [root@mac92 ~]# sar -u 1 3  Linux 3.10.0-327.36.1.el7.x86\_64 (mac92) 11/03/2016 \_x86\_64\_ (4 CPU) |

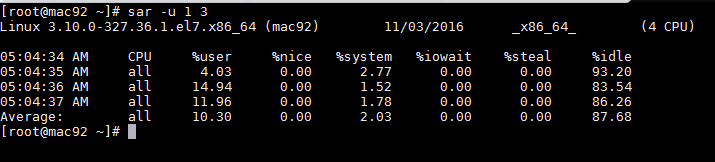
* Linux 3.10.0-327.36.1.el7.x86\_64 – Linux kernel version of the system.
* (mac92) – The hostname where the sar data was collected.
* 11/03/2016 – The date when the sar data was collected.
* \_x86\_64\_ – The system architecture
* (4 CPU) – Number of CPUs available on this system. On multi core systems, this indicates the total number of cores.

You can check manual of sar command using **man sar**

Below are some important sar options

## CPU Usage of ALL CPUs (sar -u)

This gives the cumulative real-time CPU usage of all CPUs. Option “1 3”- reports for every 1 second, total of 3 times. Most likely you’ll focus on the last field “%idle” to see the CPU load



Following are few variations:

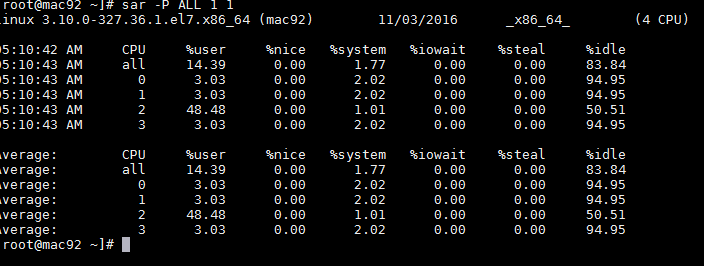
* **sar -u** Displays CPU usage for the current day that was collected until that point.
* **sar -u 1 3** Displays real time CPU usage every 1 second for 3 times.
* **sar -u ALL** Same as “sar -u” but displays additional fields.
* **sar -u ALL 1 3** Same as “sar -u 1 3” but displays additional fields.
* **sar -u -f /var/log/sa/sa10** Displays CPU usage for the 10day of the month from the sa10 file.

## CPU Usage of Individual CPU or Core (sar -P)

If you have 4 Cores on the machine and would like to see what the individual cores are doing, do the following.

“-P ALL” indicates that it should displays statistics for ALL the individual Cores.

In the following example under “CPU” column 0, 1, 2, and 3 indicates the corresponding CPU core numbers.

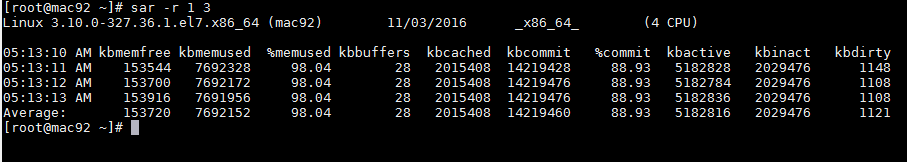


Following are few variations:

* **sar -P ALL** Displays CPU usage broken down by all cores for the current day.
* **sar -P ALL 1 3** Displays real time CPU usage for ALL cores every 1 second for 3 times (broken down by all cores).
* **sar -P 1** Displays CPU usage for core number 1 for the current day.
* **sar -P 1 1 3** Displays real time CPU usage for core number 1, every 1 second for 3 times.
* **sar -P ALL -f /var/log/sa/sa10** Displays CPU usage broken down by all cores for the 10day day of the month from sa10 file.

## Memory Free and Used (sar -r)

This reports the memory statistics. Option - “1 3” reports for every 1 second a total of 3 times. Most likely you’ll focus on “kbmemfree” and “kbmemused” for free and used memory.

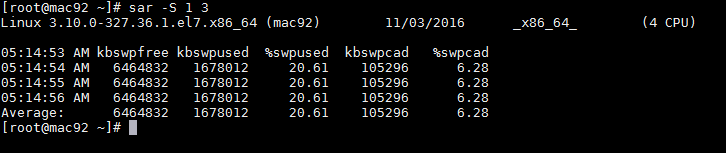


Following are few variations:

* sar -r
* sar -r 1 3
* sar -r -f /var/log/sa/sa10

## Swap Space Used (sar -S)

This reports the swap statistics. Option - “1 3” reports for every 1 second a total of 3 times. If the “kbswpused” and “%swpused” are at 0, then your system is not swapping.

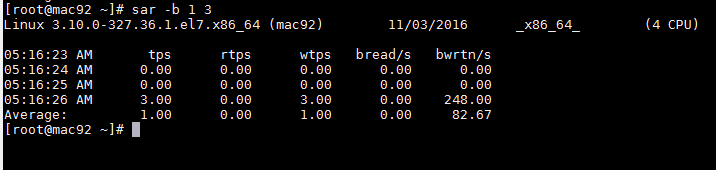


## Overall I/O Activities (sar -b)

This reports I/O statistics. Option - “1 3” reports for every 1 second a total of 3 times.

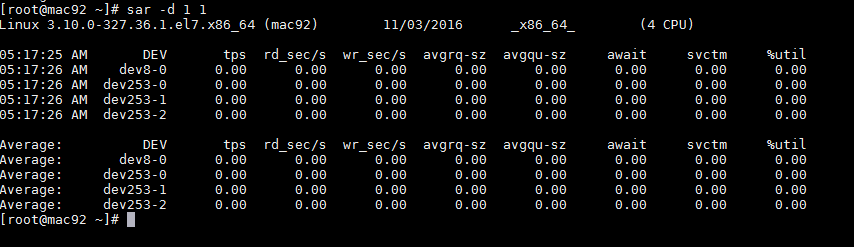
Following fields are displays in the example below.

* tps – Transactions per second (this includes both read and write)
* rtps – Read transactions per second
* wtps – Write transactions per second
* bread/s – Bytes read per second
* bwrtn/s – Bytes written per second



## Individual Block Device I/O Activities (sar -d)

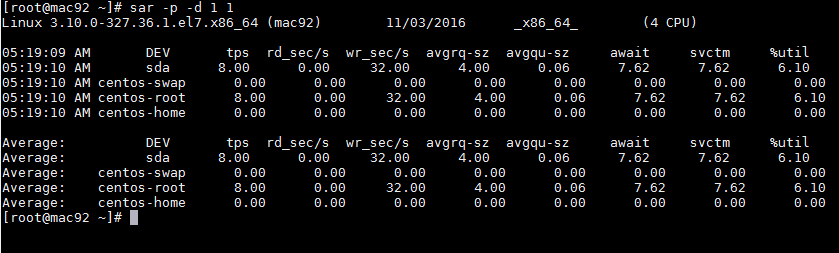
To identify the activities by the individual block devices (i.e a specific mount point, or LUN, or partition), use “sar -d”



In the above example “DEV” indicates the specific block device.

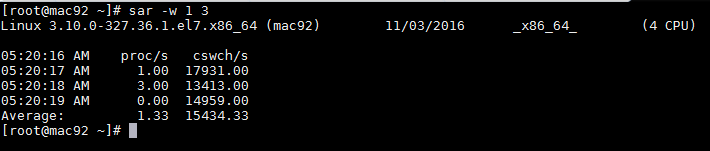
For example: “dev253-1” means a block device with 253 as major number, and 1 as minor number.

The device name (DEV column) can display the actual device name (for example: sda, sda1, sdb1 etc.,), if you use the -p option (pretty print) as shown below.



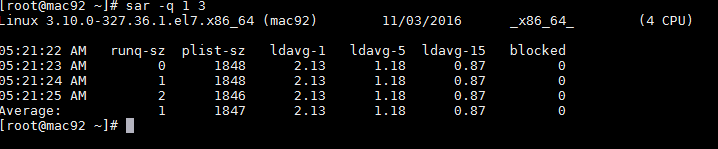
## Display context switch per second (sar -w)

This reports the total number of processes created per second and total number of context switches per second. Option - “1 3” reports for every 1 second a total of 3 times.



## Reports run queue and load average (sar -q)

This reports the run queue size and load average of last 1 minute, 5 minutes, and 15 minutes. Option - “1 3” reports for every 1 second a total of 3 times.



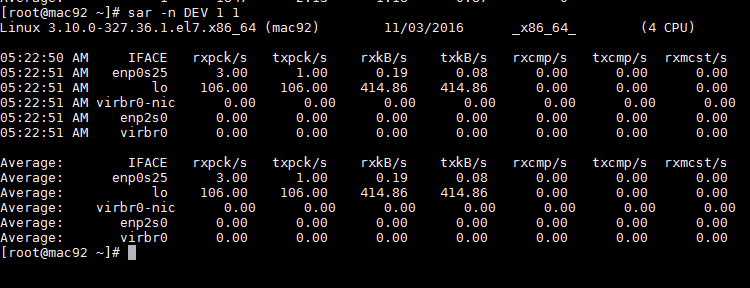
## Report network statistics (sar -n)

This reports various network statistics. For example: number of packets received (transmitted) through the network card, statistics of packet failure etc.,.Option - “1 3” reports for every 1 seconds a total of 3 times.

sar -n KEYWORD

KEYWORD can be one of the following:

* DEV – Displays network devices vital statistics for eth0, eth1, etc.,
* EDEV – Display network device failure statistics
* NFS – Displays NFS client activities
* NFSD – Displays NFS server activities
* SOCK – Displays sockets in use for IPv4
* IP – Displays IPv4 network traffic
* EIP – Displays IPv4 network errors
* ICMP – Displays ICMPv4 network traffic
* EICMP – Displays ICMPv4 network errors
* TCP – Displays TCPv4 network traffic
* ETCP – Displays TCPv4 network errors
* UDP – Displays UDPv4 network traffic
* SOCK6, IP6, EIP6, ICMP6, UDP6 are for IPv6
* ALL – This displays all of the above information. The output will be very long.



## Report Sar Data Using Start Time (sar -s)

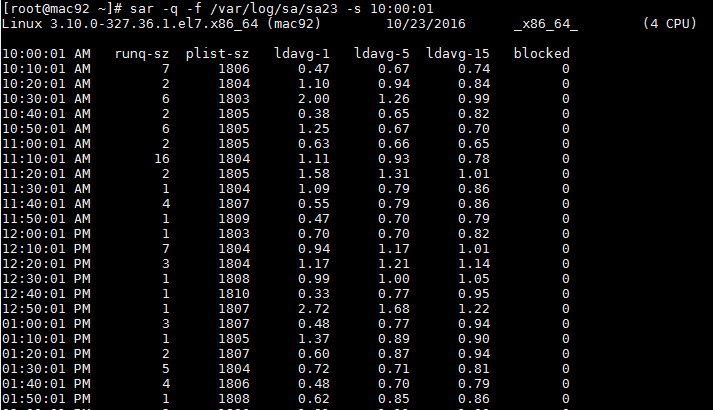
When you view historic sar data from the /var/log/sa/saXX file using “sar -f” option, it displays all the sar data for that specific day starting from 12:00 a.m for that day.

Using “-s hh:mi:ss” option, you can specify the start time. For example, if you specify “sar -s 10:00:00”, it will display the sar data starting from 10 a.m (instead of starting from midnight) as shown below.

You can combine -s option with other sar option.

For example, to report the load average on 26th of this month starting from 10 a.m in the morning, combine the -q and -s option as shown below.

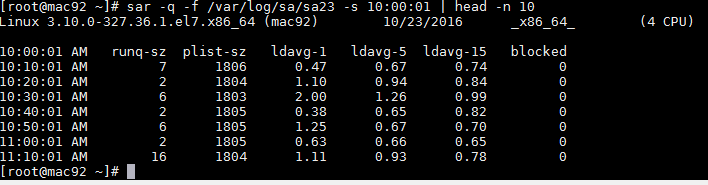
|  |
| --- |
| sar -q -f /var/log/sa/sa23 -s 10:00:01 |



There is no option to limit the end-time. You just have to get creative and use head command as shown below.

For example, starting from 10 a.m, if you want to see 7 entries, you have to pipe the above output to “head -n 10”.

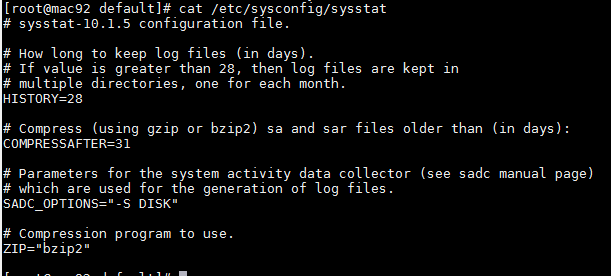
|  |
| --- |
| sar -q -f /var/log/sa/sa23 -s 10:00:01 | head -n 10 |



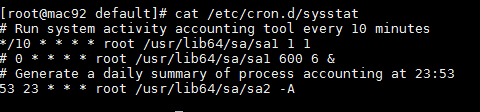
## Enable sar Logging

For Red Hat-based system sar logging is automatically enabled once it is installed, but on a Debian-based system (like Ubuntu), you might have to edit **/etc/default/sysstat**, and make sure that **ENABLED** is set to **true**

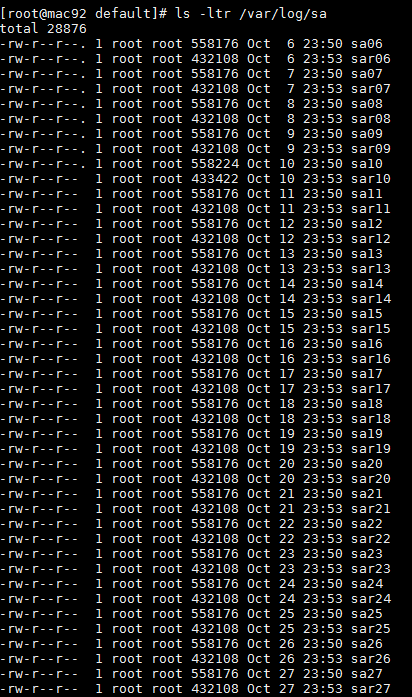
On a Red Hat-based system, sar will log 28 days of statistics by default. If you want to log more than that, you can edit /etc/sysconfig/sysstat and change the HISTORY option.



Once sysstat is configured and enabled, it will collect statistics about your system every ten minutes and store them in a logfile under either **/var/log/sysstat** or **/var/log/sa** via a cron job in **/etc/cron.d/sysstat**. There is also a daily cron job that will run right before midnight and rotate out the day's statistics. By default, the logfiles will be date-stamped with the current day of the month, so the logs will rotate automatically and overwrite the log from a month ago.



You can see data files under **var/log/sa** directory



# KSAR Graphical SAR Analysis Tool

sar provides CLI based output. The output may confuse all new users / sys admin. So you need to use kSar which is a java application that graph your sar data. It also permits to export data to PDF/JPG/PNG/CSV. You can load data from three methods: local file, local command execution, and remote command execution via SSH

## Download And Install kSar

You can download the KSar by executing below command

|  |
| --- |
| wget http://downloads.sourceforge.net/project/ksar/ksar/5.0.6/kSar-5.0.6.zip |

Unzip the zipped file

|  |
| --- |
| unzip kSar-5.0.6.zip |

Go to Ksar folder and give execute permission to run.sh file

|  |
| --- |
| chmod +x run.sh |

## Run KSAR

Create sar.data.ALL.txt file using below command

|  |
| --- |
| LC\_ALL=C sar -A > sar.data.ALL.txt |

By default sar uses 12 hours format with AM and PM this format not worked with KSAR so to change it in 24 hours format we need to use LC\_ALL=S option before the command.

We can set this in .bashrc file also so that we do not need to give it while running above command

|  |
| --- |
| vi ~/.bashrc  # .bashrc  # User specific aliases and functions  alias rm='rm -i'  alias cp='cp -i'  alias mv='mv -i'  alias sar="LANG=C sar"  # Source global definitions  if [ -f /etc/bashrc ]; then  . /etc/bashrc  fi  ~ |

We can set yellow heighted alias and reload the bashrc file using below command

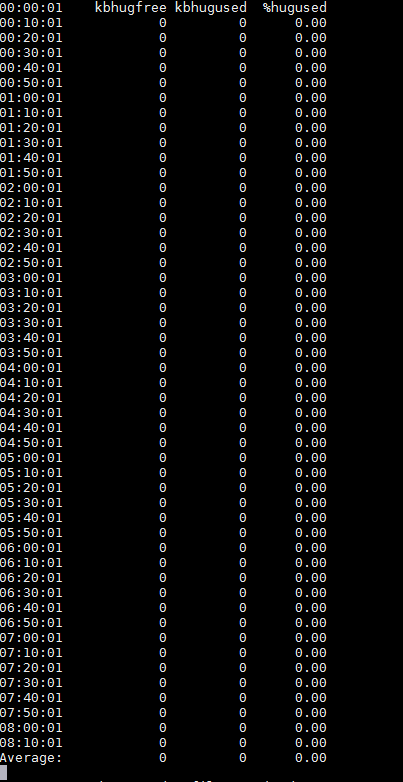
|  |
| --- |
| source ~/.bashrc  Or  . ~/.bashrc |

There is an issue with KSAR utility with sar version 10 to resolve this we need to do below step

Open sar.data.ALL.txt file and remove whole section that contains **kbhugfree kbhugused %hugused**

In vi editor you can press: /**kbhugfree** and press <enter>, dd to remove the whole section

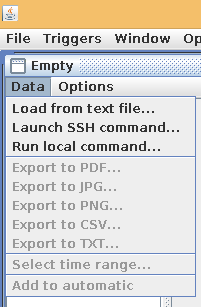
Remove below section till the last blank line



Execute run.sh file

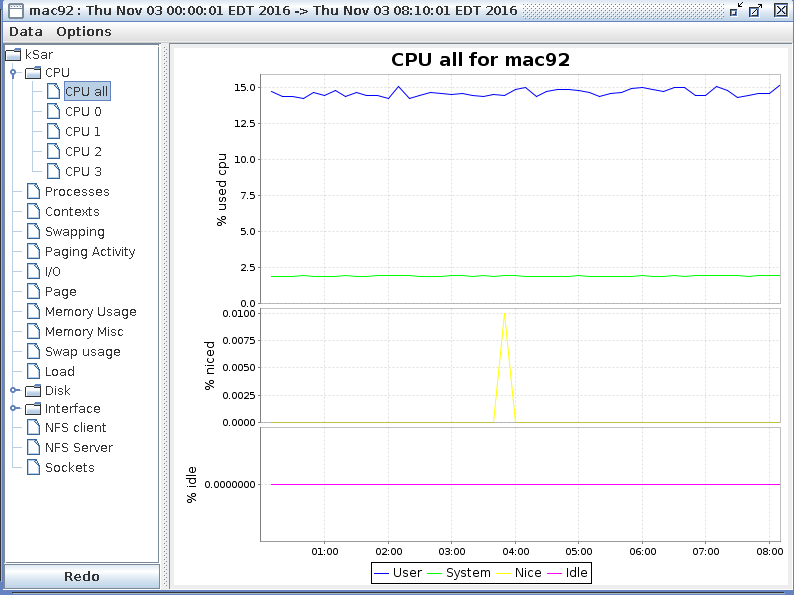
|  |
| --- |
| ./run.sh |

Click on load from text file



Open sar.data.ALL.txt file

You will see all graphs now



## Check Statistics for particular day

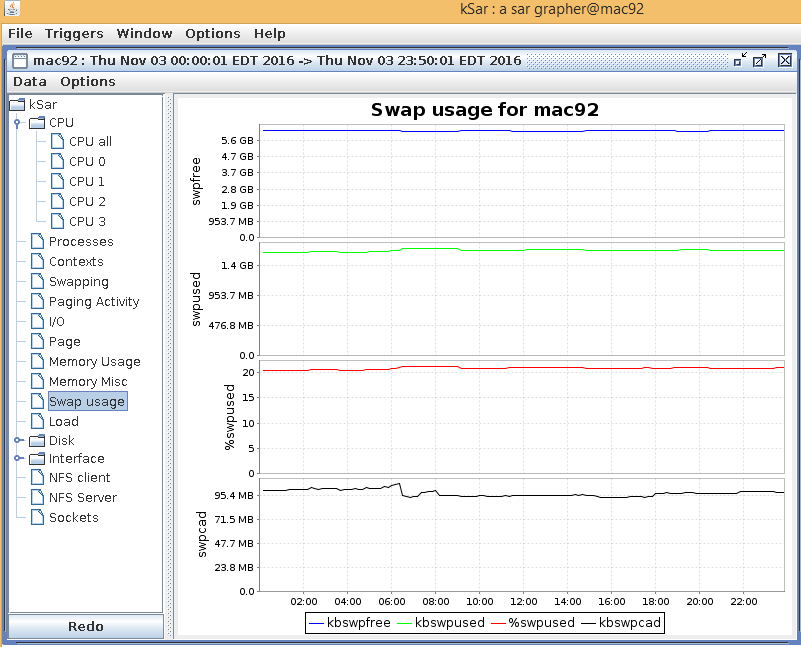
Let’s say if we want to check the performance of our system for a particular day, we need to check the sadd file located under **/var/log/sa** directory where dd denotes current day.

First convert that file into text format using below command, here we are using –A to get all statistics of the system, you can use other option also, -f is to convert this data file into text format

|  |
| --- |
| LC\_ALL=C sar -A -f /var/log/sa/sa03 > sa03.txt |

Now edit sa03.txt and remove whole section that contains **kbhugfree kbhugused %hugused** as described above

Now run Ksar and open that file



Now you can see all system statistics in graph format