

How to Display Linux Applications Using the Most CPU and Memory With a Bash Script?

written by sysadmin | 2 August 2025

I have some Linux servers used to run some of the necessary applications in the office. But occasionally, I can't reach certain servers due to issues, and I saw in the monitoring tool that the CPU or RAM on the server had increased significantly, resulting in the server having to be forcibly shut down and restarted. I tried to find out what applications caused the CPU or RAM to go up, but unfortunately, the monitoring application I used only showed the value of the CPU or RAM, but did not feature applications that used the CPU or RAM the most.

Problem

How to display Linux applications using the most CPU and memory with a bash script?

Solution

The `ps` command displays information about a selection of the active processes, so that by using the command, you can see which applications are using the most CPU and memory. Let's say you want to display the 50 most CPU-intensive applications on Linux, then use the command below:

```
ps -eo pid,%cpu,args --sort=--%cpu | head -n 10
```

And use the command below to display the 10 applications that use the most memory on Linux:

```
ps -eo pid,%mem,args --sort=--%mem | head -n 10
```

Based on the two commands above, you can create a bash script to see which applications are using the most CPU and RAM on Linux servers. Create a **mon_cpu_ram.sh** file and copy the bash script below into the file:

```
#!/bin/bash

# Get 50 application that take biggest cpu
ps -eo pid,%cpu,args --sort=-%cpu | head -n 50 > get_50_app_cpu.txt

# total cpu
awk 'NR > 1 {mem[$3 " "$4 " "$5 " "$6 " "$7 " "$8 " "$9 " "$10 " "$11 " "$12 " "$13 " "$14 " "$15 " "$16 " "$17] += $2} END {for (cmd in mem) printf "%.1f\t%s\n", mem[cmd], cmd}}' get_50_app_cpu.txt | sort -nr | head -n 5 > total_cpu.txt

# Display date and uptime
date +"%Y-%m-%d %H:%M:%S" >> cpu_history.txt
uptime=`uptime -p`
echo "Uptime: $uptime" >> app_cpu_mem.txt

# get 3 apps that takeing biggest CPU
head -n3 total_cpu.txt >> cpu_history.txt
echo >> cpu_history.txt

# send the output of biggest CPU
date +"%Y-%m-%d %H:%M:%S" >> app_cpu_mem.txt
echo "Big CPU" >> app_cpu_mem.txt
head -n3 total_cpu.txt >> app_cpu_mem.txt

# Get 50 application that take biggest memory
ps -eo pid,%mem,args --sort=-%mem | head -n 50 > get_50_app_mem.txt

# total memory
awk 'NR > 1 {mem[$3 " "$4 " "$5 " "$6 " "$7 " "$8 " "$9 " "$10 " "$11 " "$12 " "$13 " "$14 " "$15 " "$16 " "$17] += $2} END {for (cmd in mem) printf "%.1f\t%s\n", mem[cmd], cmd}}' get_50_app_mem.txt | sort -nr | head -n 5 > total_mem.txt

# get 3 apps that taking biggest memory
date +"%Y-%m-%d %H:%M:%S" >> mem_history.txt
head -n3 total_mem.txt >> mem_history.txt
echo >> mem_history.txt

# send the output of biggest Memory
echo "Big Memory" >> app_cpu_mem.txt
head -n3 total_mem.txt >> app_cpu_mem.txt

echo >> app_cpu_mem.txt
```

After that, permit the file and run it using the command below:

```
chmod +x mon_cpu_ram.sh
./mon_cpu_ram.sh
```

If you run the bash script, it will produce 3 files:

- `cpu_history.txt` => Display the history of the CPU only
- `mem_history.txt` => Display the history of the memory history
- `app_cpu_mem.txt` => Displays the CPU and Memory history simultaneously

```
sysadmin@LinuxMint:~/Documents/scripts/mon_cpu_ram$ chmod +x mon_cpu_ram.sh
sysadmin@LinuxMint:~/Documents/scripts/mon_cpu_ram$ ./mon_cpu_ram.sh
sysadmin@LinuxMint:~/Documents/scripts/mon_cpu_ram$ cat app_cpu_mem.txt
Uptime: up 13 hours, 6 minutes
2025-08-01 22:08:21
Big CPU
20,0  simplescreenrecorder --logfile
8,0  /usr/lib/firefox/firefox-bin -contentproc -isForBrowser -prefsHandle 0:42799 -prefMapHandle 1:272767 -jsInitHandle 2:242012 -parentBuildID 20250717180000 -sandboxReporter 3 -chrootClient 4
7,0  /usr/lib/firefox/firefox
Big Memory
6,0  /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=10097 --enable-crash-reporters,stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=4 --enable-main-frame-before-activation --renderer-client-id=269 --time-ticks-at-unix-epoch=1754013738023118 --running-vivaldi --launch-time-ticks=3489649215 --shared-files=v8_context_snapshot_data:100 --field-trial-handle=3,1,9663310222,679089897,10098509825225764213,262144 --variations-seed-version
5,0  /usr/lib/firefox/firefox-bin -contentproc -isForBrowser -prefsHandle 0:42799 -prefMapHandle 1:272767 -jsInitHandle 2:242012 -parentBuildID 20250717180000 -sandboxReporter 3 -chrootClient 4
2,0  /usr/lib/firefox/firefox --class WebApp-WhatsApp1662 --name WebApp-WhatsApp1662 --profile /home/sysadmin/.local/share/ice/firefox/WhatsApp1662 --no-remote http://web.whatsapp.com
sysadmin@LinuxMint:~/Documents/scripts/mon_cpu_ram$
```

Run the bash script

If you want the file to run every 5 minutes, enter the file in the crontab by copying the script below into the crontab:

```
*/5 * * * * cd /home/sysadmin/; . /mon_cpu_ram.sh
```

Note

I think this script is useful for sysadmins, especially me, to identify any applications that use the most CPU and RAM in Linux, so you can [limit CPU](#) or RAM to these applications.

References

man7.org

geeksforgeeks.org