

You have 1 free member-only story left this month. [Sign up for Medium and get an extra one](#)



Prashant Lakhera

Follow

Apr 7, 2019 · 4 min read ·  Member-only  Listen











100 Days of DevOps — Day 56-Debugging Performance Issue using SAR

Welcome to Day 56 of 100 Days of DevOps, Focus for today is Debugging Performance Issue using SAR

What is SAR?

SAR collect, report, or save system activity information. It's 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.

Installing SAR

```
# yum -y install sysstat
```

- To enable SAR on boot

```
# systemctl enable sysstat (Centos7/RHEL7)
```

How SAR works

- When we install a sysstat package it places a file in /etc/cron.d/sysstat

```
# cat /etc/cron.d/sysstat

# Run system activity accounting tool every 10 minutes

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

# 0 * * * * root /usr/lib64/sa/sa1 600 6 &

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

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

This file setup two cron jobs

- 1 job to record statistics every 10 minutes.
- 2 job to write the binary sa1#1# file to a text sar1#1# file once a day (typically right before midnight).

Additional config can be placed in a configuration file(/etc/sysconfig/sysstat)

```
# cat /etc/sysconfig/sysstat

# sysstat-10.1.5 configuration 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"
```

How SAR is useful

- Sar data is useful in pinpointing the system resource (networking, memory, IO, CPU, etc.) that is causing a performance issue.
- Sar data contains several useful sections for determining how the system was performing at a given time. By default, this is configured to run at ten minute intervals. Information on any of the descriptors used below (such as runq-sz, kbmempused, etc.) can be obtained by searching for these names after executing man sar.

Debugging High Load Average

- Load — information regarding how many tasks are currently on the system.

```
# sar -q 1 4

00:00:01      runq-sz  plist-sz    ldavg-1    ldavg-5    ldavg-15
[...]
04:30:01          1      2751         0.08         0.34         0.42
04:40:03          0       955         2.18         1.37         1.01
04:50:14         10      1645         8.18         6.58         3.83
Average:         25      1704        170.37       113.04       55.24
```

- From the above information, we can see that the load had a massive spike around 4:50. Typically a system's load should remain at 70% of the number of cores or lower. If the system's load is consistently above this amount there may be performance degradation, and if the load ever rises above the number of cores there will be a significant slowdown.

Debugging High Memory Utilization

- Memory — information regarding memory and swap usage.

```
# sar -r 1 3

00:00:01      kbmempfree kbmempused %memused kbbuffers  kbcached
kbswpfree kbswpused %swpused kbswpcad
[...]
04:30:01      62716272 201531148      76.27     15952     1556572
2048236      12      0.00          4
04:40:03      191904 264055516     99.93      2692      28908
0 2048248      100.00      8496
04:50:14      184100 264063320     99.93      1388      10600
0 2048248      100.00          0
Average:      4415719 259831701     98.33    1357749    20307185
1906978      141270      6.90      297
```

- Linux likes to use memory at around 99%; however, if the system is actively swapping then the system is most likely experiencing memory pressure. Considering that we see all of the swap used above within 10 minutes the system containing this data was experiencing memory issues during this time.

Debugging High I/O Utilization

- IO — information pertaining to the number of disk accesses.

```
# sar -b 1 3

00:00:01      tps      rtps      wtps    bread/s    bwrtn/s
[...]
04:30:01          67.95      51.84      16.12    4303.82    8664.14
04:40:03        564.60     227.07    337.52   34338.84   87719.02
04:50:14        51.05      40.25     10.80    1326.32    245.06
Average:        31.12     11.00     20.12    1383.64    3346.65
```

- The number of disk reads and writes will vary based on the underlying hardware; however, we can take a look at what is considered 'normal' for this system by examining the data over a period of time, and then look for spikes. We can see a large spike at 4:40 where the number of reads and writes increases dramatically. Note that shortly after these go back down, indicating that this massive burst was resolved.

Debugging High CPU Utilization

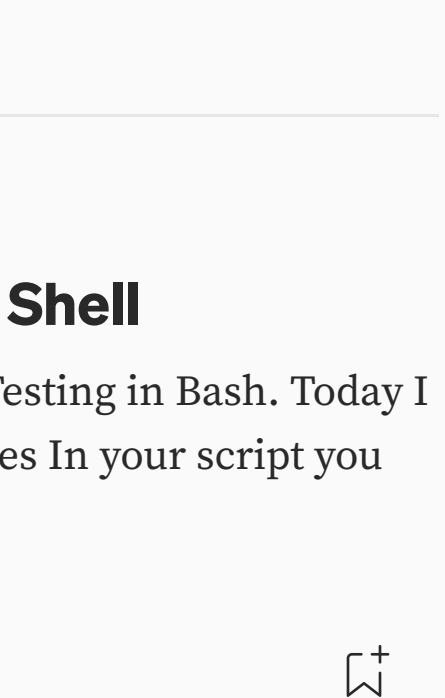
- CPU — information regarding where each of the system's cycles are spent.

```
# sar -u 1 10

00:00:01      CPU      %user      %nice      %system      %iowait
%steal      %idle
[...]
04:40:03      all     10.45         0.00         1.67         0.89
04:50:14      all     0.19         0.00        62.06         1.98
0.00      35.78
```

- Spending time in %user is expected behavior, as this is where all non-system tasks are accounted for. If cycles are actively being spent in %system then much of the execution time is being spent in lower-level code. If %iowait is high then it indicates processes are actively waiting due to disk accesses being a bottleneck on the system.
- In addition, sar data may be viewed graphically by downloading and using the 'ksar' tool. This tool is not provided from RedHat; however, it may be useful in pinpointing problematic times on the system. Documentation on this tool is at the following link

ksar : a sar grapher
Download ksar : a sar grapher for free. ksar is a sar graphing tool that can graph for now linux,mac and solaris sar...
sourceforge.net





Looking forward from you guys to join this journey and spend a minimum an hour every day for the next 100 days on DevOps work and post your progress using any of the below medium.

- Twitter: [@100daysofdevops](#) OR [@lakhera2015](#)
- Facebook: <https://www.facebook.com/groups/795382630808645/>
- Medium: <https://medium.com/@devopslearning>
- Slack: <https://devops-myworld.slack.com/messages/CF41EFG49/>
- GitHub Link: <https://github.com/100daysofdevops>

Reference

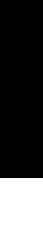
100 Days of DevOps — Day 0
D-day is just one day away and finally, this is a continuation of the post(I posted a month earlier)
medium.com

 1



Get an email whenever Prashant Lakhera publishes.

Your email

 Subscribe

By signing up, you will create a Medium account if you don't already have one. Review our [Privacy Policy](#) for more information about our privacy practices.

More from Prashant Lakhera

AWS Community Builder, Ex-Redhat, Author, Blogger, YouTuber, RHCA, RHCHS, RHCE, Docker Certified,4XAWS, CCNA, MCP, Certified Jenkins, Terraform Certified, 1XGCP

Apr 6, 2019 Member-only

100 Days of DevOps — Day 55-Introduction to YUM

Welcome to Day 55 of 100 Days of DevOps, Focus for today is Introduction to YUM What is YUM?

Yellowdog Updater Modified Resolve one of the major issues with rpm where we need to resolve all the package dependencies It acts as a frontend to rpm and handles all dependency...

Linux · 10 min read

Share your ideas with millions of readers. [Write on Medium](#)

Apr 5, 2019 Member-only

100 Days of DevOps — Day 54-And You Thought You Knew RPM

Welcome to Day 54 of 100 Days of DevOps, Focus for today is Introduction to Regular Expression Finally, I get a chance to write about this topic which I was planning to do for quite a long time.

The common trend I am seeing now a days (your opinion might...

Linux · 8 min read

Apr 4, 2019 Member-only

100 Days of DevOps — Day 53-Introduction to Regular Expression — Part 1

Welcome to Day 53 of 100 Days of DevOps, Focus for today is Introduction to Regular Expression What is a Regular Expression? It's a...

Regex · 4 min read

Apr 3, 2019 Member-only

100 Days of DevOps — Day 52-Conditional Testing in Shell

Welcome to Day 52 of 100 Days of DevOps, Focus for today is Conditional Testing in Bash. Today I am going to discuss test command test - check file types and compare values In your script you generally see, it starts with left bracket([) when need to perform...

Linux · 3 min read

Apr 2, 2019 Member-only

100 Days of DevOps — Day 51-Introduction to Bash Scripting

Welcome to Day 51 of 100 Days of DevOps, Focus for today is Bash Scripting. Wow, we have completed 50 days and in the last 50 days...

Linux · 4 min read

Love podcasts or audiobooks? Learn on the go with our new app. [Try Knowable](#)

Recommended from Medium

What's new in CSS?

Codesphere in Codesphere

Should You Really be Coding in Dark Mode?

Trending Pen

Is Windows 11 better than Windows 10?

What happens when you type google.com in your browser and press Enter

Configuring Haproxy and Updating its configuration file automatically via Ansible

How to Map a []Struct to a []interface in Go

Golem Beta V—Web Application Development toolset!

