Linux Performance Monitoring

Linux Performance Monitoring

- Linux system administrators should be proficient in Linux performance monitoring and tuning.
- This course session gives a high level overview on how we should approach performance monitoring and tuning in Linux.
- To identify system bottlenecks and come up with solutions to fix it, you should understand how various components of Linux works.
- For example, how to identify performance related issues like High CPU Load, High memory utilization, high disk io, high swap utilization and different tools & commands used to narrow down the issue etc.,

Linux Performance Monitoring

Course Contents

➢ Introduction about Performance monitoring in Linux

❑ Important four subsystems that needs to be monitored.

- CPU
- Memory
- I/O
- Network

Familiar & understanding with CPU Utilization

Like Context switches, Run Queue, CPU utilization & Load Average

Linux Performance Monitoring

Course Contents

- Important four subsystems that needs to be monitored.

  - CPU

Familiar & understanding with CPU Utilization

Like Context switches, Run Queue, CPU utilization & Load Average

Linux Performance Monitoring

Course Contents

- Familiar & understanding with Memory Utilization
- Familiar & understanding with I/O Analyses
- Familiar & understanding with Network Monitoring
- Familiar with various Commands to manage performance issues in Linux Servers
  - top
  - vmstat
  - iostat
  - free
  - lsof
  - tcpdump- Network Packet Analyzer

Linux Performance Monitoring

- Lab on SAR (System Activities Statistics)

- Following are the Linux performance statistics using sar command.

- Collective CPU usage
- Individual CPU statistics

- Memory used and available
- Swap space used and available

- Overall I/O activities of the system
- Individual device I/O activities

- Run queue and load average data
- Network statistics
- Report sar data from a specific time

Linux Performance Monitoring

- Lab on tcpdump- Network Packet Analyzer

  - For example: number of packets received (transmitted) through the network card, statistics of packet failure etc

- Lab on lsof - List open files

  - lsof command used in many Linux/Unix like system that is used to display list of all the open files and the processes.

Linux Performance Monitoring

- Linux system administrators should be proficient in Linux performance monitoring and tuning.
- This course session gives a high level overview on how we should approach performance monitoring and tuning in Linux.
- To identify system bottlenecks and come up with solutions to fix it, you should understand how various components of Linux works.
- For example, how to identify performance related issues like High CPU Load, High memory utilization, high disk io, high swap utilization and different tools & commands used to narrow down the issue etc.,

Linux Performance Monitoring

- On a very high level, following are the four subsystems that needs to be monitored.

  - CPU
  - Memory
  - I/O
  - Network

Linux Performance Monitoring

CPU

- You should understand the four critical performance metrics for CPU — context switch, run queue, cpu utilization, and load average.

Context Switch

- When CPU switches from one process (or thread) to another, it is called as context switch.
- However, a higher level of context switching can cause performance issues.

Linux Performance Monitoring

- CPU
- Context Switch

- Linux is a multitasking operating system. Which means the kernel has to switch between processes many times. Although it looks simple, the processor has to do multiple things while doing multitasking. For running multiple processes at the same time(which is very normal) the processor has to do the following things.

- Processor needs to save all context information of the current running process, before switching to another process execution. This is very necessary as the processor needs to again switch back to this process later.

- The processor has to fetch context information of the new process to process.

Linux Performance Monitoring

- CPU

- Context Switch

You can view information about your process's context switches in /proc/<pid>/status.

```
$ pid=307
$ grep ctxt /proc/$pid/status
voluntary_ctxt_switches:        41
nonvoluntary_ctxt_switches:     16
```

To see these numbers updating continuously, run

```
$ # Update twice a second.
$ watch -n.5 grep ctxt /proc/$pid/status
```

Linux Performance Monitoring

- CPU

- Run Queue

- Run queue indicates the total number of active processes in the current queue for CPU.
- When CPU is ready to execute a process, it picks it up from the run queue based on the priority of the process.
- Please note that processes that are in sleep state, or i/o wait state are not in the run queue.
- So, a higher number of processes in the run queue can cause performance issues.

Linux Performance Monitoring

- CPU

- Cpu Utilization

- This indicates how much of the CPU is currently getting used.
- This is fairly straight forward, and you can view the CPU utilization from the top command.
- 100% CPU utilization means the system is fully loaded.
- So, a higher %age of CPU utilization will cause performance issues.

Linux Performance Monitoring

- CPU

- Load Average

- This indicates the average CPU load over a specific time period.
- On Linux, load average is displayed for the last 1 minute, 5 minutes, and 15 minutes. This is helpful to see whether the overall load on the system is going up or down.
- For example, a load average of “0.75 1.70 2.10” indicates that the load on the system is coming down. 0.75 is the load average in the last 1 minute. 1.70 is the load average in the last 5 minutes. 2.10 is the load average in the last 15 minutes.
- Please note that this load average is calculated by combining both the total number of process in the queue, and the total number of processes in the uninterruptable task status.

Linux Performance Monitoring

➢ Memory

- As you know, RAM is your physical memory. If you have 4GB RAM installed on your system, you have 4GB of physical memory.
- Virtual memory = Swap space available on the disk + Physical memory. The virtual memory contains both user space and kernel space.
- Using either 32-bit or 64-bit system makes a big difference in determining how much memory a process can utilize.
- On a 32-bit system a process can only access a maximum of 4GB virtual memory. On a 64-bit system there is no such limitation.

Linux Performance Monitoring

➢ Swap

- Swap space in Linux is used when the amount of physical memory (RAM) is full. If the system needs more memory resources and the RAM is full, inactive pages in memory are moved to the swap space. While swap space can help machines with a small amount of RAM, it should not be considered a replacement for more RAM. Swap space is located on hard drives, which have a slower access time than physical memory.
- Swap space can be a dedicated swap partition (recommended), a swap file, or a combination of swap partitions and swap files.

Linux Performance Monitoring

➢I/O

- I/O wait is the amount of time CPU is waiting for I/O. If you see consistent high i/o wait on you system, it indicates a problem in the disk subsystem.
- You should also monitor reads/second, and writes/second. This is measured in blocks. i.e number of blocks read/write per second. These are also referred as bi and bo (block in and block out).
- tps indicates total transactions per seconds, which is sum of rtps (read transactions per second) and wtps (write transactions per seconds).

Linux Performance Monitoring

Network

- A good understanding of TCP/IP concepts is helpful while analyzing any network issues and packet loss using tcpdump utility.

- For network interfaces, you should monitor total number of packets (and bytes) received/sent through the interface, number of packets dropped, etc.

Linux Performance Monitoring

- Commands to manage performance issues in Linux Servers
- Managing performance on Linux systems is a lot easier with a few commands.
- Listed below are some of commands including top, vmstat, iostat, free, and sar. They may help in resolving performance issues quickly and easily.

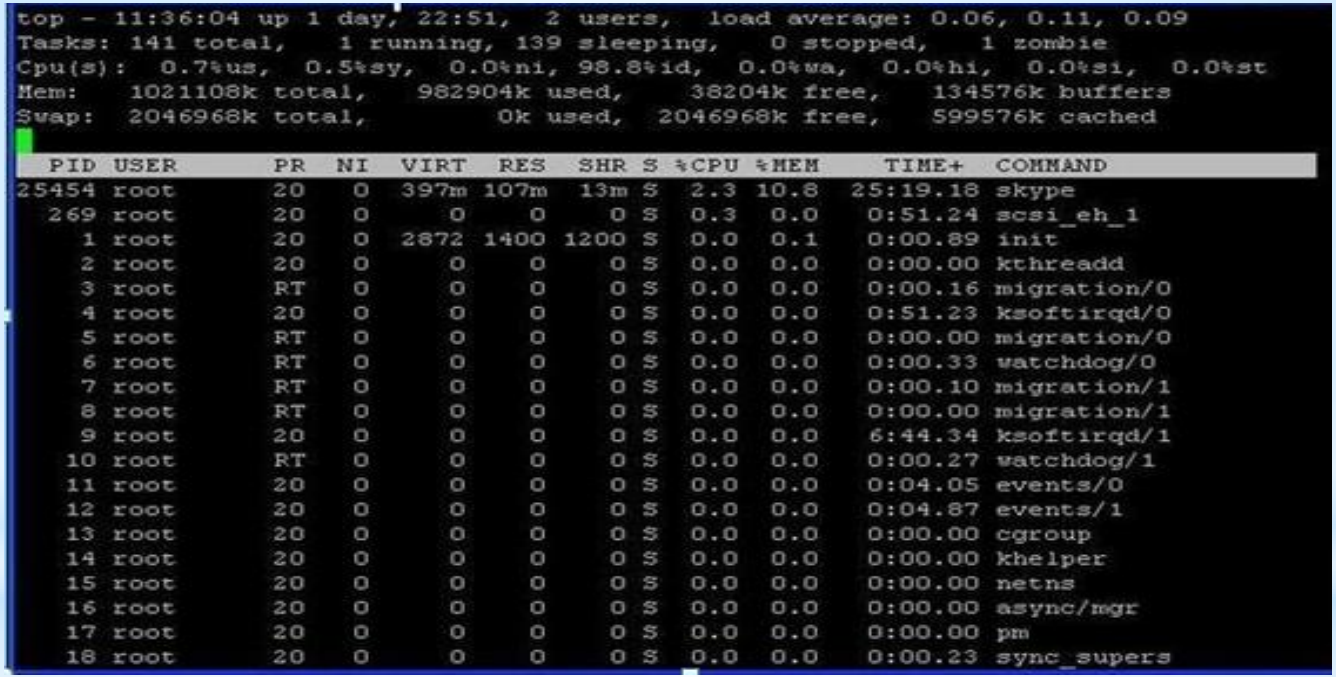
Linux Performance Monitoring

➢ Commands to manage performance issues in Linux Servers

- **Top**

- Linux Top command is a performance monitoring program which is used frequently by many system administrators to monitor Linux performance and it is available under many Linux/Unix like operating systems. The top command used to display all the running and active real-time processes in ordered list and updates it regularly. It display CPU usage, Memory usage, Swap Memory, Cache Size, Buffer Size, Process PID, User, Commands and much more. It also shows high memory and cpu utilization of a running processes. The top command is much useful for system administrator to monitor and take correct action when required. Let's see top command in action.

Linux Performance Monitoring



```
top - 11:36:04 up 1 day, 22:51,  2 users,  load average: 0.06, 0.11, 0.09
Tasks: 141 total,   1 running, 139 sleeping,   0 stopped,   1 zombie
Cpu(s):  0.7%us,  0.5%sy,  0.0%ni, 98.8%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   1021108k total,   982904k used,    38204k free,   134576k buffers
Swap:  2046968k total,        0k used,  2046968k free,   599576k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
25454 root      20   0  397m 107m  13m S  2.3 10.8  25:19.18 skype
  269 root      20   0     0    0    0 S  0.3  0.0   0:51.24 scsi_eh_1
    1 root      20   0  2872 1400 1200 S  0.0  0.1   0:00.89 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kthreadd
    3 root      RT   0     0    0    0 S  0.0  0.0   0:00.16 migration/0
    4 root      20   0     0    0    0 S  0.0  0.0   0:51.23 ksoftirqd/0
    5 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 migration/0
    6 root      RT   0     0    0    0 S  0.0  0.0   0:00.33 watchdog/0
    7 root      RT   0     0    0    0 S  0.0  0.0   0:00.10 migration/1
    8 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 migration/1
    9 root      20   0     0    0    0 S  0.0  0.0   6:44.34 ksoftirqd/1
   10 root      RT   0     0    0    0 S  0.0  0.0   0:00.27 watchdog/1
   11 root      20   0     0    0    0 S  0.0  0.0   0:04.05 events/0
   12 root      20   0     0    0    0 S  0.0  0.0   0:04.87 events/1
   13 root      20   0     0    0    0 S  0.0  0.0   0:00.00 cgroup
   14 root      20   0     0    0    0 S  0.0  0.0   0:00.00 khelper
   15 root      20   0     0    0    0 S  0.0  0.0   0:00.00 netns
   16 root      20   0     0    0    0 S  0.0  0.0   0:00.00 async/mgr
   17 root      20   0     0    0    0 S  0.0  0.0   0:00.00 pm
   18 root      20   0     0    0    0 S  0.0  0.0   0:00.23 sync_supers
```

Linux Performance Monitoring

- Commands to manage performance issues in Linux Servers

- **vmstat**

- The ‘vmstat’ command gives a snapshot of current CPU, IO, processes and memory usage. Similar to the top command, it dynamically updates and can be executed with this command:

- $ vmstat 10

- # vmstat

```
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu-----
 r  b      swpd   free      inact active      si   so    bi    bo   in   cs us sy id wa st
 1  0      0      810420  97380  70628     0    0     115   4    89   79   1  6 90  3  0
```

Linux Performance Monitoring

- Commands to manage performance issues in Linux Servers

- **sar**

- Use the ‘sar’ command line tool to collect, view and record performance data. This command is considerably more sophisticated than all the commands discussed above. It can collect and display data over longer periods.

Linux Performance Monitoring

- Commands to manage performance issues in Linux Servers

- **iostat**

- The ‘iostat’ command offers three reports. These are CPU utilization, device utilization, and network file system utilization. In case of running the command without options,it will display all three reports. The individual reports can be specified with the -c, -d and -h switches respectively.

Linux Performance Monitoring

- Commands to manage performance issues in Linux Servers

Iostat

- To identify whether I/O is causing system slowness you can use several commands but the easiest is the unix command top.

```
# top
top - 14:31:20 up 35 min, 4 users, load average: 2.25, 1.74, 1.68
Tasks: 71 total, 1 running, 70 sleeping, 0 stopped, 0 zombie
Cpu(s): 2.3%us, 1.7%sy, 0.0%ni, 0.0%id, 96.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 245440k total, 241004k used, 4436k free, 496k buffers
Swap: 409596k total, 5436k used, 404160k free, 182812k cached
```

From the CPU(s) line you can see the current percentage of CPU in I/O Wait; The higher the number the more cpu resources are waiting for I/O access.

```
wa -- iowait
 Amount of time the CPU has been waiting for I/O to complete.
```

Linux Performance Monitoring

Finding which disk is being written to

The above top command shows I/O Wait from the system as a whole but it does not tell you what disk is being affected; for this we will use the `iostat` command.

```
$ iostat -x 2 5
avg-cpu: %user %nice %system %iowait %steal %idle
 3.66 0.00 47.64 48.69 0.00 0.00

Device: rrqm/s wrqm/s r/s w/s rkB/s wkB/s avgrq-sz avgqu-sz await r_await w_await svctm %util
sda 44.50 39.27 117.28 29.32 11220.94 13126.70 332.17 65.77 462.79 9.80 2274.71 7.60 111.41
dm-0 0.00 0.00 83.25 9.95 10515.18 4295.29 317.84 57.01 648.54 16.73 5935.79 11.48 107.02
dm-1 0.00 0.00 57.07 40.84 228.27 163.35 8.00 93.84 979.61 13.94 2329.08 10.93 107.02
```

The iostat command in the example will print a report every 2 seconds for 5 intervals; the `-x` tells iostat to print out an extended report.

Linux Performance Monitoring

- Commands to manage performance issues in Linux Servers

- **free**

- The ‘free’ command shows memory statistics for both main memory and swap. A total memory amount can be displayed by specifying the -t switch. The amounts in bytes can also be displayed by specifying the -b switch and megabytes using the -m switch (it displays in kilobytes by default).
- Free can also be run continuously using the -s switch with a delay specified in seconds:

- $ free -s 5

Linux Performance Monitoring

- Commands to manage performance issues in Linux Servers

- free

```
[root@localhost ~]# free
              total        used        free      shared  buff/cache   available
Mem:        1868664      368060     1099712        9320      400892     1298444
Swap:       1583100           0     1583100
```

Linux Performance Monitoring

- Commands to manage performance issues in Linux Servers

total Total installed memory (MemTotal and SwapTotal in /proc/meminfo)

used Used memory (calculated as **total** - **free** - **buffers** - **cache**)

free Unused memory (MemFree and SwapFree in /proc/meminfo)

shared Memory used (mostly) by tmpfs (Shmem in /proc/meminfo, available on kernels 2.6.32, displayed as zero if not available)

buffers
Memory used by kernel buffers (Buffers in /proc/meminfo)

cache Memory used by the page cache and slabs (Cached and Slab in /proc/meminfo)

buff/cache
Sum of **buffers** and **cache**

available
Estimation of how much memory is available for starting new applications, without swapping. Unlike the data provided by the **cache** or **free** fields, this field takes into account page cache and also that not all reclaimable memory slabs will be reclaimed due to items being in use (MemAvailable in /proc/meminfo, available on kernels 3.14, emulated on kernels 2.6.27+, otherwise the same as **free**)

Linux Performance Monitoring

➢Commands to manage performance issues in Linux Servers

- **Lsof – List Open Files**

- Lsof command used in many Linux/Unix like system that is used to display list of all the open files and the processes. The open files included are disk files, network sockets, pipes, devices and processes. One of the main reason for using this command is when a disk cannot be unmounted and displays the error that files are being used or opened. With this commmand you can easily identify which files are in use. The most common format for this command is.
- $ lsof

Linux Performance Monitoring

- Commands to manage performance issues in Linux Servers

- **Tcpdump – Network Packet Analyzer**

- Tcpdump one of the most widely used command-line network packet analyzer or packets sniffer program that is used capture or filter TCP/IP packets that received or transferred on a specific interface over a network. It also provides a option to save captured packages in a file for later analysis. tcpdump is almost available in all major Linux distributions.

Linux Performance Monitoring

- SAR (System Activities Statistics)

- Following are the Linux performance statistics using sar.

  - Collective CPU usage
  - Individual CPU statistics
  - Memory used and available
  - Swap space used and available
  - Overall I/O activities of the system
  - Individual device I/O activities
  - Run queue and load average data
  - Network statistics
  - Report sar data from a specific time

Linux Performance Monitoring

- SAR (System Activities Statistics)

- Using sar you can monitor performance of various Linux subsystems (CPU, Memory, I/O, Network Statistics) in real time.

- Using sar, you can also collect all performance data on an on-going basis, store them, and do historical analysis to identify bottlenecks.

Linux Performance Monitoring

- SAR (System Activities Statistics)

- 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

Linux Performance Monitoring

- SAR (System Activities Statistics)
- Install Sysstat Package
- Once installed, verify the sar version using “sar -V”. Version 10 is the current stable version of sysstat.
- $ sar -V

Linux Performance Monitoring

- SAR (System Activities Statistics)

1. CPU Usage of ALL CPUs (sar -u)

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

Linux Performance Monitoring

- SAR (System Activities Statistics)

1. CPU Usage of ALL CPUs (sar -u)

```
LinuxGuru@Server#sar -u 1 2
Linux 2.6.18-404.el5                                    04/09/17

10:53:47        CPU    %user    %nice  %system  %iowait  %steal    %idle
10:53:48        all        2.04     0.00     2.04     0.00     0.00    95.92
10:53:49        all     0.00     0.00     0.00     0.00     0.00   100.00
Average:        all     1.02     0.00     1.02     0.00     0.00    97.97
```

Linux Performance Monitoring

❖SAR (System Activities Statistics)

1. CPU Usage of ALL CPUs (sar -u)

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.

Linux Performance Monitoring

- SAR (System Activities Statistics)

2. 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.

Linux Performance Monitoring

- SAR (System Activities Statistics)
- 2. CPU Usage of Individual CPU or Core (sar -P)

```
LinuxGuru@Server#sar -P ALL 1 1
Linux 2.6.18-404.el5                                        04/09/17

10:38:19        CPU    %user    %nice   %system   %iowait    %steal    %idle
10:38:20        all     0.00     0.00      0.00      0.00      0.00   100.00
10:38:20          0     0.00     0.00      0.00      0.00      0.00   100.00

Average:        CPU    %user    %nice   %system   %iowait    %steal    %idle
Average:        all     0.00     0.00      0.00      0.00      0.00   100.00
Average:          0     0.00     0.00      0.00      0.00      0.00   100.00
```

Linux Performance Monitoring

- SAR (System Activities Statistics)

3. Memory Free and Used (sar -r)

```
LinuxGuru@Server#sar -r 1 3
Linux 2.6.18-404.el5                                    04/09/17

10:38:58    kbmemfree kbmemused  %memused kbbuffers  kbcached kbswpfree kbswpused  %swpused  kbswpcad
10:38:59        45148   3998636     98.88    524492   2997728   5996424       112      0.00         0
10:39:00        45148   3998636     98.88    524492   2997728   5996424       112      0.00         0
10:39:01        45212   3998572     98.88    524500   2997720   5996424       112      0.00         0
Average:        45169   3998615     98.88    524495   2997725   5996424       112      0.00         0
LinuxGuru@Server#
```

Linux Performance Monitoring

- SAR (System Activities Statistics)

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.

Linux Performance Monitoring

❖SAR (System Activities Statistics)

3. Memory Free and Used (sar -r)

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

Linux Performance Monitoring

- SAR (System Activities Statistics)

3. Memory Free and Used (sar -r)

Following are few variations:

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

Linux Performance Monitoring

- SAR (System Activities Statistics)

4. Overall I/O Activities (sar -b)

This reports I/O statistics. “1 3” reports for every 1 seconds 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

Linux Performance Monitoring

- SAR (System Activities Statistics)

```
LinuxGuru@Server#sar -b 1 3
Linux 2.6.18-404.el5                                    04/09/17

10:40:03       tps      rtps      wtps   bread/s   bwrtn/s
10:40:04      0.00      0.00      0.00      0.00      0.00
10:40:05      0.00      0.00      0.00      0.00      0.00
10:40:06     98.99      0.00     98.99      0.00   2133.33
Average:     32.78      0.00     32.78      0.00    706.35
```

Linux Performance Monitoring

- SAR (System Activities Statistics)

Following are few variations:

sar -b

sar -b 1 3

sar -b -f /var/log/sa/sa10

Note: Use “sar -v” to display number of inode handlers, file handlers, and pseudo-terminals used by the system.

Linux Performance Monitoring

- SAR (System Activities Statistics)

5. 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”

Linux Performance Monitoring

- SAR (System Activities Statistics)

```
LinuxGuru@Server#sar -d 1 1
Linux 2.6.18-404.el5                                        04/09/17

10:41:07       DEV       tps  rd_sec/s  wr_sec/s  avgrq-sz  avgqu-sz     await     svctm     %util
10:41:08       dev8-0    2.00      0.00    176.00     88.00      0.00      1.00      1.00      0.20
10:41:08       dev8-1    0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
10:41:08       dev8-2    2.00      0.00    176.00     88.00      0.00      1.00      1.00      0.20
10:41:08       dev8-16   0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
10:41:08       dev8-17   0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.20
```

Linux Performance Monitoring

- SAR (System Activities Statistics)

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

For example: “dev8-1” means a block device with 8 as major number, and 1 as minor number.

Linux Performance Monitoring

SAR (System Activities Statistics)

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.

```
LinuxGuru@Server#sar -p -d 1 1
Linux 2.6.18-404.el5                                    04/09/17

10:42:18        DEV       tps  rd_sec/s  wr_sec/s  avgrq-sz  avgqu-sz     await     svctm     %util
10:42:19        sda          0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
10:42:19        sda1         0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
10:42:19        sda2         0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
10:42:19        sdb          0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
10:42:19        sdb1         0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
10:42:19        sdc          0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
```

Linux Performance Monitoring

- SAR (System Activities Statistics)

Following are few variations:

sar -d

sar -d 1 3

sar -d -f /var/log/sa/sa10

sar -p -d

Linux Performance Monitoring

- SAR (System Activities Statistics)

6. 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. “1 3” reports for every 1 seconds a total of 3 times.

Linux Performance Monitoring and Tuning Introduction

- SAR (System Activities Statistics)

```
LinuxGuru@Server#sar -q 1 3
Linux 2.6.18-404.el5                                    04/09/17

10:42:56      runq-sz  plist-sz   ldavg-1   ldavg-5  ldavg-15
10:42:57            0       296      0.00      0.00      0.00
10:42:58            0       296      0.00      0.00      0.00
10:42:59            0       296      0.00      0.00      0.00
Average:            0       296      0.00      0.00      0.00
LinuxGuru@Server#
```

Linux Performance Monitoring and Tuning Introduction

- SAR (System Activities Statistics)

Note: The “blocked” column displays the number of tasks that are currently blocked and waiting for I/O operation to complete.

Following are few variations:

sar -q

sar -q 1 3

sar -q -f /var/log/sa/sa10

Linux Performance Monitoring and Tuning Introduction

- SAR (System Activities Statistics)

7. 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.,. “1 3” reports for every 1 seconds a total of 3 times.

Linux Performance Monitoring and Tuning Introduction

- SAR (System Activities Statistics)

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
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.

Linux Performance Monitoring and Tuning Introduction

- SAR (System Activities Statistics)

$ sar -n DEV 1 1

```
LinuxGuru@Server#sar -n DEV 1 1
Linux 2.6.18-404.el5                                                    04/09/17

10:45:26      IFACE   rxpck/s   txpck/s   rxbyt/s   txbyt/s   rxcmp/s   txcmp/s  rxmcst/s
10:45:27         lo      0.00      0.00      0.00      0.00      0.00      0.00      0.00
10:45:27       eth0      7.07      4.04   1245.45    363.64      0.00      0.00      0.00
10:45:27       sit0      0.00      0.00      0.00      0.00      0.00      0.00      0.00

Average:      IFACE   rxpck/s   txpck/s   rxbyt/s   txbyt/s   rxcmp/s   txcmp/s  rxmcst/s
Average:         lo      0.00      0.00      0.00      0.00      0.00      0.00      0.00
Average:       eth0      7.07      4.04   1245.45    363.64      0.00      0.00      0.00
Average:       sit0      0.00      0.00      0.00      0.00      0.00      0.00      0.00
```

Linux Performance Monitoring and Tuning Introduction

- SAR (System Activities Statistics)

8. 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.

Linux Performance Monitoring and Tuning Introduction

- SAR (System Activities Statistics)

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
Linux 2.6.18-194.el5PAE (dev-db)        03/26/2011      _i686_  (8 CPU)

10:00:01 AM   runq-sz  plist-sz   ldavg-1   ldavg-5  ldavg-15   blocked
10:10:01 AM         0       127      2.00      3.00       5.00         0
10:20:01 AM         0       127      2.00      3.00       5.00         0
...
11:20:01 AM         0       127      5.00      3.00       3.00         0
12:00:01 PM         0       127      4.00      2.00       1.00         0
```

Linux Performance Monitoring and Tuning Introduction

- SAR (System Activities Statistics)

- Sample Performance Issue reported

CPU Utilization:

```
# sar -f /var/log/sa/sa11 -u 2 -s 06:30:00 -e 07:30:00
Linux 2.6.32-431.20.3.el6.s390x 11/10/16 _s390x_ (1 CPU)

06:30:01 CPU %user %nice %system %iowait %steal %idle
06:40:01 all 28.61 0.02 3.13 0.09 5.05 63.12
06:50:01 all 30.17 0.00 3.26 0.04 5.43 61.10
07:00:01 all 31.32 0.00 6.91 18.85 5.21 37.71 <== iowait is little bit high
07:10:01 all 40.83 0.00 13.10 13.98 5.56 26.53 <== iowait is little bit high
07:20:01 all 34.33 0.00 3.63 0.09 7.13 54.82
Average: all 33.05 0.00 6.01 6.61 5.67 48.66
```

Linux Performance Monitoring and Tuning Introduction

- SAR (System Activities Statistics)
- Sample Performance Issue reported

```
# sar -f  /var/log/sa/sa11 -r 2 -s 05:30:00  -e 07:00:00
Linux 2.6.32-431.20.3.el6.s390x  11/10/16        _s390x_ (1 CPU)

05:30:01    kbmemfree kbmemused  %memused kbbuffers  kbcached  kbcommit   %commit
05:40:01       576056   3534540     85.99    288032   2406268   2959556     29.97
05:50:01       579432   3531164     85.90    288640   2402832   2958840     29.97
06:00:01       570692   3539904     86.12    289300   2403720   2974996     30.13
06:10:01       576428   3534168     85.98    289968   2404044   2959556     29.97
06:20:01       576688   3533908     85.97    290568   2403600   2958840     29.97
06:30:01       575272   3535324     86.01    291212   2402080   2970632     30.09
06:40:01       577636   3532960     85.95    291856   2400944   2959556     29.97
06:50:01       577796   3532800     85.94    292536   2400664   2958840     29.97
```

Linux Performance Monitoring and Tuning Introduction

- SAR (System Activities Statistics)

- Sample Performance Issue reported

Disk Utilization:

==========

```
[root@nc2z01lx012 ~]# sar -f /var/log/sa/sa11 -p -d 1 -s 06:30:00 -e 07:30:00 | grep -i swap
06:30:01        DEV                       tps                rd_sec/s  wr_sec/s  avgrq-sz  avgqu-sz  await     svctm     %util

06:40:01    vg_root-lv_swap     0.00         0.00     0.00     0.00     0.00     0.00         0.00         0.00
06:50:01    vg_root-lv_swap     0.00         0.00     0.00     0.00     0.00     0.00         0.00         0.00
07:00:01    vg_root-lv_swap     0.12                              0.75     0.25     8.00         0.00        15.33        2.40         0.03
07:10:01    vg_root-lv_swap     0.12         0.21     0.72     8.00     0.00    17.29         6.57         0.08
07:20:01    vg_root-lv_swap     0.00         0.00     0.01     8.00     0.00    10.00        10.00         0.00
Average:    vg_root-lv_swap     0.05         0.19     0.20     8.00     0.00    16.23         4.45         0.00
```

disk await is high during the same period of time and the disk is swap disk. It is trying to access the swap disk but unable to get it. So the swap utilization is normal but unable to get the swap disk to swapin swap out.

Linux Performance Monitoring

Tcpdump

- tcpdump is a most powerful and widely used command-line packets sniffer or package analyzer tool which is used to capture or filter TCP/IP packets that received or transferred over a network on a specific interface.

- It is available under most of the Linux/Unix based operating systems. tcpdump also gives us a option to save captured packets in a file for future analysis.

Linux Performance Monitoring

- Tcpdump

1. Capture Packets from Specific Interface

```
# tcpdump -i eth0
```

2. Capture Only N Number of Packets

When you run tcpdump command it will capture all the packets for specified interface, until you Hit cancel button. But using -c option, you can capture specified number of packets. The below example will only capture 6 packets.

```
# tcpdump -c 5 -i eth0
```

Linux Performance Monitoring

- Tcpdump

3. Print Captured Packets in ASCII

The below tcpdump command with option -A displays the package in ASCII format. It is a character-encoding scheme format.

```
# tcpdump -A -i eth0
```

4. Display Available Interfaces

To list number of available interfaces on the system, run the following command with -D option.

```
# tcpdump -D
1.eth0
2.eth1
```

Linux Performance Monitoring

- Tcpdump

5. Capture and Save Packets in a File

As we said, that tcpdump has a feature to capture and save the file in a .pcap format, to do this just execute command with -w option.

```
# tcpdump -w 0001.pcap -i eth0
```

6. Read Captured Packets File

To read and analyze captured packet 0001.pcap file use the command with -r option, as shown below.

```
# tcpdump -r 0001.pcap
```

Linux Performance Monitoring

- Tcpdump

7. Capture IP address Packets

To capture packets for a specific interface, run the following command with option -n.

```
# tcpdump -n -i eth0
```

8. Capture only TCP Packets.

To capture packets based on TCP port, run the following command with option tcp.

```
# tcpdump -i eth0 tcp
```

Linux Performance Monitoring

- Tcpdump

9. Capture Packet from Specific Port

Let's say you want to capture packets for specific port 22, execute the below command by specifying port number 22 as shown below.

- # tcpdump -i eth0 port 22

10. To collect the packet details on eth0

```
# tcpdump -n -i eth0 -s 0 -w /tmp/ppte-esi-eth0.cap
Press Cntrl+C to stop
```

Linux Performance Monitoring

- Tcpdump

11. tcpdump on particular host IP:

```
tcpdump -i eth0 -s 0 host 10.165.107.73
```

Linux Performance Monitoring

- lsof

* It is easy to remember lsof command if you think of it as “ls + of”, where ls stands for list, and of stands for open files.
*
* It is a command line utility which is used to list the information about the files that are opened by various processes. In unix, everything is a file, (pipes, sockets, directories, devices, etc.). So by using lsof, you can get the information about any opened files.

Linux Performance Monitoring

✓ High CPU Utilization

- Below are commands which can be used to find out biggest cpu consuming processes

- top
- ps –eo pmem,pcpu,pid,args | tail –n +2|sort –rnk 1|head

Linux Performance Monitoring

✓ High Memory Utilization

- Below are commands which can be used to find out biggest memory consuming processes

- top
- ps –eo pmem,pcpu,pid,args | tail –n +2|sort –rnk 2|head

Linux Administration

➢Swap

- Swap space in Linux is used when the amount of physical memory (RAM) is full. If the system needs more memory resources and the RAM is full, inactive pages in memory are moved to the swap space. While swap space can help machines with a small amount of RAM, it should not be considered a replacement for more RAM. Swap space is located on hard drives, which have a slower access time than physical memory.
- Swap space can be a dedicated swap partition (recommended), a swap file, or a combination of swap partitions and swap files.

Linux Administration

➢How to Increase Swap in Linux

- There are two methods we can create the swap space

- Using swap partition
- Using swap file

Linux Administration

- How to Increase Swap in Linux

- There are two methods we can create the swap space

- Using swap partition
- Using swap file