**Wednesday, 25 July 2012**

**Linux System Resource & Performance Monitoring**

**Linux System Resource & Performance Monitoring:**

**Monitoring the Hard Disk Space**

Use a simple command like:

$ **df -h**

This results in the output:

Filesystem Size Used Avail Use% Mounted on

/dev/sda1 22G 5.0G 16G 24% /

/dev/sda2 34G 23G 9.1G 72% /home

This shows there are two partitions (1 & 2) of the hard disk sda, which are currently at 24% and 72% utilization. The total size is shown in gigabytes (G). How much is used and balance available is shown as well. However, checking each hard disk to see the percentage used can be a big drag. It is better that the system checks the disks and informs you by email if there is a potential danger. Bash scripts may be written for this and run at specific times as a cron job.

For the GUI, there is a graphical tool called ‘**Baobab**’ for checking the disk usage. It shows how a disk is being used and displays the information in the form of either multicolored concentric rings or boxes.

**Monitoring Memory Usage**

RAM or memory is used to run the current application. Under Linux, there are a number of ways you can check the used memory space -- both in static and dynamic conditions.

For a static snapshot of the memory, use ‘**free -m**’ which results in the output:

$ **free -m** total used free shared buffers cached

Mem: 1998 1896 101 0 59 605

-/+ buffers/cache: 1231 766

Swap: 290 77 213

Here, the total amount of RAM is depicted in megabytes (MB), along with cache and swap. A somewhat more detailed output can be obtained by the command ‘**vmstat**’:

root@gateway [~]# **vmstat**  
procs -----------memory------------- ---swap-- -----io---- --system-- -----cpu------  
r b swpd free buff cache si so bi bo in cs us sy id wa st  
1 0 0 767932 0 0 0 0 10 3 0 1 2 0 97 0 0  
root@gateway [~]#

However, if a dynamic situation of what is happening to the memory is to be examined, you have to use ‘**top**’ or ‘**htop**’. Both will give you a picture of which process is using what amount of memory and the picture will be updated periodically. Both ‘**top**’ and ‘**htop**’ will also show the CPU utilization, tasks running and their PID. Whereas ‘**top**’ has a purely numerical display, ‘**htop**’ is somewhat more colorful and has a semi-graphic look. There is also a list of command menus at the bottom for set up and specific operations.

root@gateway [~]# **top**  
  
top - 01:04:18 up 81 days, 11:05, 1 user, load average: 0.08, 0.28, 0.33  
Tasks: 47 total, 1 running, 45 sleeping, 0 stopped, 1 zombie  
Cpu(s): 2.4%us, 0.4%sy, 0.0%ni, 96.7%id, 0.5%wa, 0.0%hi, 0.0%si, 0.0%st  
Mem: 1048576k total, 261740k used, 786836k free, 0k buffers  
Swap: 0k total, 0k used, 0k free, 0k cached  
  
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND   
1 root 15 0 10372 736 624 S 0.0 0.1 1:41.86 init   
5407 root 18 0 12424 756 544 S 0.0 0.1 0:13.71 dovecot   
5408 root 15 0 19068 1144 892 S 0.0 0.1 0:12.09 dovecot-auth   
5416 dovecot 15 0 38480 2868 2008 S 0.0 0.3 0:10.80 pop3-login   
5417 dovecot 15 0 38468 2880 2008 S 0.0 0.3 0:49.31 pop3-login   
5418 dovecot 16 0 38336 2700 2020 S 0.0 0.3 0:01.15 imap-login   
5419 dovecot 15 0 38484 2856 2020 S 0.0 0.3 0:04.69 imap-login   
9745 root 18 0 71548 22m 1400 S 0.0 2.2 0:01.39 lfd   
11501 root 15 0 160m 67m 2824 S 0.0 6.6 1:32.51 spamd   
23935 firewall 18 0 15276 1180 980 S 0.0 0.1 0:00.00 imap   
23948 mailnull 15 0 64292 3300 2620 S 0.0 0.3 0:05.62 exim   
23993 root 15 0 141m 49m 2760 S 0.0 4.8 1:00.87 spamd   
24477 root 18 0 37480 6464 1372 S 0.0 0.6 0:04.17 queueprocd   
24494 root 18 0 44524 8028 2200 S 0.0 0.8 1:20.86 tailwatchd   
24526 root 19 0 92984 14m 1820 S 0.0 1.4 0:00.00 cpdavd   
24536 root 33 18 23892 2556 680 S 0.0 0.2 0:02.09 cpanellogd   
24543 root 18 0 87692 11m 1400 S 0.0 1.1 0:33.87 cpsrvd-ssl   
25952 named 22 0 349m 8052 2076 S 0.0 0.8 20:17.42 named   
26374 root 15 -4 12788 752 440 S 0.0 0.1 0:00.00 udevd   
28031 root 17 0 48696 8232 2380 S 0.0 0.8 0:00.07 leechprotect   
28038 root 18 0 71992 2172 132 S 0.0 0.2 0:00.00 httpd   
28524 root 18 0 90944 3304 2584 S 0.0 0.3 0:00.01 sshd

For a graphical display of how the memory is being utilized, the Gnome System Monitor gives a detailed picture. There are other system monitors available under various window managers in Linux.

**What is Your CPU Doing?**

You may have a single, a dual core, or a quad core CPU in your system. To see what each CPU is doing or how two CPUs are sharing the load, you have to use ‘**top**’ or ‘**htop**’. These command line applications show the percentage of each CPU being utilized. You can also see process statistics, memory utilization, uptime, load average, CPU status, process counts, and memory and swap space utilization statistics.

Similar output statistics may be seen by using command line tools such as the ‘**mpstat**’, which is part of a group package called ‘**sysstat**’. You may have to install ‘**sysstat**’ in your system, since it may not be installed by default. Once installed, you can monitor a variety of parameters, for example compare the CPU utilization of an SMP system or multi-processor system.

Finding out if any specific process is hogging the CPU needs a little more command line instruction such as:

$ **ps -eo pcpu,pid,user,args | sort -r -k1 | less**

OR

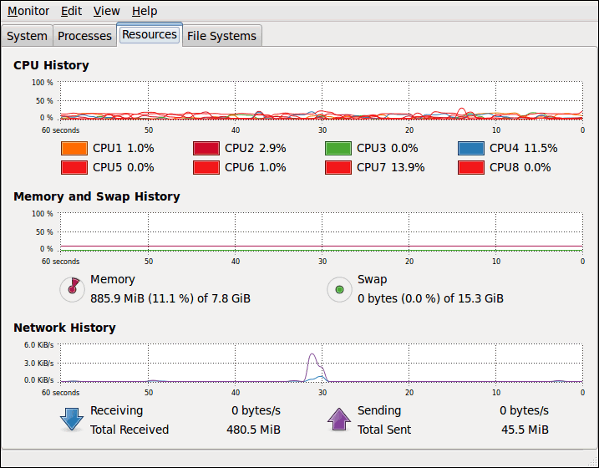
$ **ps -eo pcpu,pid,user,args | sort -k 1 -r | head -10**

Similar output can be obtained by using the command ‘**iostat**’ as root:

root@gateway [~]# **iostat -xtc 5 3**  
Linux 2.6.18-028stab094.3 (gateway.firewall.cx) 01/11/2012  
  
Time: 01:13:15 AM  
avg-cpu: %user %nice %system %iowait %steal %idle  
2.38 0.01 0.43 0.46 0.00 96.72  
  
Time: 01:13:20 AM  
avg-cpu: %user %nice %system %iowait %steal %idle  
3.89 0.00 0.26 0.09 0.00 95.77  
  
Time: 01:13:25 AM  
avg-cpu: %user %nice %system %iowait %steal %idle  
0.31 0.00 0.15 1.07 0.00 98.47

This will show three outputs every five seconds and show the information since the last reboot.

CPU usage under GUI is very well depicted by the Gnome System Monitor and other system monitoring applications. These are also useful for monitoring remote servers. Detailed memory maps can be accessed, signals can be sent and processes controlled remotely.

****

Gnome-System-Monitor

**What’s Cooking?**

How do you know what processes are currently running in your Linux system? There are innumerable ways of getting to see this information. The handiest applications are the old faithfuls - ‘**top**’ and ‘**htop**’. They will give a real-time image of what is going on under the hood. However, if you prefer a more static view, use ‘**ps**’. To see all processes try ‘**ps -A’** or ‘**ps -e**’:

root@gateway [~]# **ps -e**  
PID TTY TIME CMD  
1 ? 00:01:41 init  
3201 ? 00:00:00 leechprotect  
3208 ? 00:00:00 httpd  
3360 ? 00:00:00 httpd  
3490 ? 00:00:00 httpd  
3530 ? 00:00:00 httpd  
3532 ? 00:00:00 httpd  
3533 ? 00:00:00 httpd  
3535 ? 00:00:00 httpd  
3575 ? 00:00:00 httpd  
3576 ? 00:00:00 httpd  
3631 ? 00:00:00 imap  
3694 ? 00:00:00 httpd  
3705 ? 00:00:00 httpd  
3770 ? 00:00:00 imap  
3774 pts/0 00:00:00 ps  
5407 ? 00:00:13 dovecot  
5408 ? 00:00:12 dovecot-auth  
5416 ? 00:00:10 pop3-login  
5417 ? 00:00:49 pop3-login  
5418 ? 00:00:01 imap-login  
5419 ? 00:00:04 imap-login  
9745 ? 00:00:01 lfd  
11501 ? 00:01:35 spamd  
23948 ? 00:00:05 exim  
23993 ? 00:01:00 spamd  
24477 ? 00:00:04 queueprocd  
24494 ? 00:01:20 tailwatchd  
24526 ? 00:00:00 cpdavd  
24536 ? 00:00:02 cpanellogd  
24543 ? 00:00:33 cpsrvd-ssl  
25952 ? 00:20:17 named  
26374 ? 00:00:00 udevd  
28524 ? 00:00:00 sshd  
28531 pts/0 00:00:00 bash  
29834 ? 00:00:00 sshd  
30426 ? 00:11:27 syslogd  
30429 ? 00:00:00 klogd  
30473 ? 00:00:00 xinetd  
30485 ? 00:00:00 mysqld\_safe  
30549 ? 1-15:07:28 mysqld  
32158 ? 00:06:29 httpd  
32166 ? 00:12:39 pure-ftpd  
32168 ? 00:07:12 pure-authd  
32181 ? 00:01:06 crond  
32368 ? 00:00:00 saslauthd  
32373 ? 00:00:00 saslauthd  
PS is an extremely powerful and versatile command, and you can learn more by ‘**ps --h**’:  
root@gateway [~]# **ps --h**\*\*\*\*\*\*\*\*\* simple selection \*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\* selection by list \*\*\*\*\*\*\*\*\*  
-A all processes -C by command name  
-N negate selection -G by real group ID (supports names)  
-a all w/ tty except session leaders -U by real user ID (supports names)  
-d all except session leaders -g by session OR by effective group name  
-e all processes -p by process ID  
T all processes on this terminal -s processes in the sessions given  
a all w/ tty, including other users -t by tty  
g OBSOLETE -- DO NOT USE -u by effective user ID (supports names)  
r only running processes U processes for specified users  
x processes w/o controlling ttys t by tty  
\*\*\*\*\*\*\*\*\*\*\* output format \*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\* long options \*\*\*\*\*\*\*\*\*\*\*  
-o,o user-defined -f full --Group --User --pid --cols --ppid  
-j,j job control s signal --group --user --sid --rows --info  
-O,O preloaded -o v virtual memory --cumulative --format --deselect  
-l,l long u user-oriented --sort --tty --forest --version  
-F extra full X registers --heading --no-heading --context  
\*\*\*\*\*\*\*\*\* misc options \*\*\*\*\*\*\*\*\*  
-V,V show version L list format codes f ASCII art forest  
-m,m,-L,-T,H threads S children in sum -y change -l format  
-M,Z security data c true command name -c scheduling class  
-w,w wide output n numeric WCHAN,UID -H process hierarchy

Linux system monitoring Tools

Default:

sar

top

iostat

mpstat

VMSTAT

free

mrtg

lsof

netstat

watch

others

======

NAGIOS

opmanager