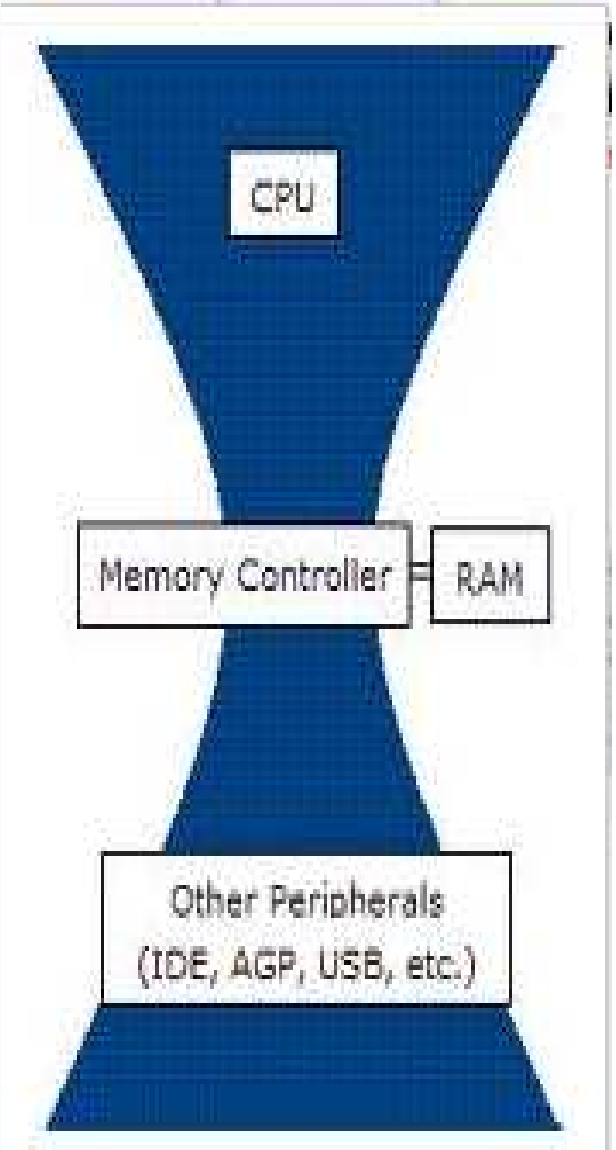
SYSTEM AVAILABILITY MONITORING

JASPREET SINGH
Jr. Consultant
CERT-In

ISSUES COVERED

- Finding out bottlenecks.
  - Disk (storage) bottlenecks.
  - CPU and memory bottlenecks.
  - Network bottlenecks.



BOTTLENECK

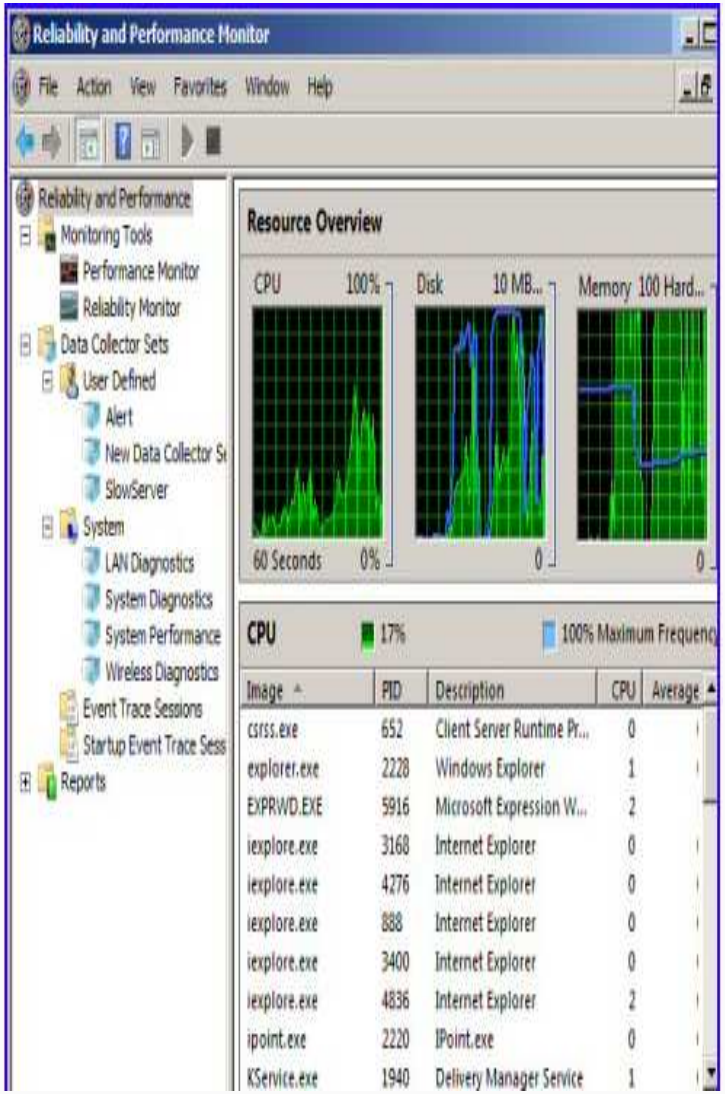
- A **bottleneck** is a phenomenon where the performance or capacity of an entire system is limited by a single or limited number of components or resources.
- A **bottleneck** in Computer World is one process in a chain of processes, such that its limited capacity reduces the capacity of the whole chain.

DISK (STORAGE) BOTTLENECK

- Workloads are now multi-tenant, with multiple shared servers and networks trying to access storage.
- The complexity of data in today's world not only increases the physical size required to store this data but also the processing and storage I/O required to create, modify, analyze or test the data.
- In all cases reliable, predictable, scalable storage I/O performance is critical.
- Installing another disk offers a quick fix.
- Compress your drive
- Apply disk quotas to restrict growth of user files.

CPU BOTTLENECK

- CPU bottleneck is like a roof. The power of your CPU determines how high the roof is.
- Processors never rest. Once powered up, they must always be executing some thread of instructions. When not executing the thread of an active user or system process, they execute a thread of a process called *Idle*.
- CPU Util ~= 100%, then CPU bottleneck
- exists



MEMORY BOTTLENECK

- Machines with plenty of RAM rarely give problems.
- Abundant RAM compensates for strain on other resources.
- The servers most likely to suffer from memory shortage are pure database servers for example, Oracle or SQL. Email servers also require plenty of RAM.
- Paging
- Increase size of virtual memory.

NETWORK BOTTLENECK

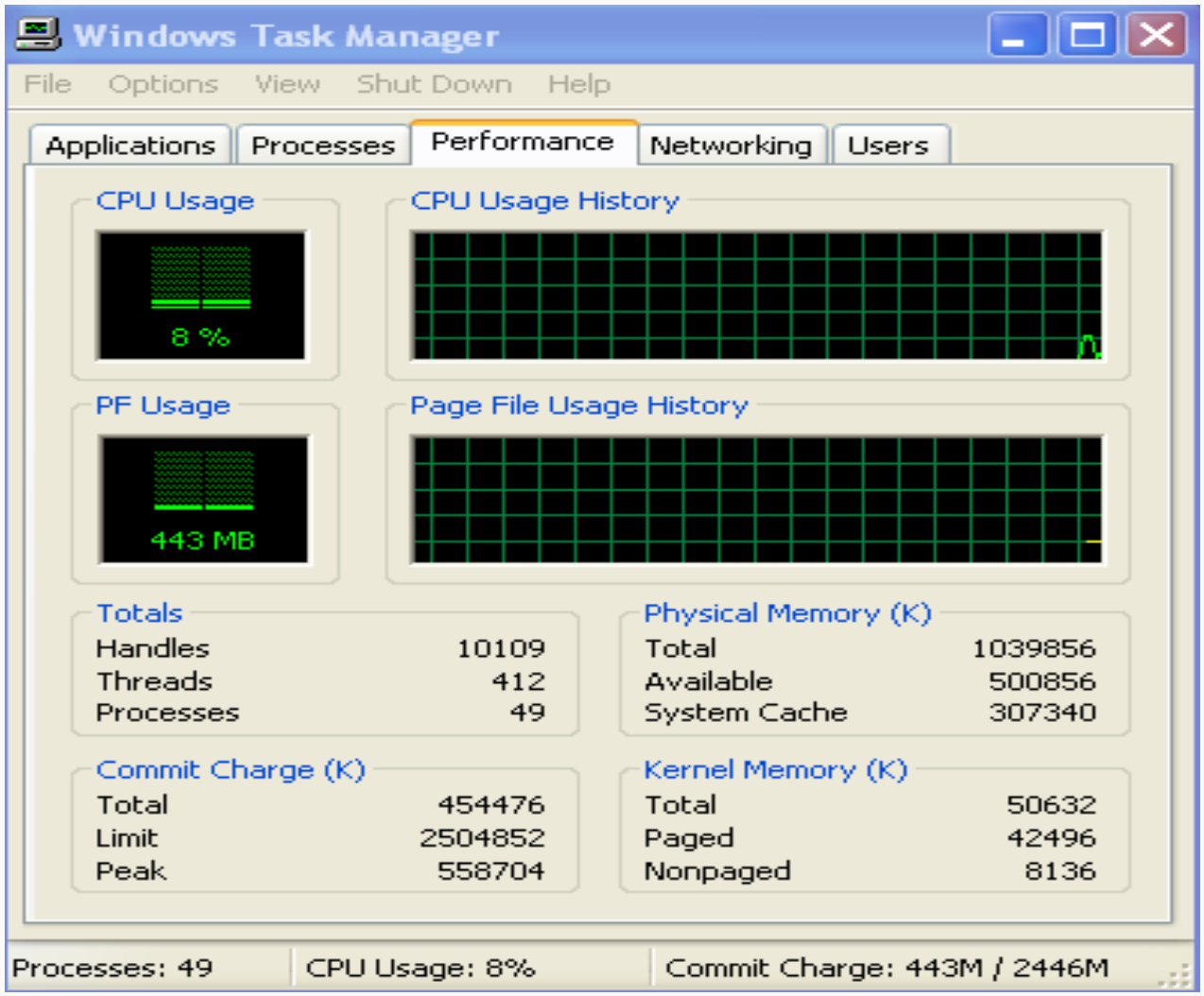
- Typical causes for network bottlenecks are an overloaded server, an overloaded network, or a loss of network integrity.
- Splitting the adapters across multiple Ethernet segments is an effective way to eliminate an overloaded network

NB CONTD.

- Use adapters with the highest bandwidth available for best performance. Remove unused network adapters to reduce overhead.
- Reducing the number of protocols installed can increase performance.
- Use offline folders to work on network applications without being connected to a network. Offline folders make use of client-side caching, thereby reducing network traffic.

TOOLS TO MONITOR AND FIX BOTTLENECK

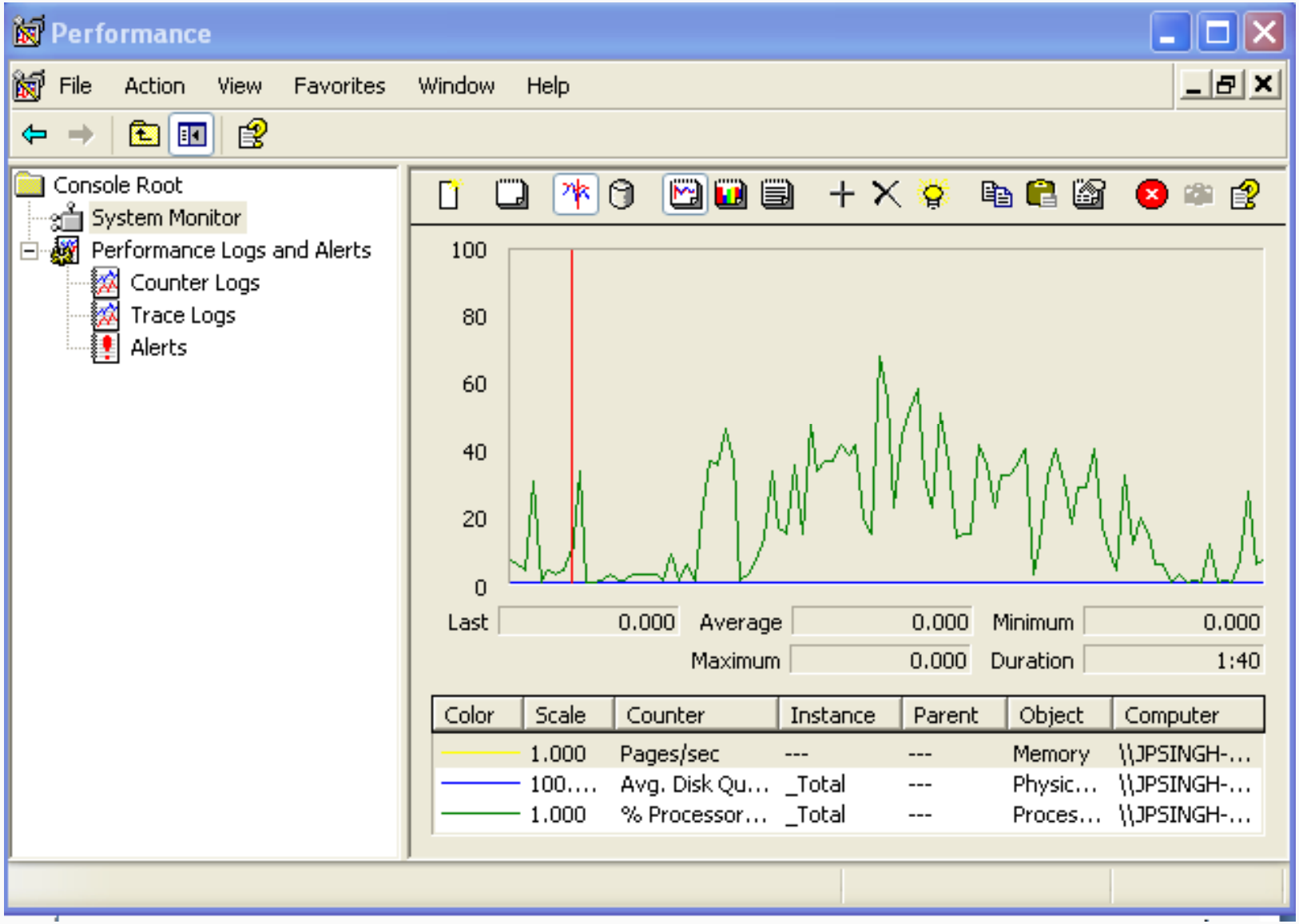
WINDOWS TASK MANAGER



WTM CONTD

- Provide a solution to monitor many issues.
- Check Users logged in the system with the user tab.
- Check the memory usage with Processes tab which also shows the running processes.
- Terminate unnecessary processes.
- Status of applications whether running or not responding from Applications tab.

PERFMON


Performance
File
Action
View
Favorites
Window
Help
Console Root
System Monitor
Performance Logs and Alerts
Counter Logs
Trace Logs
Alerts
100
80
60
40
20
0
Last 0.000
Average 0.000
Minimum 0.000
Maximum 0.000
Duration 1:40
Color Scale Counter Instance Parent Object Computer
1.000 Pages/sec --- --- Memory \\JPSINGH-...
100.... Avg. Disk Qu... _Total --- Physic... \\JPSINGH-...
1.000 % Processor... _Total --- Proces... \\JPSINGH-...

PERFMON CONTD.

- Perfmon is a Performance Monitoring tool that is shipped with windows.
- Performance counters are configured before running the test and it automatically collects the data during the scenario run.

top

- **top - Process Activity Command**
- The top program provides a dynamic real-time view of a running system i.e. actual process activity. By default, it displays the most CPU-intensive tasks running on the server and updates the list every five seconds.

OUTPUT

```
top - 00:14:33 up 1 day, 20:07,  5 users,  load average: 0.53, 0.60, 0.55
Tasks: 187 total,   2 running, 184 sleeping,   0 stopped,   1 zombie
Cpu(s):  4.6%us,  0.5%sy,  0.0%ni, 94.8%id,  0.1%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   8299044k total,  8020704k used,   279160k free,   221276k buffers
Swap:  19518888k total,    29764k used,  19489124k free,  6454792k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 8352 vivek     20   0  240m  73m  25m S   10  0.9   6:21.26 liferea-bin
 4575 vivek     20   0  227m 123m  38m S    5  1.5  86:26.28 deluge
25031 vivek     20   0  766m 472m  34m S    3  5.8  42:02.18 firefox-bin
 6873 root      20   0  308m  85m  17m S    2  1.2  88:44.31 Xorg
 7268 vivek     20   0 31412 5240 3828 S    1  0.1  10:26.03 pulseaudio
13592 root      15  -5     0    0    0 S    1  0.0  10:54.52 ntop_wq
32484 vivek     20   0 84312  26m  11m S    0  0.3   0:07.67 gnome-terminal
    1 root      20   0  1908  804  652 S    0  0.0   0:01.69 init
    2 root      15  -5     0    0    0 S    0  0.0   0:00.00 kthreadd
    3 root      RT  -5     0    0    0 S    0  0.0   0:00.02 migration/0
    4 root      15  -5     0    0    0 R    0  0.0   0:27.71 ksoftirqd/0
    5 root      RT  -5     0    0    0 S    0  0.0   0:00.00 watchdog/0
    6 root      RT  -5     0    0    0 S    0  0.0   0:00.01 migration/1
    7 root      15  -5     0    0    0 S    0  0.0   0:06.38 ksoftirqd/1
    8 root      RT  -5     0    0    0 S    0  0.0   0:00.00 watchdog/1
    9 root      RT  -5     0    0    0 S    0  0.0   0:00.02 migration/2
   10 root      15  -5     0    0    0 S    0  0.0   0:05.59 ksoftirqd/2
   11 root      RT  -5     0    0    0 S    0  0.0   0:00.00 watchdog/2
   12 root      RT  -5     0    0    0 S    0  0.0   0:00.03 migration/3
   13 root      15  -5     0    0    0 S    0  0.0   0:05.85 ksoftirqd/3
   14 root      RT  -5     0    0    0 S    0  0.0   0:00.00 watchdog/3
   15 root      15  -5     0    0    0 S    0  0.0   0:00.27 events/0
   16 root      15  -5     0    0    0 S    0  0.0   0:00.52 events/1
   17 root      15  -5     0    0    0 S    0  0.0   0:00.44 events/2
   18 root      15  -5     0    0    0 S    0  0.0   0:00.50 events/3
   19 root      15  -5     0    0    0 S    0  0.0   0:00.01 khelper
   61 root      15  -5     0    0    0 S    0  0.0   0:00.00 kintegrityd/0
   62 root      15  -5     0    0    0 S    0  0.0   0:00.00 kintegrityd/1
   63 root      15  -5     0    0    0 S    0  0.0   0:00.00 kintegrityd/2
   64 root      15  -5     0    0    0 S    0  0.0   0:00.00 kintegrityd/3
   66 root      15  -5     0    0    0 S    0  0.0   0:00.05 kblockd/0
   67 root      15  -5     0    0    0 S    0  0.0   0:00.18 kblockd/1
   68 root      15  -5     0    0    0 S    0  0.0   0:00.23 kblockd/2
   69 root      15  -5     0    0    0 S    0  0.0   0:00.11 kblockd/3
   71 root      15  -5     0    0    0 S    0  0.0   0:00.00 kacpid
   72 root      15  -5     0    0    0 S    0  0.0   0:00.00 kacpi_notify
  153 root      15  -5     0    0    0 S    0  0.0   0:00.00 cqueue
  157 root      15  -5     0    0    0 S    0  0.0   0:00.01 kseriod
  231 root      15  -5     0    0    0 S    0  0.0   0:02.25 kswapd0
```

Commonly Used Hot Keys

The top command provides several useful hot keys:

t - Displays summary information off and on.

m - Displays memory information off and on.

A - Sorts the display by top consumers of various system resources. Useful for quick identification of performance-hungry tasks on a system.

f - Enters an interactive configuration screen for top. Helpful for setting up top for a specific task.

Commonly Used Hot Keys

o - Enables you to interactively select the ordering within top.

r - Issues renice command.

k - Issues kill command.

z - Turn on or off color/mono

w

- Find Out Who Is Logged on And What They Are Doing
- w command displays information about the users currently on the machine, and their processes.
- # w username
  # w vivek

Sample Outputs:

```
 17:58:47 up 5 days, 20:28,  2 users,  load average: 0.36, 0.26, 0.24
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
root     pts/0    10.1.3.145       14:55    5.00s  0.04s  0.02s vim /etc/resolv.conf
root     pts/1    10.1.3.145       17:43    0.00s  0.03s  0.00s w
```

uptime

- Tell How Long The System Has Been Running
- The uptime command can be used to see how long the server has been running. The current time, how long the system has been running, how many users are currently logged on, and the system load averages for the past 1, 5, and 15 minutes.
- # uptime
- 18:02:41 up 41 days, 23:42, 1 user, load average: 0.00, 0.00, 0.00

ps

- Displays The Processes
- ps command will report a snapshot of the current processes. To select all processes use the -A or -e option

  # ps –A

  # ps –e
- ps is just like top but provides more information.
- Print All Process On The Server

  # ps -ef OR # ps axu

Sample Outputs:

```
  PID TTY          TIME CMD
    1 ?        00:00:02 init
    2 ?        00:00:02 migration/0
    3 ?        00:00:01 ksoftirqd/0
    4 ?        00:00:00 watchdog/0
    5 ?        00:00:00 migration/1
    6 ?        00:00:15 ksoftirqd/1
....
.....
 4881 ?        00:53:28 java
 4885 tty1     00:00:00 mingetty
 4886 tty2     00:00:00 mingetty
 4887 tty3     00:00:00 mingetty
 4888 tty4     00:00:00 mingetty
 4891 tty5     00:00:00 mingetty
 4892 tty6     00:00:00 mingetty
 4893 ttyS1    00:00:00 agetty
12853 ?        00:00:00 cifsoplockd
12854 ?        00:00:00 cifsdnotifyd
14231 ?        00:10:34 lighttpd
14232 ?        00:00:00 php-cgi
54981 pts/0    00:00:00 vim
55465 ?        00:00:00 php-cgi
55546 ?        00:00:00 bind9-snmp-stat
55704 pts/1    00:00:00 ps
```

free

- Memory Usage
- The command free displays the total amount of free and used physical and swap memory in the system, as well as the buffers used by the kernel.
- # free

Sample Output:

```
             total       used       free     shared    buffers     cached
Mem:      12302896    9739664    2563232          0     523124    5154740
-/+ buffers/cache:    4061800    8241096
Swap:      1052248          0    1052248
```

pmap

- Process Memory Usage
- The command pmap report memory map of a process. Use this command to find out causes of memory bottlenecks.
- # pmap -d PID
- To display process memory information for pid # 47394, enter:
- # pmap -d 47394

Sample Outputs:

The last line is very important:

mapped: 933712K total amount of memory mapped to files
writeable/private: 4304K the amount of private address space
shared: 768000K the amount of address space this process is sharing with others

```
47394:   /usr/bin/php-cgi
Address           Kbytes Mode  Offset           Device    Mapping
0000000000400000    2584 r-x-- 0000000000000000 008:00002 php-cgi
0000000000886000     140 rw--- 0000000000286000 008:00002 php-cgi
00000000008a9000      52 rw--- 00000000008a9000 000:00000   [ anon ]
0000000000aa8000      76 rw--- 0000000002a8000 008:00002 php-cgi
00000000f678000    1980 rw--- 00000000f678000 000:00000   [ anon ]
000000314a600000     112 r-x-- 0000000000000000 008:00002 ld-2.5.so
000000314a81b000       4 r---- 000000000001b000 008:00002 ld-2.5.so
000000314a81c000       4 rw--- 000000000001c000 008:00002 ld-2.5.so
000000314aa00000    1328 r-x-- 0000000000000000 008:00002 libc-2.5.so
000000314ab4c000    2048 ----- 000000000014c000 008:00002 libc-2.5.so
.....
......
..
00002af8d48fd000       4 rw--- 0000000000006000 008:00002 xsl.so
00002af8d490c000      40 r-x-- 0000000000000000 008:00002 libnss_files-2.5.so
00002af8d4916000    2044 ----- 000000000000a000 008:00002 libnss_files-2.5.so
00002af8d4b15000       4 r---- 0000000000009000 008:00002 libnss_files-2.5.so
00002af8d4b16000       4 rw--- 000000000000a000 008:00002 libnss_files-2.5.so
00002af8d4b17000  768000 rw-s- 0000000000000000 000:00009 zero (deleted)
00007fffc95fe000      84 rw--- 00007ffffffea000 000:00000   [ stack ]
ffffffffff600000    8192 ----- 0000000000000000 000:00000   [ anon ]
mapped: 933712K    writeable/private: 4304K    shared: 768000K
```

/Proc file system

- /proc file system provides detailed information about various hardware devices and other Linux kernel information. Common /proc examples:
- # cat /proc/cpuinfo
  # cat /proc/meminfo
  # cat /proc/zoneinfo
  # cat /proc/mounts

THANK YOU