Troubleshooting Memory and CPU Issues

Andrew Mallett
Author and Trainer

@theurbanpenguin www.theurbanpenguin.com

Overview

Troubleshooting CPU and Memory

Process Monitoring

- runaway
- zombies
- high load avg / run queues / utilities

Monitoring Memory

- managing swap
- OOM / memory exhaustion

Installing Sysstat

Historical Monitoring

We can look at load averages over the last few minutes but without adding extra packages we cannot gather information over a period of time:

- Installing sysstat collect data every 10 mins
- cat /etc/cron.d/sysstat

Demo

To begin we will install sysstat:

- We will be collecting data from when it is installed
- Collection interval is every 10 minutes

CPU Load Averages

Uptime

The uptime command will show load averages for your CPUs over the last 1 minute, 5 minutes and 15 minutes.

```
$ uptime
$ uptime -s (just uptime no load averages)
$ uptime -p (just uptime but simplified or so-called pretty output)
$ man uptime
$ cat /proc/uptime
$ man procfs
$ lscpu
$ lscpu | grep '^CPU(s):' | cut -f2 -d: | tr -d ' '
```

Understanding Uptime and Load Averages

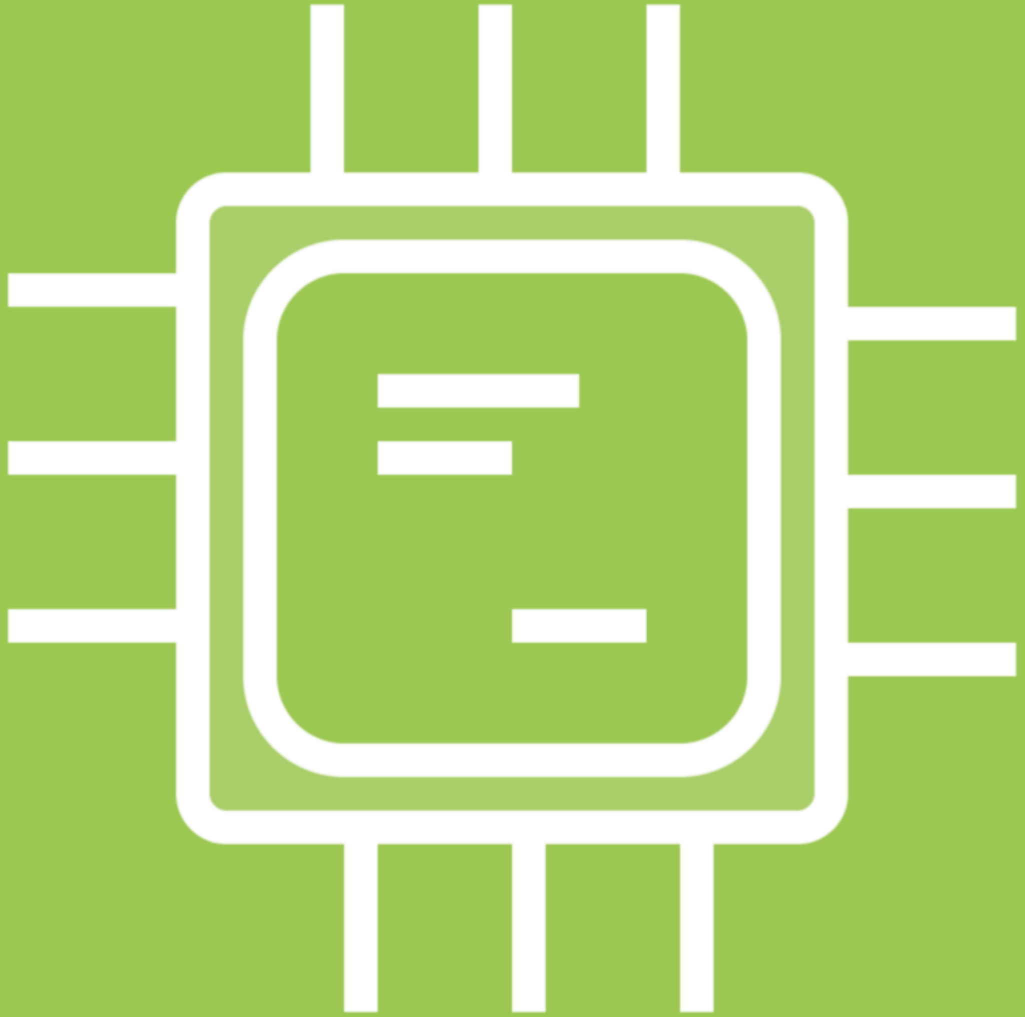
The command uptime will show load averages, but these relate to the number of CPUs you have on the system. High averages are OK, but it just means that some requests may be queued, for a short time this is acceptable but not for longer periods.

Demo

Let's look for the uptime command and load averages:

- uptime
- /proc/uptime
- lscpu

Monitoring High CPU



Mpstat

As well as collecting regular data, sysstat provides real time commands we can use to monitor CPU performance. To see the historical data, we use **sar -u.**

```
$ mpstat
$ tar -cJf ysr.tar.xz /usr/share &> /dev/null &
$ mpstat 1 60
$ tar -cJf ysr.tar.xz /usr/share &> /dev/null &
$ mpstat -P 0 1 60
$ sar -u / sar -u
```

Monitoring CPU Utilization in Realtime

The sysstat utility, mpstat, monitors CPU utilization. My system has 2 CPUs, the xz compressor is a single process and runs on a single CPU. Without specifying the CPU, we see a high of 50% utilization. If we specify the processor, it will be maxed out at 100%.

```
$ tar -cJf ysr.tar.xz /usr/share &> /dev/null &
$ top
```

Identify CPU Hog Using Top

The command top, will identify the process taking the CPU time.

Demo

Using mpstat to monitor CPU:

- All processors
- Single processor
- Using top to identify CPU hog

Understanding Zombie Processes

Zombie Processes

A Zombie process is quite normal in Linux but should not be long-lived. A parent process starts a child process. When the child process completes the child becomes a zombie and is cleaned up by the parent.

```
$ apt install -y build-essential
```

Adding Compiler Tools

We will the build-essential package, allowing us to compile a simple program to demonstrate zombie processes.

Source Code For Zombie Process

zombie.c

```
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
int main () {
  pid_t child_pid;
  child_pid = fork ();
  if (child_pid > 0) {
    sleep (300);
  }
  else {
    exit (0);
  }
  return 0;
}
```

```
$ gcc zombie.c -o zombie
$ ./zombie &
$ ps  -elf | grep -w Z
$ top -b -n 1 -u $USER
```

Monitoring Zombie Processes

The parent process will sleep for 5 minutes, during the sleep time the child process will end and will stay in a zombie state until the parent wakes and stops the process.

"I am sure we have all had a nap and woken to find our children have become zombies 🤣".

Demo

Creating and monitoring zombies:

- create source
- compile and link program
- execute
- monitor

Memory Monitoring

Memory

In Linux memory can be in RAM or as a SWAP space on disk. RAM will be fastest, but SSD SWAP space can come a close second

```
$ free -h
$ cat /proc/meminfo
$ sar -r
```

Viewing Memory Usage

We can view memory utilization as a whole, that data come from the procfs. To view collected information, we have sar -r.

```
$ top
f
Arrow Down To MEM
s
q
```

Show Memory Hogs

To show processes sorted by most memory used we can use top and filter on memory.

Demo

Working with our system RAM:

- display memory information
- display historical memory information
- display the most memory consuming processes on the system

Managing Swap Space

```
$ free -h
$ swapon -s
```

Viewing Memory Usage

We can view memory utilization as a whole, or specifically for swap. If we have multiple swap devices, we can set a priority for each device. The highest value will be used first

```
$ sudo fallocate -l 500m /var/swap1
$ sudo fallocate -l 500m /var/swap2
$ sudo chmod 600 /var/swap*
$ sudo mkswap /var/swap1
$ sudo mkswap /var/swap2
$ sudo swapon /var/swap1
$ swapon -s
$ sudo swapon -p 10 /var/swap2
```

Create Swap Space

We can use disks, partitions or files as swap space. For ease we will use swap files. We add swap headers to the space with mkswap and we mount the swap space with swapon. The default priority starts at -2 but we can set a higher priority for space we want to use first.

```
$ cat /proc/sys/vm/swappiness
```

Swappiness

We can affect how likely the Kernel is to swap to disk over dropping memory pages through the swappiness setting. Ranging from 0 through to 100, with 0 being less likely to swap to disk and 100 being most likely to swap to disk. The default value is 60.

Demo

Create and monitor swap:

- create raw files
- set permissions
- add headers
- mount swap

Summary

We have introduced monitoring of processes and memory

- uptime
- lscpu
- top
- mpstat
- sar
- zombie processes
- swap management and system swappiness

Up Next:
Troubleshoot Storage Issues