Practical Linux Performance and Application Troubleshooting

by Tanel Poder | https://blog.tanelpoder.com/seminar

Table of Contents

The training sessions of this class are split into two separate weeks:

- Part 1 - Application Troubleshooting on Linux (intermediate)
- Part 2 - Linux OS Troubleshooting and Performance Tuning (advanced)

The *Part 1* starts at higher level and aims to stay *mostly* at application troubleshooting and OS fundamentals level, assuming that the system is not overloaded and the kernel & lower levels of hardware/OS stack work just fine. The *Part 2* then dives deeper, revisiting some of the topics at much more detailed level (including OS kernel issues) and addressing new areas too.

Part 1 – Application Troubleshooting on Linux

1. **Introduction: Application and OS touchpoint**
   - How do applications interact with the OS and hardware?
   - How do system calls work?
   - Tracing process activity
   - Process virtual memory
   - Private memory allocation basics
   - Executable loading and memory mapping
   - Do we have a memory shortage?

2. **Measuring Process Activity**
   - Processes, threads, tasks
   - Process creation, cleanup and zombies
   - CPU scheduler and thread states
   - What does the Linux system load really mean? (It's different from classic Unix)
   - Sampling & profiling any OS process activity
   - Application CPU usage & CPU profiling (Java, C/C++, Python etc)

3. **File access and disk I/O**
   - Everything is a file
   - File descriptors
   - Where are my log files?
   - Filesystems & caching
   - Block devices and disk I/O
   - Using iostat for IO subsystem performance summary

4. **Process Memory usage**
    - Process heaps, data segments
    - Java applications' memory analysis (JVM)
    - PageCache and buffers
    - How much memory is really free?
    - Who's using the most of physical RAM?
    - Who is causing swapping?
    - Configuring Swap space and memory overcommitting
    - When should I change Kernel "swappiness"?
    - Static HugePages, Transaprent HugePages and page size considerations

5. **TCP and Networking**
    - TCP, UDP sockets and network I/O
    - Connectivity troubleshooting throughout the network stack
    - Routing examination from a Linux host
    - Network latency analysis
    - Application activity tracking with tcpdump and WireShark
    - Remote applications' response time analysis
    - Network throughput validation & TCP buffer tuning for throughput

Part 2 – Linux OS Troubleshooting and Performance Tuning

6. **How does Linux Kernel virtualize memory and talk to hardware?**
    - Hardware topology and interfaces
    - Userland memory, kernel memory and device mapped memory
    - Page tables, Page faults, TLB misses, exceptions and traps
    - CPU caching and why RAM is slow
    - NUMA memory & process placement considerations
    - Device drivers & kernel modules
    - Interrupt handling
    - Kernel worker threads
    - Tracing Linux kernel

7. **Troubleshooting Linux Kernel activity and CPU usage**
    - Troubleshooting high kernel CPU usage
    - Troubleshooting high system load without high I/O or CPU usage
    - CPU scheduling modes, OS jitter and scheduling latency
    - Scheduler priorities and priority inversion
    - Dealing with a large number of processes
    - CPU usage in containers, VMs & the cloud
    - Reading kernel source – finding the relevant files and functions

8. **Achieving high-performance block device I/O on Linux**
   - Modern flash disk arrays can do millions of IOPS!
   - At millions of IOPS, I/O becomes a CPU problem
   - Configuring Linux for 1M IOPS
   - Measuring I/O latency through layers (filesystem, logical volumes, disk devices)
   - Configuring I/O scheduler and priorities
   - Choosing the right filesystem and pagecache strategy

9. **Troubleshooting IPC, RPC, process communication and synchronization**
   - Sockets, pipes, STDOUT and STDIN
   - Signal handling
   - Spinlocks, Mutexes, Futexes and Semaphores
   - Timers, sleeping, waiting
   - File system advisory locks
   - NFS access troubleshooting
   - NFS I/O performance & configuration

10. **Troubleshooting errors, crashes & dumps**
    - Interpreting system call errors
    - Configuring Linux coredumps
    - Analyzing crashed processes
    - Diagnosing and avoiding the Out-of-Memory killer (OOM killer)
    - Configuring Kernel crashdumps
    - Analyzing kernel Panics & OS Crashes

Tools & Techniques

An example set of tools we'll use in the class is listed below. This is by no means the complete list. This training teaches you to use the right tool for the right problem, for safe and systematic troubleshooting drilldown.

- psnapper (The Process Snapper – an open source "passive profiling" tool written by me)
- vmstat, sar, dstat
- ps, top, htop
- iostat, iotop, blktrace
- free, smem
- lsof, fuser, /proc/PID/fd
- strace, systemTap, ftrace
- application analysis with perf, strace, gdb, pstack
- kernel thread analysis with /proc/PID/stack, perf
- /proc/meminfo, /proc/vmstat, /proc/cpuinfo
- pmap (/proc/PID/maps, /proc/PID/smaps)

- lsblk, lsscsi, lscpu, lslocks
- netstat, nc, ss, traceroute, tracepath, ip, ifconfig, arp
- tcpdump, WireShark, TraceCompass
- iperf, fio, JMeter, MySQL & Postgres benchmarking tools
- tools reading the /proc filesystem
- various custom shell and Python scripts

We are going to be practical, we'll use a simpler and less intrusive tool whenever its available for a given troubleshooting task. In other words, we aren't going to install kernel debuggers or write SystemTap scripts when the problem is systematically diagnosable using a simple shell script that just reads */proc* file system entries.

Platform & Linux Versions

I'll use the typical enterprise Linux distributions for the class demos:

- RHEL 6, RHEL 7 clones (CentOS & OEL)
- Ubuntu Server 16.04, 18.04
- Running on bare-metal, VMs, container setups and in the cloud

Many of the tools and techniques will work on somewhat older Linux versions too.

Further Questions & Sign up

- See the dates and sign up here – https://blog.tanelpoder.com/seminar
- If you have more questions – email seminars@poderc.com