Performance & Resource Troubleshooting Case Studies for CloudOps Engineers

10 Real-world Performance and Resource Scenarios with Commands, Metrics, and Solutions

Case 1: High CPU Usage by Rogue Process

Problem: Service latency increased; CPU usage at 100% on one core.

Investigation:

```
$ top -b -n1 | head -20
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
4321 webapp 20 0 1.2g 300m 20m R 99.9 2.1 123:45 python3 worker.py
$ strace -p 4321 | head -n 5
read(3, ... ) = ... (looping system calls)
```

Root Cause: Worker process entered busy-loop due to bug handling malformed messages.

Resolution: Restarted service and deployed hotfix to handle message parsing gracefully.

Prevention: Add process-level CPU alerts and use cgroups to limit runaway processes.

Case 2: Memory Leak in Application Container

Problem: Container memory usage grew until OOMKilled by kernel.

Investigation:

```
$ docker stats --no-stream
MEM USAGE / LIMIT 4.8GiB / 4GiB
$ dmesg | grep -i oom
Out of memory: Kill process 5589 (java) score 987
```

Root Cause: Application had memory leak in cache implementation inside container.

Resolution: Updated code to use bounded cache and redeployed container with resource limits.

Prevention: Enforce memory limits in container runtime and add heap profiling.

Case 3: I/O Wait Spike Due to Disk Saturation

Problem: Users experienced slow queries with high iowait.

Investigation:

```
$ iostat -x 1 3
Device r/s w/s await %util
xvdc 10.0 200.0 350.2 99.9
$ iotop -o -b -n 3 | head -20
PID COMMAND DISK READ DISK WRITE
3245 postgres 0.00B 120.0M/s
```

Root Cause: Backup job and checkpoint combined saturated disk bandwidth.

Resolution: Staggered backups, moved WAL to separate faster disk, and tuned checkpoint settings.

Prevention: Schedule heavy I/O during low-traffic windows and use provisioned IOPS where available.

Case 4: System Slow After Swap Exhaustion

Problem: System became unresponsive when swap filled and thrashing began.

Investigation:

```
$ free -m
Mem: 8192 7900 292 Swap: 2048 2048 0
$ vmstat 1 5
si so high, indicating swapping
```

Root Cause: Insufficient RAM for workload leading to swap exhaustion under load spike.

Resolution: Added RAM and tuned swappiness to 10; optimized application memory usage.

Prevention: Right-size instances and monitor memory; use alerts for swap usage.

Case 5: Network Latency on VM After Kernel Update

Problem: Network latency increased after applying kernel patches.
Investigation:

```
$ ping -c 10 internal-service
rtt min/avg/max/mdev = 0.45/12.12/150.34/32.21 ms
$ ethtool -k eth0
Offload features changed after update
```

Root Cause: Kernel change disabled NIC offload settings, causing CPU overhead and latency.
Resolution: Restored previous offload settings via ethtool and updated network drivers.
Prevention: Test kernel updates in staging and capture baseline network metrics.

Case 6: Load Average Spike from Cron Jobs

Problem: Regular spike in load average at 02:00 every night.
Investigation:

```
$ crontab -l
0 2 * * * /usr/local/bin/backup.sh
$ uptime
load average: 25.10, 3.00, 1.20
```

Root Cause: Concurrent cron jobs started heavy tasks simultaneously causing load spike.
Resolution: Staggered cron schedules and used flock to serialize backups.
Prevention: Use centralized scheduler/orchestration instead of individual cron across nodes.

Case 7: Zombie Process Accumulation

Problem: System had thousands of defunct processes increasing PID usage.
Investigation:

```
$ ps -el | grep Z
Z 12345 0 80 0 0 ? 00:00 0:00
```

Root Cause: Parent processes not reaping child processes due to improper wait handling.
Resolution: Fixed application to call wait()/waitpid() or used reaper process; restarted service.
Prevention: Implement process supervision and monitor defunct process counts.

Case 8: NTP Desynchronization Affecting Application Timestamps

Problem: Timestamps in logs drifted and caused time-sensitive jobs to fail.
Investigation:

```
$ timedatectl status
System clock synchronized: no
$ chronyc tracking
Reference ID: .LOCL. -1 (unsynchronized)
```

Root Cause: NTP service lost peers after network partition and drift accumulated.
Resolution: Restarted chronyd and configured multiple upstream NTP servers; forced sync.
Prevention: Monitor time offset and configure redundant NTP sources.

Case 9: High Context Switches from Excessive Threads

Problem: High CPU overhead from context switching degraded throughput.

Investigation:

```
$ vmstat 1 5
cs 600000
$ pidstat -w 1 1 | grep app-service
high cs for specific PID
```

Root Cause: Application spawned excessive worker threads per request under load.

Resolution: Refactored to use event-driven model and reduced thread count.

Prevention: Profile thread usage and set reasonable thread pool limits.

Case 10: Slow SSH Login Due to Reverse DNS Lookup

Problem: SSH authentication delays around 10 seconds per login.

Investigation:

```
$ ssh -vvv user@host
GSSAPI key exchange... (timeout)
$ tail -n 50 /var/log/secure
reverse mapping checking getaddrinfo for 10.0.0.5 failed: Name or service not known
```

Root Cause: SSHd performing reverse DNS lookup which timed out due to missing PTR records.

Resolution: Disabled UseDNS in /etc/ssh/sshd_config and restarted sshd.

Prevention: Ensure DNS PTR records for hosts or disable reverse lookup to avoid login delays.