High CPU Usage:-

User Report:-The application is very slow, sometimes not responding at all.

CPU>90%

Debugging steps:-

Top – Real-time CPU/memory per process

htop- better visualization

uptime- load average

Then check which process is consuming the most CPU.

**ps -eo pid,ppid,cmd,%mem,%cpu –sort=%cpu | head**

This command help us to find PID of high CPU process.

ubuntu@ip-10-0-1-161:~$ ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%cpu | head

PID PPID CMD %MEM %CPU

2595 2540 sshd: ubuntu@pts/2 0.7 0.2

2596 2595 -bash 0.5 0.0

538 1 /usr/lib/snapd/snapd 3.8 0.0

1 0 /sbin/init 1.3 0.0

531 1 /snap/amazon-ssm-agent/1179 2.0 0.0

1579 1526 sshd: ubuntu@pts/1 0.7 0.0

969 857 sshd: ubuntu@pts/0 0.7 0.0

2327 1 /usr/libexec/packagekitd 2.0 0.0

128 1 /usr/lib/systemd/systemd-jo 1.5 0.0

Strace command intercepts every system calls that process makes(open ,read,write,futex,clock\_gettime) .

Strace command is attaches to already running process.

If I/O he heavy then there will be lots of read/write

If cpu spins then there will be separate futex/clock\_gettime loops.

**Strace command watching conversation between the process and linux kernel.**

**Strace –p <pid>**

**Strace –p 2595**

**No issue:-**

**clock\_gettime(CLOCK\_REALTIME, {tv\_sec=1756021109, tv\_nsec=89636658}) = 0**

**Above output shows that process is asking the system clock for the current time.**

**Issue:-**

**clock\_gettime(CLOCK\_REALTIME, {...}) = 0**

**clock\_gettime(CLOCK\_REALTIME, {...}) = 0**

**clock\_gettime(CLOCK\_REALTIME, {...}) = 0**

**... (hundreds/thousands of times continuously)**

**Above output showing abnormal behaviour.**

**Process stuck in tight loop, wasting CPU cycles only calling clock\_gettime( )**

**This is clear symptom of buggy code.(infinite loop, retry loop,or badly written timer function)**

**Strace command showing repeated clock\_gettime( ) calls.**

**After confirming process is taking more CPU. Then we will take below actions.**

1. **If its non-critical process- we can safely kill/restart.**

**Sudo kill -9 2595**

**Or just restart simple service**

**Sudo systemctl restart <service-name>**

1. **If its Production critical service- > Need to engage application owner before restart.**
2. **Restart the application service- clear the infinite loop temporarily.**

**If service restart not possible we may renice (make CPU lower)**

**sudo renice +10 –p <pid>**

**We report developers that process stuck in a loop calling clock\_gettime() . this indicates possible bug in timing /loop logic.**

**RCA:-**

After using strace, I noticed the process was looping with clock\_gettime() syscalls. This confirmed it was burning CPU in a busy loop.  
My immediate action was to mitigate impact by restarting/renicing the process.  
Then I collected system call summary and performance data (strace -c, perf top) and escalated to the dev team.  
The long-term fix was to correct the application code, but short-term I ensured service stability by restart and monitoring.”