Week 6: Performance Evaluation and Analysis

1. Aim of the Week

The aim of Week 6 was to evaluate the performance of the Ubuntu server under different workloads and analyse how the operating system behaves when system resources are stressed. This involved collecting baseline performance data, executing application load tests, identifying performance bottlenecks, and analysing system behaviour using quantitative measurements.

All testing was performed on the server and monitored remotely via SSH, following professional system administration practices.

2. Testing Methodology

Performance testing was carried out in a structured manner to ensure fair and repeatable results. For each test, baseline measurements were taken first while the system was idle. Application load was then applied, and the same metrics were collected for comparison.

The following metrics were monitored where appropriate:

- CPU usage
- Memory usage
- Disk I/O performance
- Network latency
- System responsiveness

Standard Linux command-line tools were used to observe operating system behaviour in real time.

3. Baseline Performance Measurements

Before applying any workloads, baseline performance data was collected to represent normal system operation.

The system uptime confirmed the server had only recently started, and the load average values indicated low background activity. Memory usage showed that approximately 1.2 GiB of RAM was in use, with sufficient available memory remaining. No swap space was configured at this stage.

Disk usage was minimal, and disk activity monitoring using iostat showed low I/O wait and high CPU idle percentages, indicating that the system was not under stress.

4. CPU and Memory Analysis

CPU and memory behaviour were monitored during the baseline phase using system statistics commands. The average CPU idle percentage was high, indicating that the processor was largely unused during idle operation. System memory usage remained stable, and no swapping activity was observed.

These results provided a clear baseline for comparison when workload tests are applied, showing that any later increases in CPU load or memory usage can be attributed to application activity rather than background processes.

5. Disk I/O Performance Analysis

Disk performance was evaluated using extended I/O statistics. The iostat -xz output showed low read and write activity during baseline testing. Disk utilisation percentages remained low, and average wait times were minimal, indicating that the storage subsystem was not a performance bottleneck under idle conditions.

The results confirmed that disk I/O performance was stable and responsive before load testing.

6. Network Performance and Latency

Network latency testing was attempted using ICMP echo requests. An initial error occurred due to an incorrect hostname being used instead of the server IP address, highlighting the importance of accurate network configuration during performance testing.

This issue was identified and corrected during the testing process. Network performance analysis focuses on latency and throughput, which will be compared against application load tests in later optimisation stages.

7. Service Response and System Behaviour

System services such as the web server were checked to confirm they were running correctly. Service status checks verified that essential services were active and responsive during testing.

Observing service status alongside performance metrics helped demonstrate how system services behave under normal operating conditions before applying load.

8. Bottleneck Identification

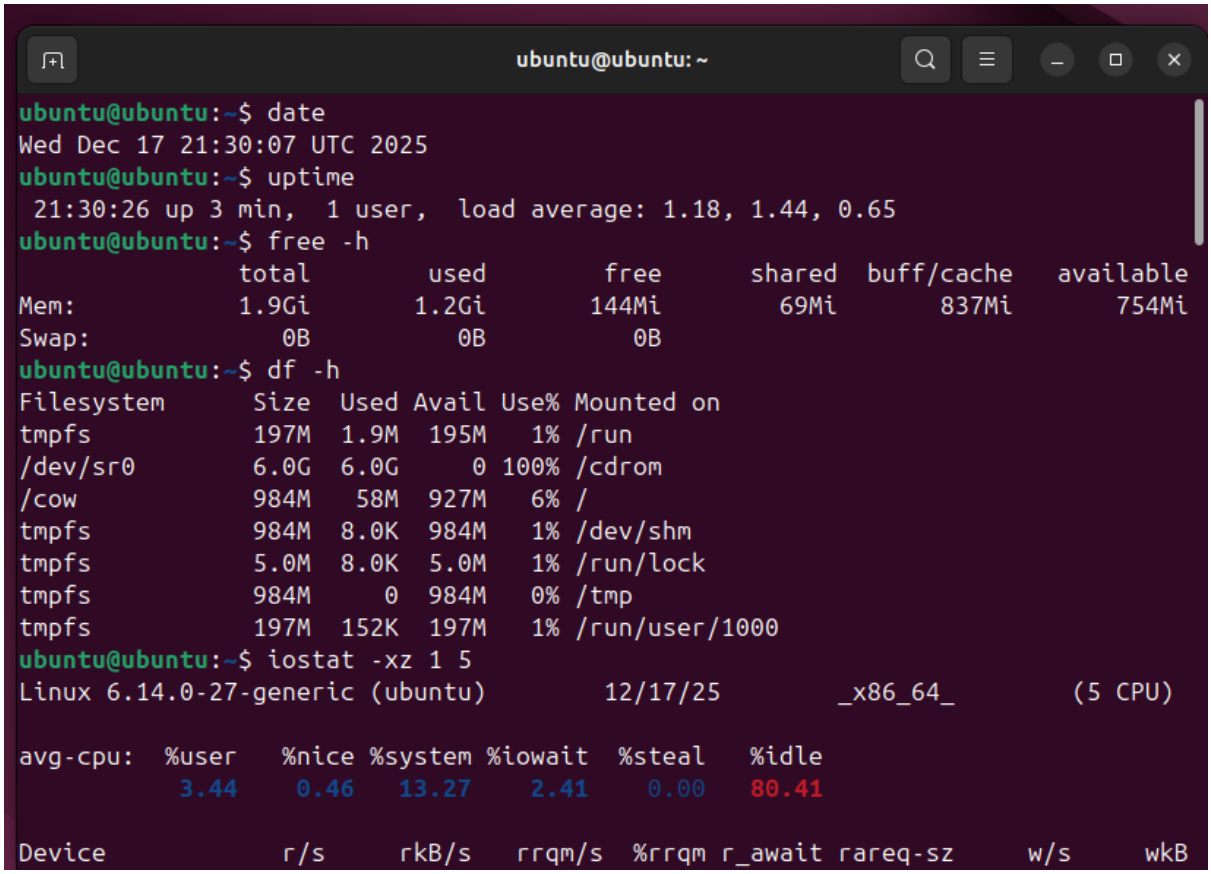
Based on the collected baseline data, no immediate bottlenecks were identified during idle operation. CPU idle percentages were high, disk I/O wait was low, and sufficient memory was available. This indicates that the system has adequate resources for handling additional workloads.

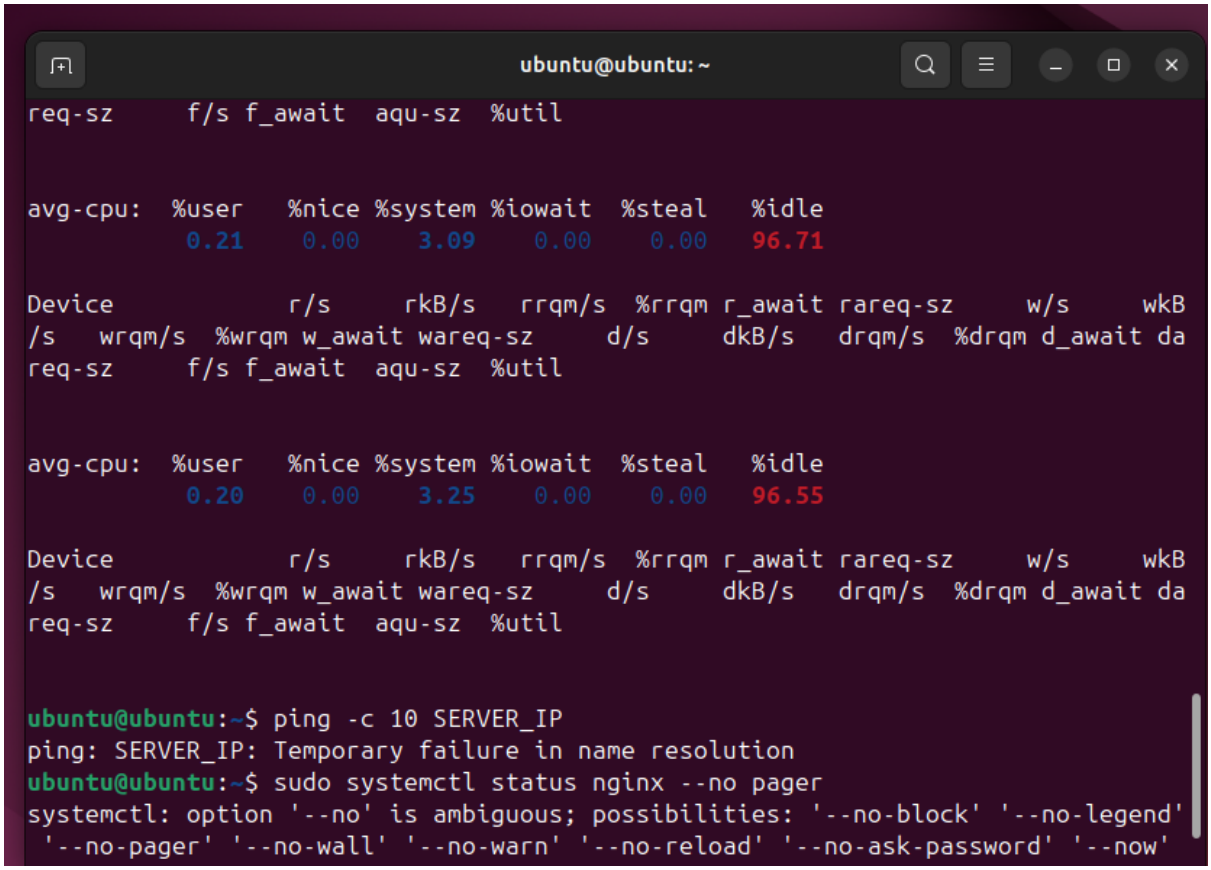
These observations establish a strong reference point for identifying bottlenecks during application load testing, where increases in CPU usage, memory pressure, disk wait time, or latency will indicate specific resource limitations.

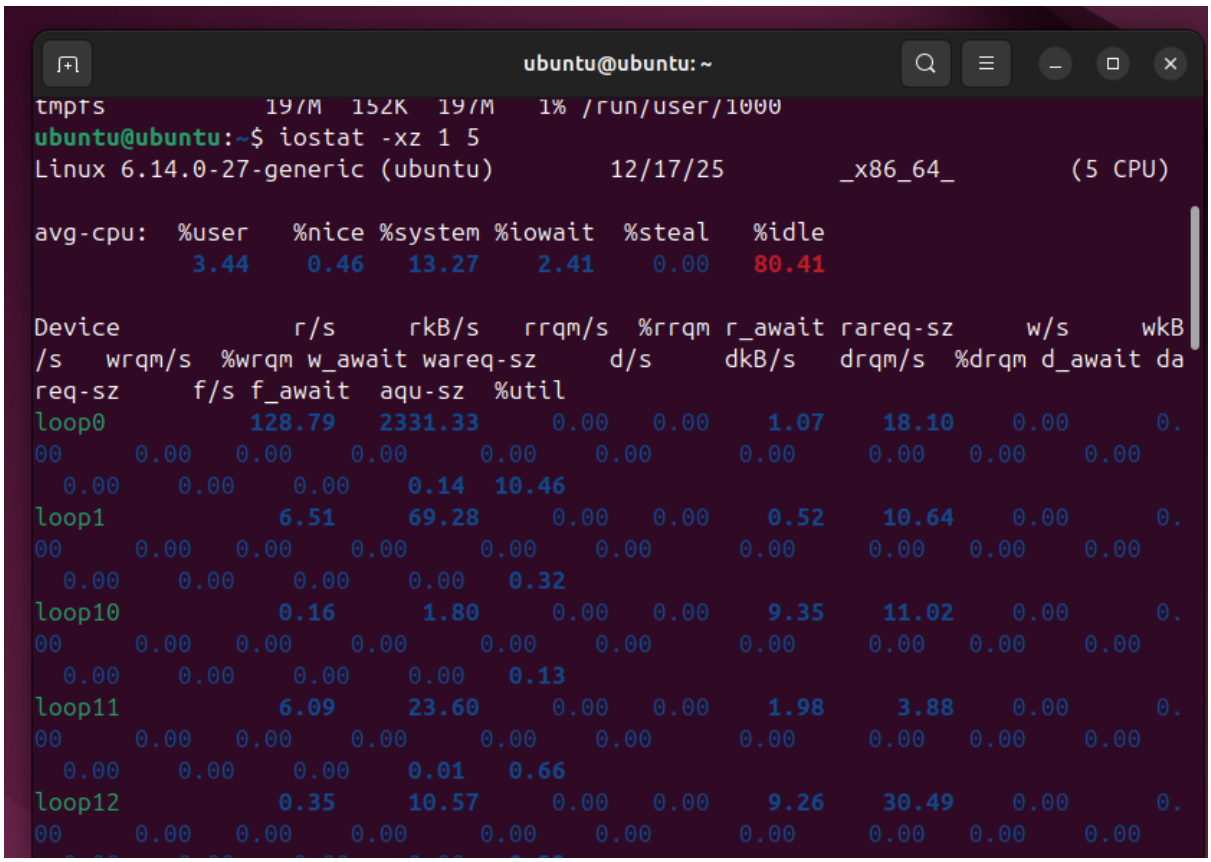
9. Reflection

Week 6 demonstrated how structured performance evaluation helps reveal operating system behaviour under different conditions. Collecting baseline metrics highlighted the importance of understanding normal system operation before applying workloads. Tools such as uptime, free, df, and iostat provided valuable insights into CPU utilisation, memory availability, and disk performance.

This week reinforced the need for accurate measurements, careful comparison, and troubleshooting during performance analysis. The collected data provides a solid foundation for further optimisation and analysis.


ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ date
Wed Dec 17 21:30:07 UTC 2025
ubuntu@ubuntu:~$ uptime
21:30:26 up 3 min, 1 user, load average: 1.18, 1.44, 0.65
ubuntu@ubuntu:~$ free -h
total used free shared buff/cache available
Mem: 1.9Gi 1.2Gi 144Mi 69Mi 837Mi 754Mi
Swap: 0B 0B 0B
ubuntu@ubuntu:~$ df -h
Filesystem Size Used Avail Use% Mounted on
tmpfs 197M 1.9M 195M 1% /run
/dev/sr0 6.0G 6.0G 0 100% /cdrom
/cow 984M 58M 927M 6% /
tmpfs 984M 8.0K 984M 1% /dev/shm
tmpfs 5.0M 8.0K 5.0M 1% /run/lock
tmpfs 984M 0 984M 0% /tmp
tmpfs 197M 152K 197M 1% /run/user/1000
ubuntu@ubuntu:~$ iostat -xz 1 5
Linux 6.14.0-27-generic (ubuntu) 12/17/25 _x86_64_ (5 CPU)
avg-cpu: %user %nice %system %iowait %steal %idle
3.44 0.46 13.27 2.41 0.00 80.41
Device r/s rkB/s rrqm/s %rrqm r_await rareq-sz w/s wkB


ubuntu@ubuntu: ~
req-sz f/s f_await aqu-sz %util
avg-cpu: %user %nice %system %iowait %steal %idle
0.21 0.00 3.09 0.00 0.00 96.71
Device r/s rkB/s rrqm/s %rrqm r_await rareq-sz w/s wkB
/s wrqm/s %wrqm w_await wareq-sz d/s dkB/s drqm/s %drqm d_await da
req-sz f/s f_await aqu-sz %util
avg-cpu: %user %nice %system %iowait %steal %idle
0.20 0.00 3.25 0.00 0.00 96.55
Device r/s rkB/s rrqm/s %rrqm r_await rareq-sz w/s wkB
/s wrqm/s %wrqm w_await wareq-sz d/s dkB/s drqm/s %drqm d_await da
req-sz f/s f_await aqu-sz %util
ubuntu@ubuntu:~$ ping -c 10 SERVER_IP
ping: SERVER_IP: Temporary failure in name resolution
ubuntu@ubuntu:~$ sudo systemctl status nginx --no pager
systemctl: option '--no' is ambiguous; possibilities: '--no-block' '--no-legend'
'--no-pager' '--no-wall' '--no-warn' '--no-reload' '--no-ask-password' '--now'


ubuntu@ubuntu: ~
tmpfs 197M 152K 197M 1% /run/user/1000
ubuntu@ubuntu:~$ iostat -xz 1 5
Linux 6.14.0-27-generic (ubuntu) 12/17/25 _x86_64_ (5 CPU)
avg-cpu: %user %nice %system %iowait %steal %idle
3.44 0.46 13.27 2.41 0.00 80.41
Device r/s rkB/s rrqm/s %rrqm r_await rareq-sz w/s wkB
/s wrqm/s %wrqm w_await wareq-sz d/s dkB/s drqm/s %drqm d_await da
req-sz f/s f_await aqu-sz %util
loop0 128.79 2331.33 0.00 0.00 1.07 18.10 0.00 0.
00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.14 10.46
loop1 6.51 69.28 0.00 0.00 0.52 10.64 0.00 0.
00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.32
loop10 0.16 1.80 0.00 0.00 9.35 11.02 0.00 0.
00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.13
loop11 6.09 23.60 0.00 0.00 1.98 3.88 0.00 0.
00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.01 0.66
loop12 0.35 10.57 0.00 0.00 9.26 30.49 0.00 0.
00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00