

Week 3 — Application Selection for Performance Testing

Overview

Week 3 focuses on the **selection and preparation of representative applications** for performance evaluation of the Linux operating system. The objective is to choose workloads that stress different subsystems of the OS, including CPU scheduling, memory management, disk I/O, and network performance. Selecting appropriate applications is essential to ensure that performance testing in later weeks produces meaningful, measurable, and comparable results.

All applications selected for this coursework are **command-line based**, lightweight, and suitable for execution on a headless Ubuntu Server environment.

Objectives

- Select applications representing diverse workload types
 - Justify application choices based on OS resource utilisation
 - Document installation procedures using SSH
 - Define expected resource usage profiles
 - Plan monitoring strategies for each application
-

Deliverables

- Application Selection Matrix
 - Installation documentation with exact commands
 - Expected resource usage profiles
 - Monitoring strategy for each workload
-

1. Application Selection Strategy

1.1 Selection Rationale

Applications were selected to ensure coverage of the **core operating system resource domains**:

- CPU-intensive workloads to evaluate scheduling and processor utilisation
- Memory-intensive workloads to analyse RAM allocation and pressure
- Disk I/O-intensive workloads to observe file system and storage behaviour

- Network-intensive workloads to measure latency and throughput

This approach enables systematic analysis of how the operating system behaves under different stress conditions.

2. Application Selection Matrix

Workload Type	Application	Purpose	Justification
CPU-intensive	stress	Generate CPU load	Tests CPU scheduling and utilisation
Memory-intensive	stress	Allocate and consume RAM	Evaluates memory management and swapping
Disk I/O-intensive	dd	Read/write performance testing	Measures disk throughput and I/O latency
Network-intensive	iperf3	Network throughput testing	Measures bandwidth and network performance
System monitoring	htop	Real-time resource monitoring	Provides live system visibility

These applications are widely used in industry and academic environments for benchmarking and system analysis.

3. Installation Documentation

All applications were installed on the Ubuntu Server **remotely via SSH** using the Advanced Package Tool (APT). The following commands were executed:

```
sudo apt update
```

```
sudo apt install stress iperf3 htop sysstat -y
```

- stress is used to generate CPU and memory load
- iperf3 is used for network performance testing
- htop provides interactive monitoring
- sysstat provides tools such as iostat and mpstat

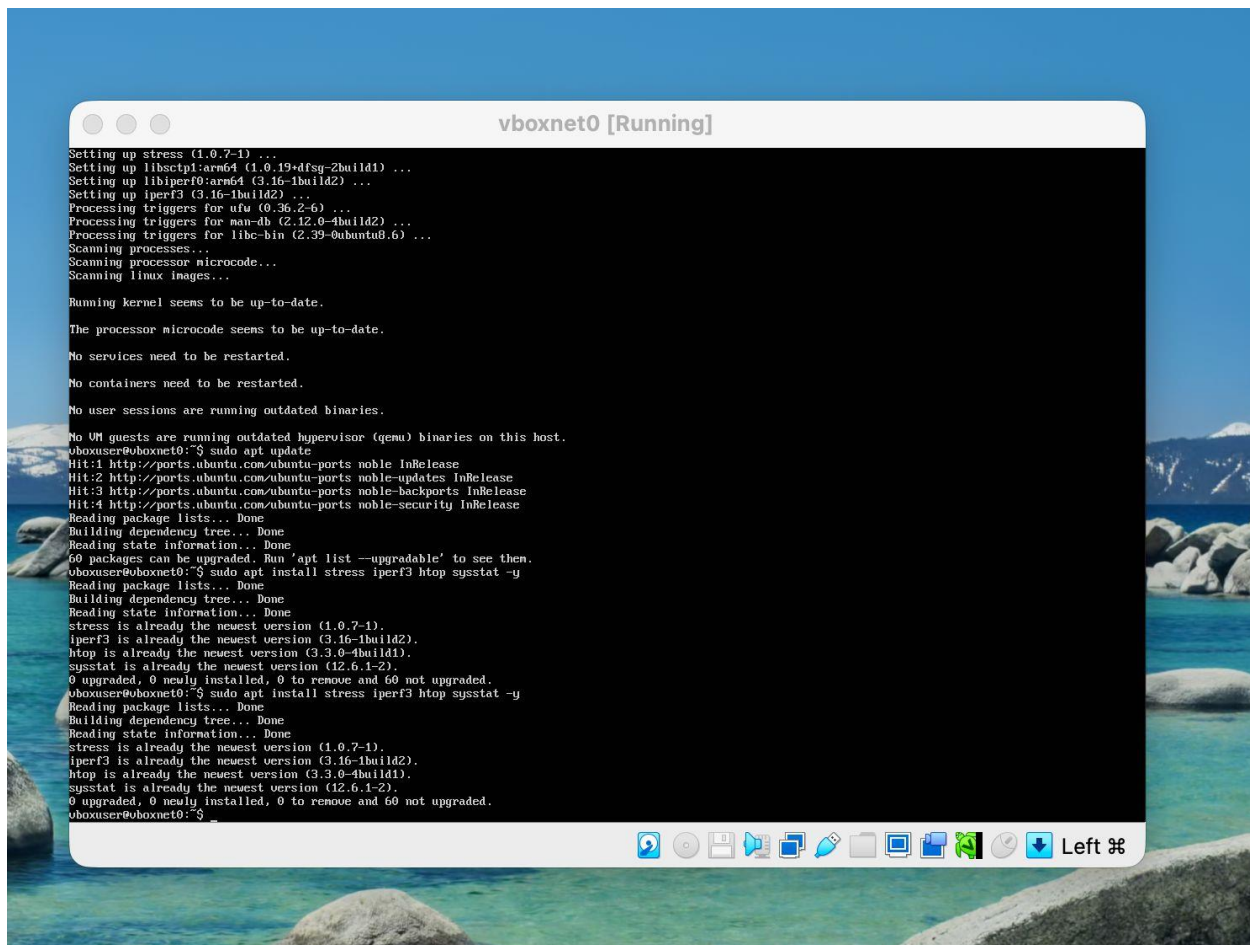


Figure W3-1: Installation of performance testing applications via SSH.

4. Expected Resource Usage Profiles

Before executing performance tests, expected behaviour for each application was documented to support later comparison and analysis.

4.1 CPU-Intensive Workload (stress)

- High CPU utilisation across multiple cores
- Minimal disk and network activity
- Increased system load average

4.2 Memory-Intensive Workload (stress)

- Progressive RAM consumption
- Potential swap usage under memory pressure
- Increased page fault activity

4.3 Disk I/O Workload (dd)

- Sustained read/write operations
- Increased disk throughput and I/O wait time
- Minimal CPU utilisation

4.4 Network-Intensive Workload (iperf3)

- Increased network bandwidth utilisation
- Observable changes in latency and throughput
- Low disk usage

Documenting expected behaviour enables identification of deviations and bottlenecks during actual testing.

5. Monitoring Strategy

5.1 Monitoring Tools and Metrics

Resource	Tools	Metrics Collected
CPU	top, htop, mpstat	CPU usage %, load average
Memory	free, vmstat	Used/free memory, swap
Disk I/O	iostat, df, dd	Read/write speed, I/O wait
Network	ping, iperf3	Latency, throughput

5.2 Monitoring Approach

- Monitoring commands will be executed **remotely from the workstation** via SSH
- Baseline metrics will be recorded before workload execution
- Metrics will be captured during active load
- Results will be logged for later visualisation and analysis

This structured approach ensures consistency and repeatability across all tests.

6. Preparation for Automation

The selected applications and monitoring tools were chosen to support later automation using shell scripts. In subsequent weeks: - Performance data will be collected automatically - Output will be redirected to CSV files - Scripts will be executed remotely to reflect professional administration practices

7. Reflection

Key Outcomes

- Selected representative workloads covering all major OS subsystems
- Installed industry-standard benchmarking and monitoring tools
- Developed expectations for system behaviour under load
- Prepared the environment for structured performance evaluation

Anticipated Challenges

- Ensuring consistent test conditions across runs
- Interpreting performance data accurately
- Balancing test duration with system stability

Learning Outcomes Achieved

- ✓ Understanding how different workloads affect OS behaviour
- ✓ Applying resource-based analysis to application selection
- ✓ Preparing structured performance evaluation methods

This week supports **Learning Outcome 4** through command-line tool usage and **Learning Outcome 5** by analysing operating system behaviour under varied workloads.

References

- Linux Stress Tool Manual. Available: <https://manpages.ubuntu.com/manpages/jammy/man1/stress.1.html> (Accessed: 2025)
- iperf3 Documentation. Available: <https://iperf.fr/iperf-doc.php> (Accessed: 2025)
- sysstat Documentation. Available: <https://github.com/sysstat/sysstat> (Accessed: 2025)