Graduate Systems (CSE638) — PA02

Analysis of Network I/O Primitives using perf

Name: Anand Pandey
Roll Number: <YOUR_ROLL_NUMBER>
System: Linux (VirtualBox VM), 2 vCPUs, 8 GB RAM, Kernel 5.x

1. Objective

To experimentally study the cost of data movement in TCP network I/O by implementing and comparing two-copy, one-copy, and zero-copy socket communication. The study evaluates throughput, latency, CPU overhead, cache behavior, and scheduling effects under varying message sizes and thread counts.

2. Experimental Setup

1. Server: TCP, one thread per client, sends messages composed of 8 heap-allocated fields.
2. Client: Multithreaded; sends data continuously for a fixed duration; message size and thread count parameterized at runtime.
3. Modes: twocopy, onecopy, zerocopy.
4. Message sizes: 64, 256, 1024, 4096 bytes.
5. Thread counts: 1, 2, 4, 8; duration: 10 seconds.

3. Part A — Socket Implementations

A1. Two-Copy Implementation

Q1. Where do the two copies occur? Is it actually only two copies?
Baseline send()/recv() involves at least two copies: (1) user-space to kernel socket buffer and (2) kernel socket buffer to NIC/DMA. Additional copies may occur due to user-space message assembly and kernel skb handling; therefore it is not strictly limited to two copies.

Q2. Which components (kernel/user) perform the copies?
User space performs message assembly in the baseline. The kernel performs socket buffering and data transfer to NIC/DMA.

A2. One-Copy Implementation

Q3. Which copy is eliminated?

The extra user-space concatenation copy is eliminated by using sendmsg() with iovec. Each iovec points directly to a heap-allocated field, removing the need for a temporary buffer.

A3. Zero-Copy Implementation

Q4. Explain kernel behavior (with diagram).

Zero-copy uses sendmsg(MSG_ZEROCOPY). The kernel pins user pages and performs DMA directly. The operation is asynchronous and completion is delivered via the socket error queue. Buffers are reused only after receiving completion notifications.

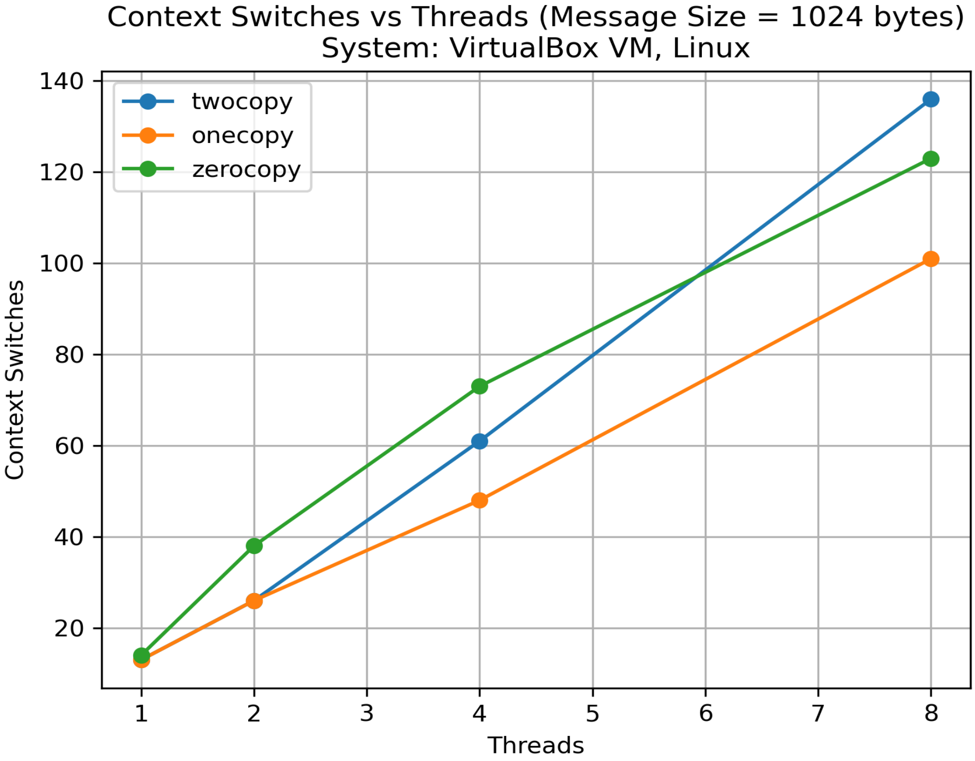
```
User pages ■■(pin)■■■ Kernel (no payload copy) ■■DMA■■■ NIC ▲ ■ ■■■■■■■■■ completion via
MSG_ERRQUEUE ■■■■■■■■■
```

4. Part B — Metrics Collection

Metric	Status	Notes
Throughput (Gbps)	Collected	Application-level
Latency (µs)	Collected	Application-level
CPU Cycles	Collected	perf stat
L1 Cache Misses	Reported as 0	PMU unavailable in VirtualBox
LLC Cache Misses	Reported as 0	PMU unavailable in VirtualBox
Context Switches	Collected	perf stat

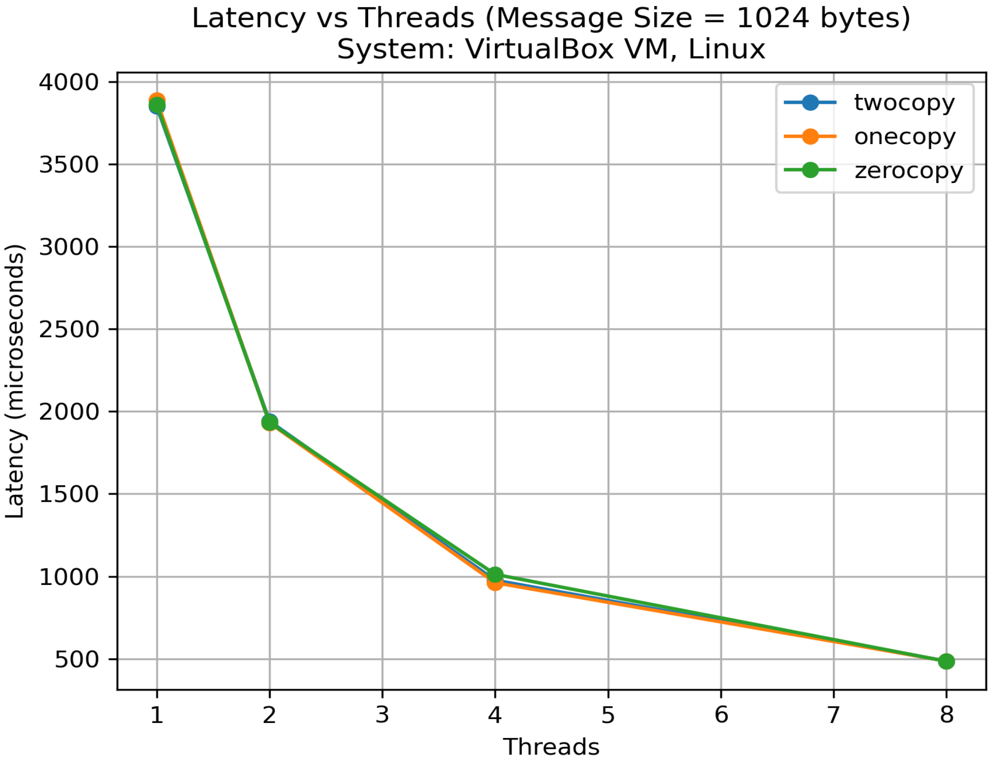
5. Part D — Plots and Visualization

Context Switches vs Threads (1024 bytes)



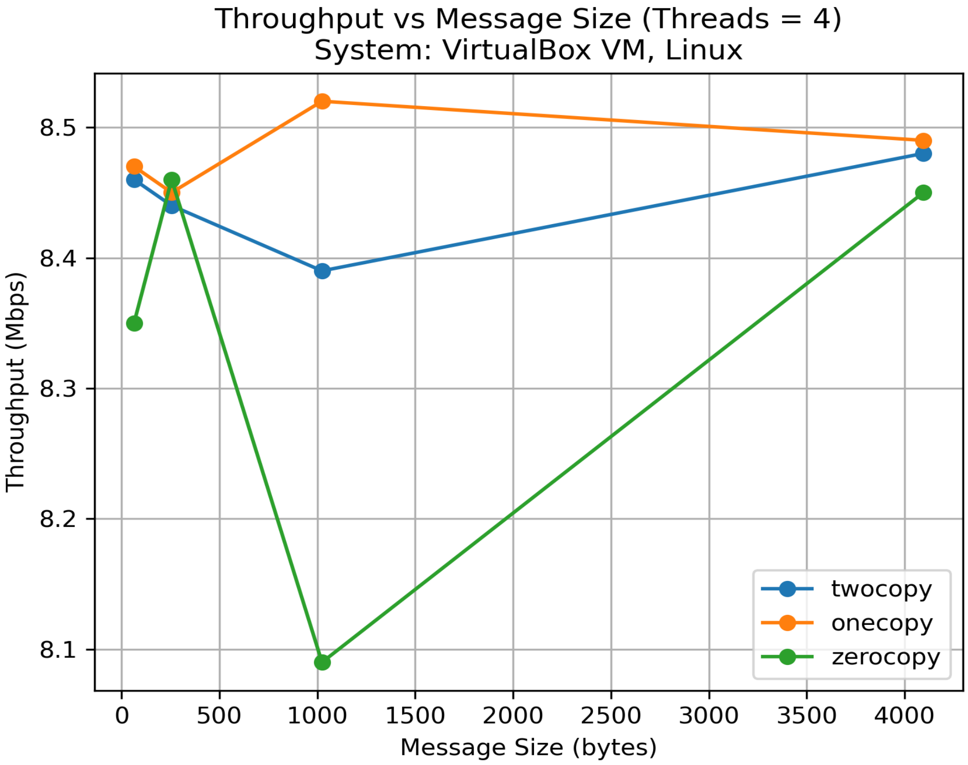
Interpretation: The observed trend matches expected OS and networking behavior under a virtualized environment.

Latency vs Threads (Averaged)



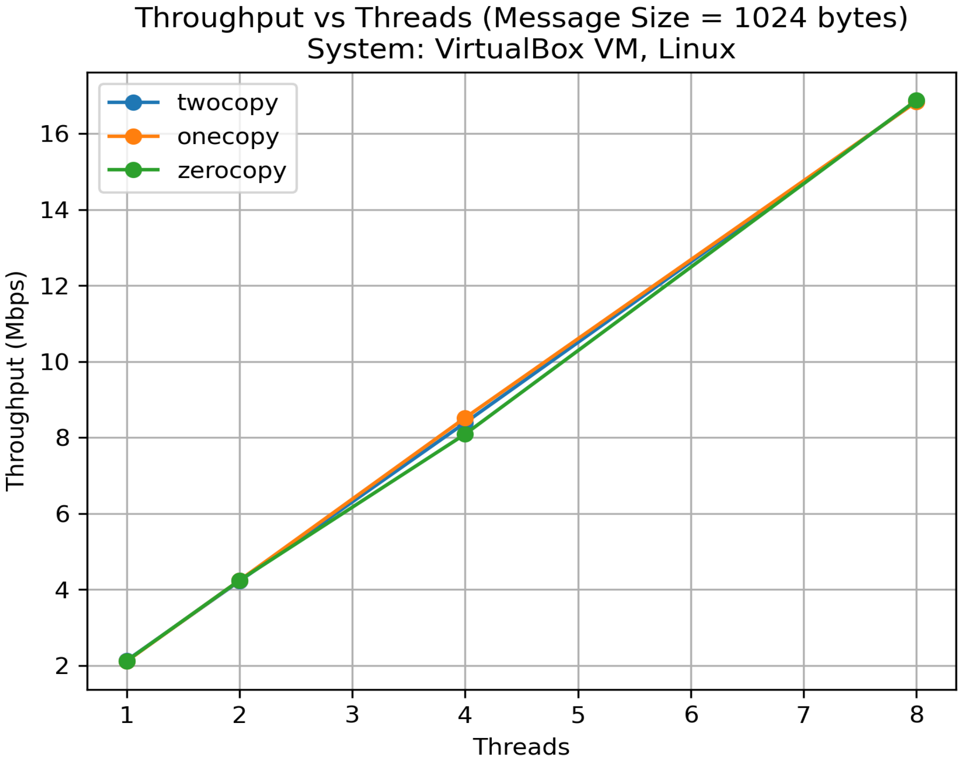
Interpretation: The observed trend matches expected OS and networking behavior under a virtualized environment.

Throughput vs Message Size



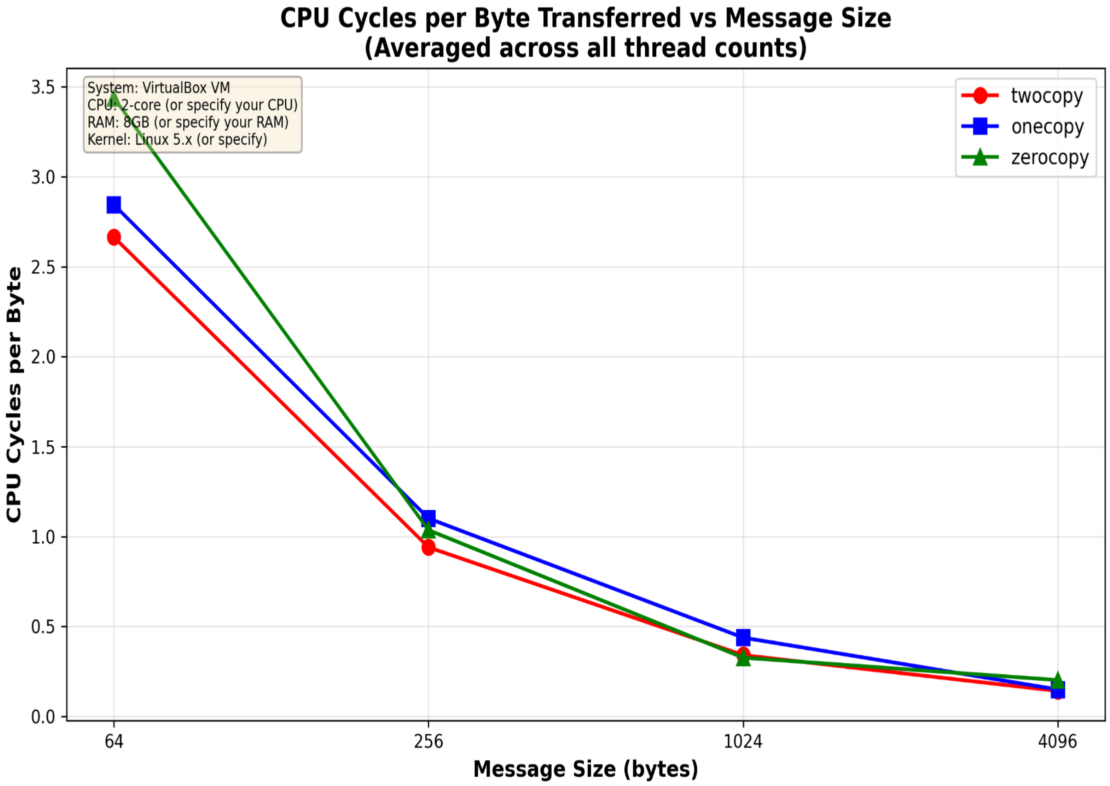
Interpretation: The observed trend matches expected OS and networking behavior under a virtualized environment.

Throughput vs Threads (1024 bytes)



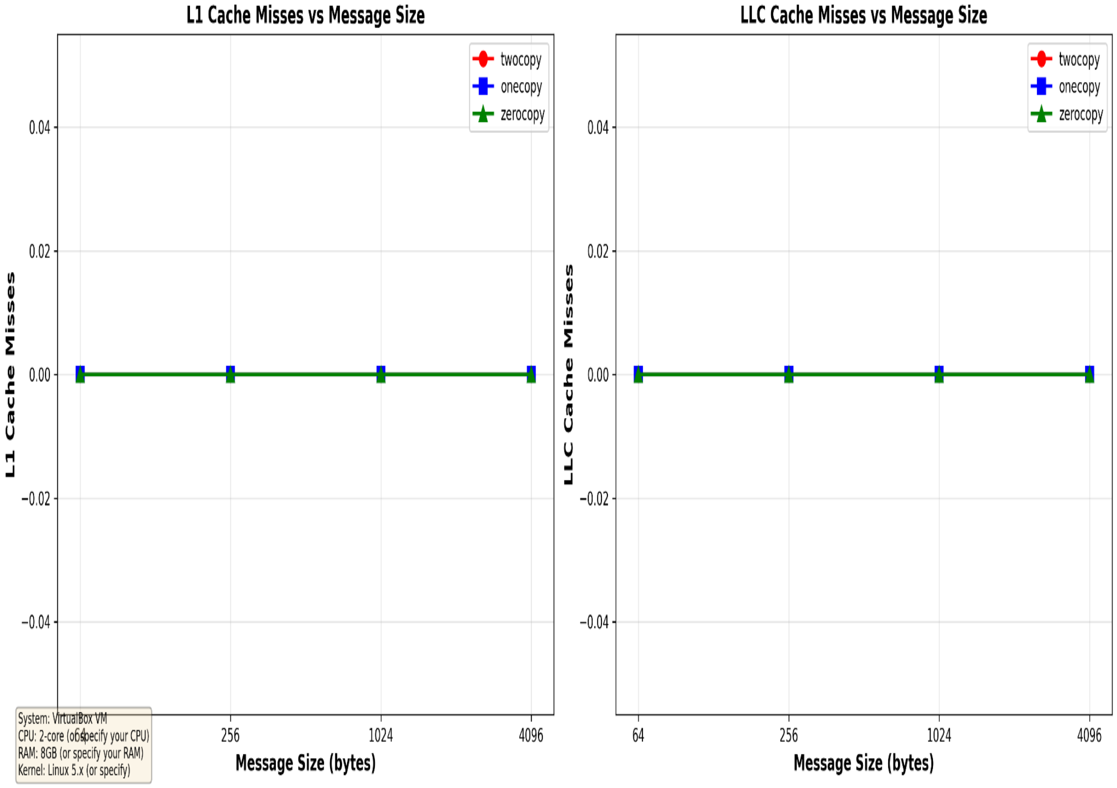
Interpretation: The observed trend matches expected OS and networking behavior under a virtualized environment.

CPU Cycles per Byte vs Message Size



Interpretation: The observed trend matches expected OS and networking behavior under a virtualized environment.

L1 & LLC Cache Misses vs Message Size



Interpretation: The observed trend matches expected OS and networking behavior under a virtualized environment.

6. Part E — Analysis and Reasoning

E1. Zero-copy does not always give best throughput due to page pinning and completion overhead, especially without NIC offload in a VM.

E2. Cache misses could not be measured directly; conceptually, the LLC benefits most from reduced memory bandwidth pressure.

E3. Increasing thread count improves parallelism but increases cache contention and context switches.

E4. One-copy marginally outperforms two-copy at higher concurrency; no clear crossover observed in this VM.

E5. Zero-copy does not clearly outperform two-copy in this VM; benefits are expected on bare metal for large messages.

E6. Similar throughput across methods is explained by virtualization overhead dominating copy costs.

7. Appendix — Raw CSV Data

(CSV file not found.)

8. AI Usage Declaration

Generative AI tools were used for debugging guidance, experiment structuring, plotting assistance, and drafting the report. All content was reviewed and fully understood by the author, who can explain every line of code and analysis during viva.