Revealing Hidden Contention through Disk I/O Analysis in Multi-Tenant Containers

Authors : 1st Mi-Yeon Kim, 2nd Sang-Hoon Choi, and Ki-Woong Park† | Number : 150

WISA 2025

1. Introduction

- In cloud multi-tenant environments where physical resources are shared, performance degradation caused by noisy neighbors becomes a problem.
- Noisy neighbor problem can affect the overall performance of the system; attackers can exploit this characteristic to execute denial of service(DoS) attacks.
- In particular, containers share the host's kernel, are less isolated than VMs and attack surface is wider.
- This papper analyzes how the performance of other container disks sharing physical resources changes when an attacker intentionally creates a large file in LXC, run, and gVisor containers, and experimentally proves that disk performance degradation can also occur through normal behavior.

2. Related Work

- Li et al. evaluated disk I/O using sequential read/write latency and random read/write latency in an environment based on application containers such as Docker, gVisor, Kata, and Qemu.
- Volpertet et al. measured CPU, disk, memory, and network performance in gVisor, Kata, and podman environments. However, disk performance results for gVisor were excluded. They explained that this was because gVisor does not support the functionality required for the experiment, which is to directly connect block devices to the container.

3. Approach

- We want to know if an innocuous-sounding activity, writing large files, can create inter-container disk interference.
- **Baseline.** As shown in Fig. 1(a), four containers (A, B, C, and D) were launched, of which containers A, B, and C were idle in the baseline measurement, with disk performance monitored in container D.
- **Scenario.** As shown in Fig. 1(b), containers A, B and C created 32GB and 128GB files, respectively, after which the disk performance of containers was monitored.
- **The configuration for test.** LXC, runc (with Docker), and runsc (with Docker) containers(were constrained to a single CPU core with 1GB of memory)
- **File Size.** To study the effects of caching, we use 32GB(within memory size) and 128GB(significantly exceeding memory capacity)

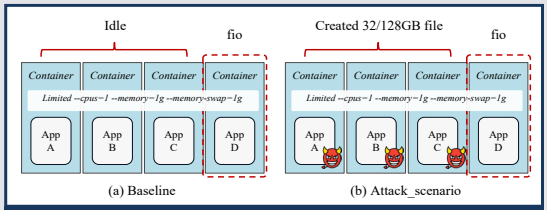

Fig.1. Design of an Attack Scenario to Evaluate Noisy Neighbor Effects.

4. Evaluation

- Fig. 2 shows Read/Write IOPS throughput for a selection of container runtimes.
- For the most part, LXC and runc showed similar qualities of performance.
- For the 32GB case, disk performance dropped by more than 50% compared to baseline.
- For 128GB, a drop of approximately 99.5% was seen.
- By contrast, for the baseline disk throughput observed for gVisor, this was only some 14% of what was seen for LXC and runc.

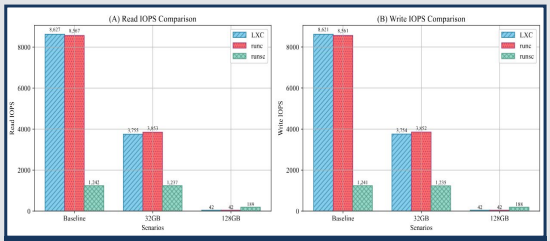

Fig.2. Comparison of Read/Write IOPS Throughput Across Container Runtimes.

- Fig. 3 Read/Write latency for each of the container runtimes. As with throughput, LXC and runc exhibited comparable latency behavior.
- For 32GB, latency increased by about 2x from the baseline, for 128GB, latency increased by over 200x.
- For 32GB case, latency remained very close to that of the baseline, but for 128GB case, read latency surged to 96.8ms, write latency to 72.5ms, about 6x increase.
- This high differential in read/write latency with runsc is because of gVisor's design approach of intercepting and emulating system calls in user space. For a read, due to interception of system calls, there is a lot of overhead, taking a much larger latency hit.

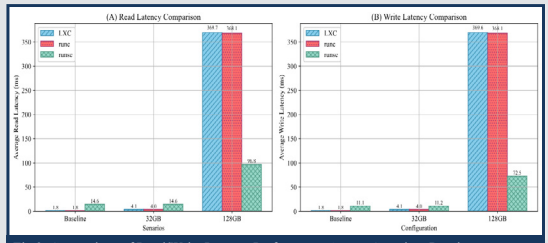

Fig.3. Comparison of Read/Write Latency Performance Across Container Runtimes.

5. Conclusion

- We analyzed the noisy neighbor problem for multi-tenant settings, can induce performance interference using regular file creation operations.
- We used LXC, runc, and runsc containers, each allocated a single CPU core and 1GB of memory.
- We saw severe performance degradation in both 32GB and 128GB settings in LXC and runc.
- runsc (gVisor) only exhibited baseline-level performance in the 32GB test but showed an 85% loss of IOPS throughput in the 128GB test, when the file size was larger than available memory.
- Even secure runtimes such as gVisor are not free from it in memory-intensive workload scenarios.
- Our future work is towards offering superior performance isolation for multi-tenant containers.

Acknowledgments

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) (2022-11220701, 60%), and the National Research Foundation of Korea (NRF) grant funded by the Korean Government (Project No. RS2023-00208460, 40%).