Analysis and Optimization of Network I/O Tax in Confidential Virtual Machines

Paper summary:

The paper aims to optimize I/O virtualization costs in the confidential VM (CVM) architecture. The authors identify three main overhead sources in CVM-IO: the bounce buffer mechanism and packets' processing. The paper proposes to bypass the latters by introducing Bitfrost, an I/O design relying on three main techniques: zeo-copy I/O NUMA (to eliminate payload bouncing), one-time trusted read (to protect the guest from *time of check to time of use* attacks), and pre-receiver packet reassembling (to reduce the number of packets' processing).

Strengths:

- Clear challenges identification for design is a good practice
- Good background section on PV IO in CVMs architectures

Weaknesses:

- No motivation section to present evidence of the CVM tax. Section 3 should be moved to a *Motivation* Section
- No state-of-the-art comparison

Comments to the authors:

- Globally, doesn't Intel SR-IOV provide complete zero-copy packet processing and data delivery to VMs? As proposed by J. Hwang et al. at NSDI'14?
- In Section 2.1 of the Background, what are the private and shared memory types in Figure 1? Can you highlight this in the Figure?
- The performance analysis in Section 3 is more a motivation analysis than a "first thorough" analysis because apart from the unique breakdown figure (Figure 3) presented in this section, some in deep CPU/IO/Storage analysis on traditional and CVM platforms (e.g., NxtGen, Azure Cloud, AWS) already exist such as this Cloud Performance Analysis Report from 2018, or this IPDPS paper from 2021.
- In Section 3, this sentence is unclear to me: *For fair performance comparison, CVM's baseline is the vanilla AMD traditional VM, while CVM+PI's baseline is the vanilla Intel traditional VM.* How can the PI's baseline be the traditional VM? How does this time measurement switching be considered fair?
- Observation O3 in Section 4.3 needs to be motivated by concrete results. When you argue that *CPUs running I/O backends have less work to do and thus have plenty of free CPU resources*, are you implying that running the VM's backend drivers is the only dedicated task of these host CPUs? Is this plausible on a Real Platform?
- The authors should also consider evaluating the impact of ZCIO NUMA's memory initialization on the boot time, as the latter is an important metric in cloud environments.
- Why isn't end-to-end encryption also applicable for packet headers (to further avoid the bounce buffer mechanism on the whole path)?
- Does the ZCIO NUMA need to be initialized at the host kernel/hypervisor boot? If not, why not implement the related modifications as a kernel module to avoid modifying the host kernel? 800+

LOCs is, however, not a negligible computing base.
- What questions do your evaluations answer? How do you breakdown your implementation improvement or overhead? For example, the ZCIO NUMA's memory initialization cost, the latency induced by packets' caching, the overhead of packet pre-processing on the host CPUs, etc.

Writting Quality

Well-written

Merit

Reject