To PRI or Not To PRI, That's the question

Yun Wang[1*], Liang Chen[2*], Jie Ji[2], Xianting Tian[2], Ben Luo[2], Zhixiang Wei[1],
Zhibai Huang[1], Kailiang Xu[1], Kaihuan Peng[2], Kaijie Guo[2], Ning Luo[2],
Guangjian Wang[2], Shengdong Dai[2], Yibin Shen[2], Jiesheng Wu[2], Zhengwei Qi[1]

1 上海交通大學 SHANGHAI JIAO TONG UNIVERSITY

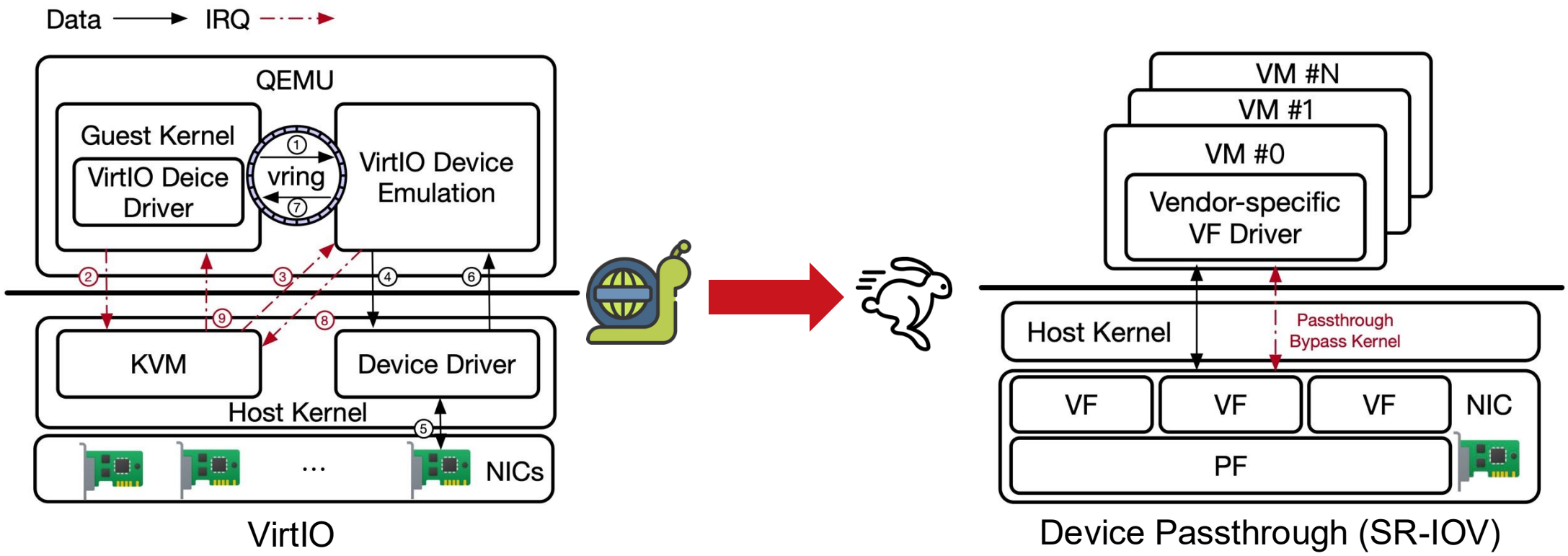
2 Alibaba Cloud

* Equal Contribution

Motivation

- Server memory is **expensive**.
- Cloud providers improve resource usage by taking back **unused/cold** memory pages.
- I/O device passthrough **makes it harder** to dynamically reclaim these resources.

Evolution in I/O Virtualization



I/O Devices Passthrough

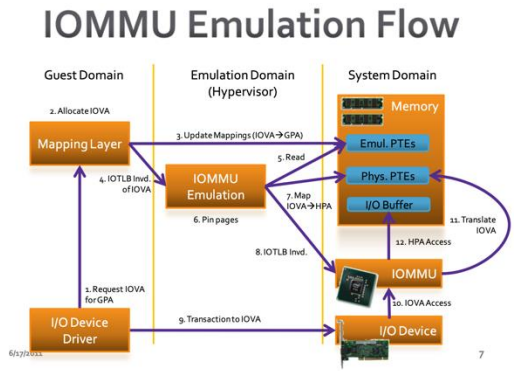
The Goal: Near-Native Performance

The Problem: I/O Devices Can't Handle Page Faults

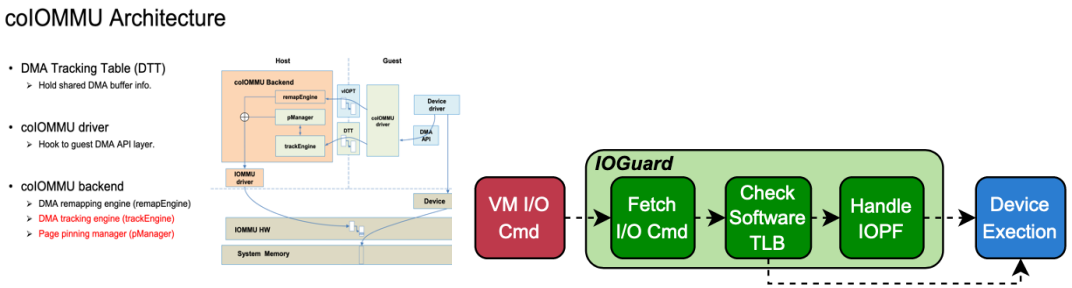
The "Official" Fix: The Page Request Interface (PRI)

Existing Solutions

Software



vIOMMU



coIOMMU

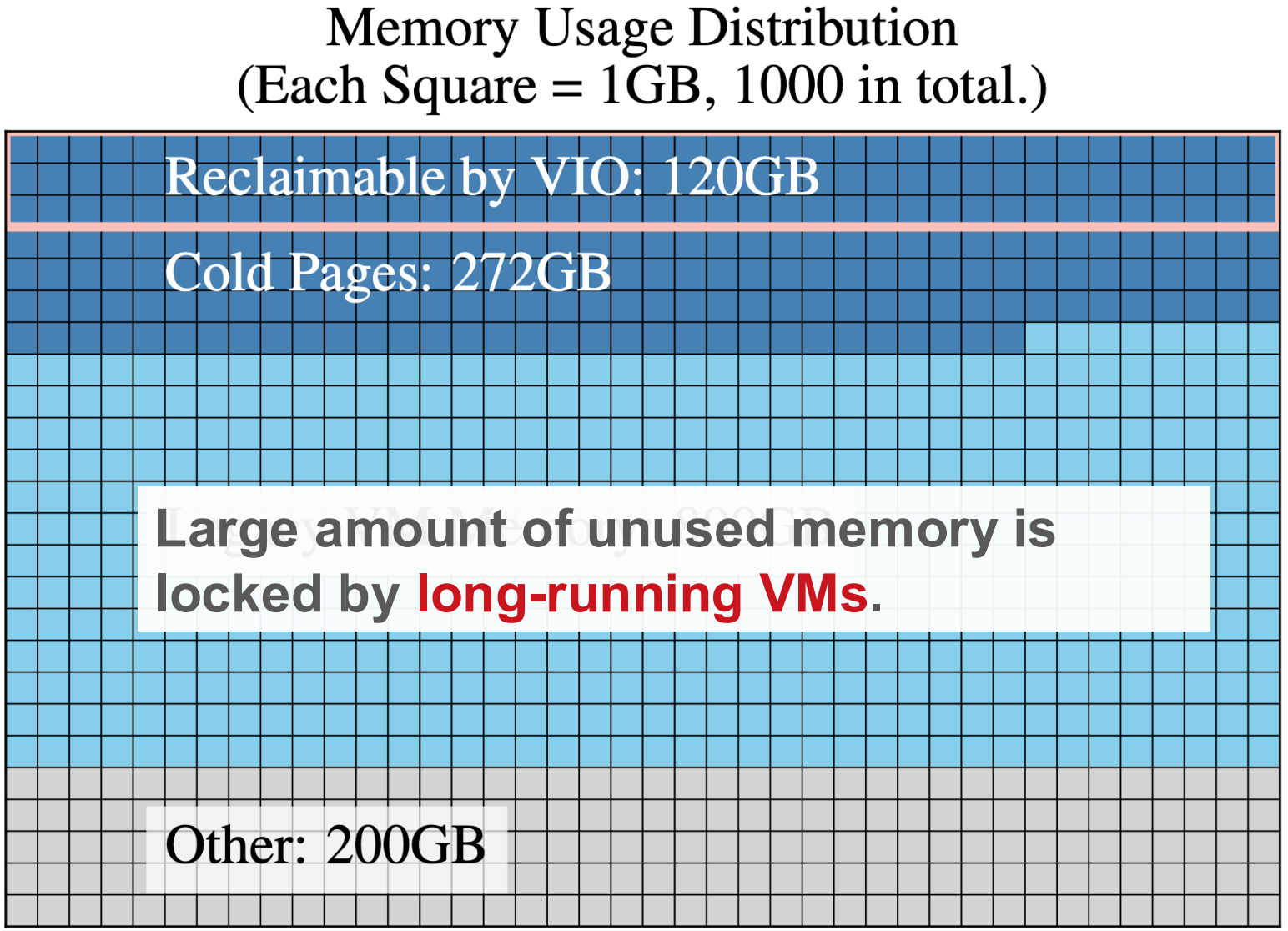
IOGuard

Hardware

On-Demand Paging

VPRI

Existing Solutions Fall Shorts



Existing Solutions Fall Shorts

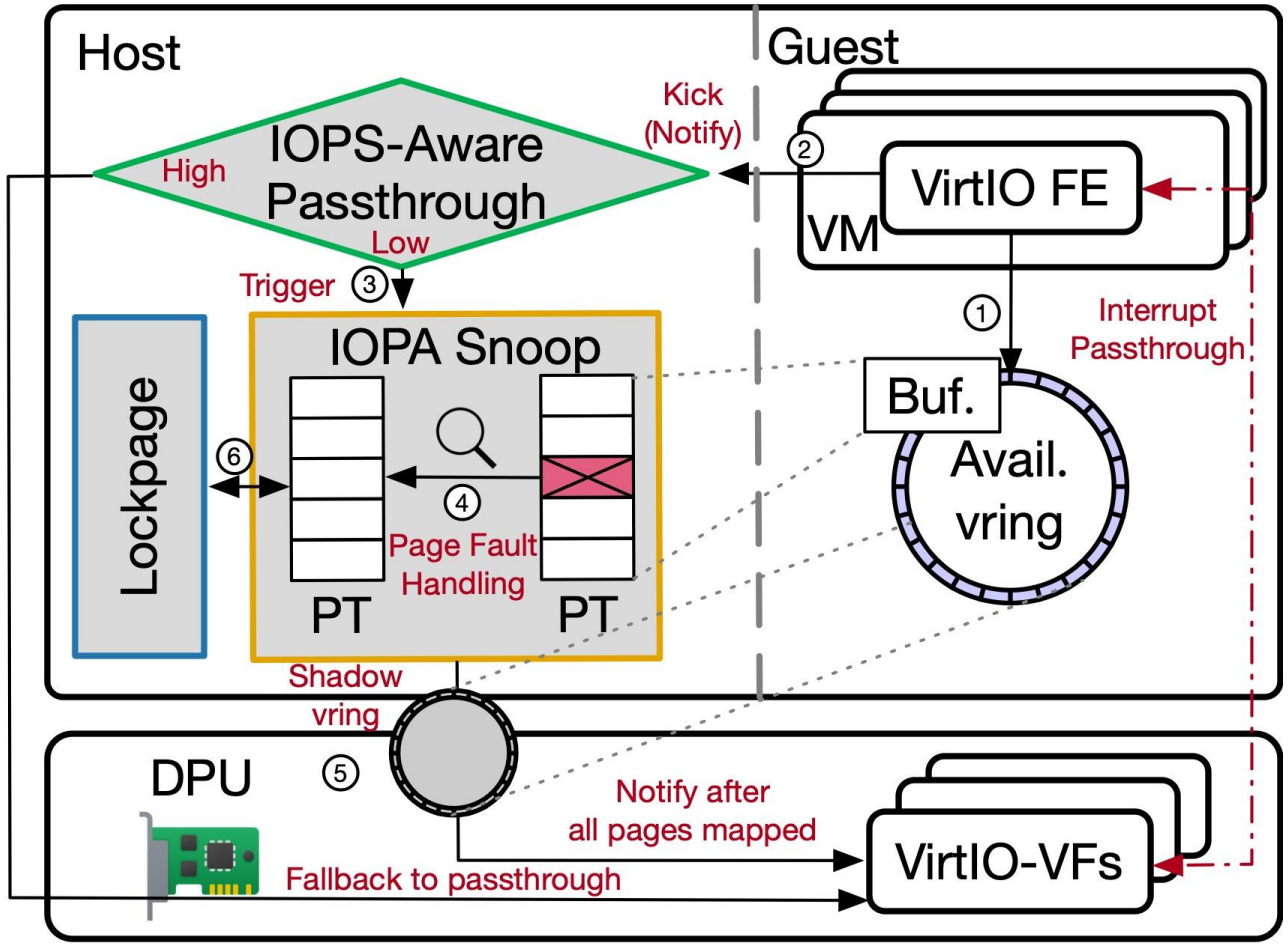
	Software	Hardware	Ours
	vIOMMU[ATC'11] coIOMMU[ATC'20] IOGuard[APSys'24]	ODP[ASPLOS'17] VPRI[SOSP'24]	**VIO**
Hardware Requirement	✅General Hardware	❌Proprietary Hardware	✅General VirtIO
Guest Modifications	❌Guest Modification	✅No Modification	✅No Modification
Low CPU Overhead	⚠Non-negligible	✅Minimal	✅Minimal

Limitation: Adoption for the vast number of existing, long-running legacy VMs in production environments.

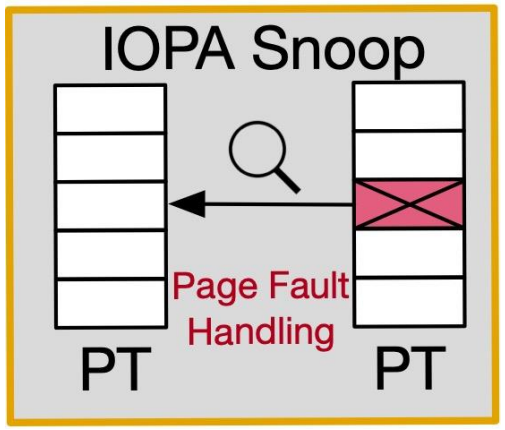
Can we *non-intrusively* handle IOPFs from legacy VMs on *existing hardware*, and with *minimal performance overhead*?

VIO Overview

- IOPA-Snoop
- Elastic Passthrough
- Lockpage



Challenges



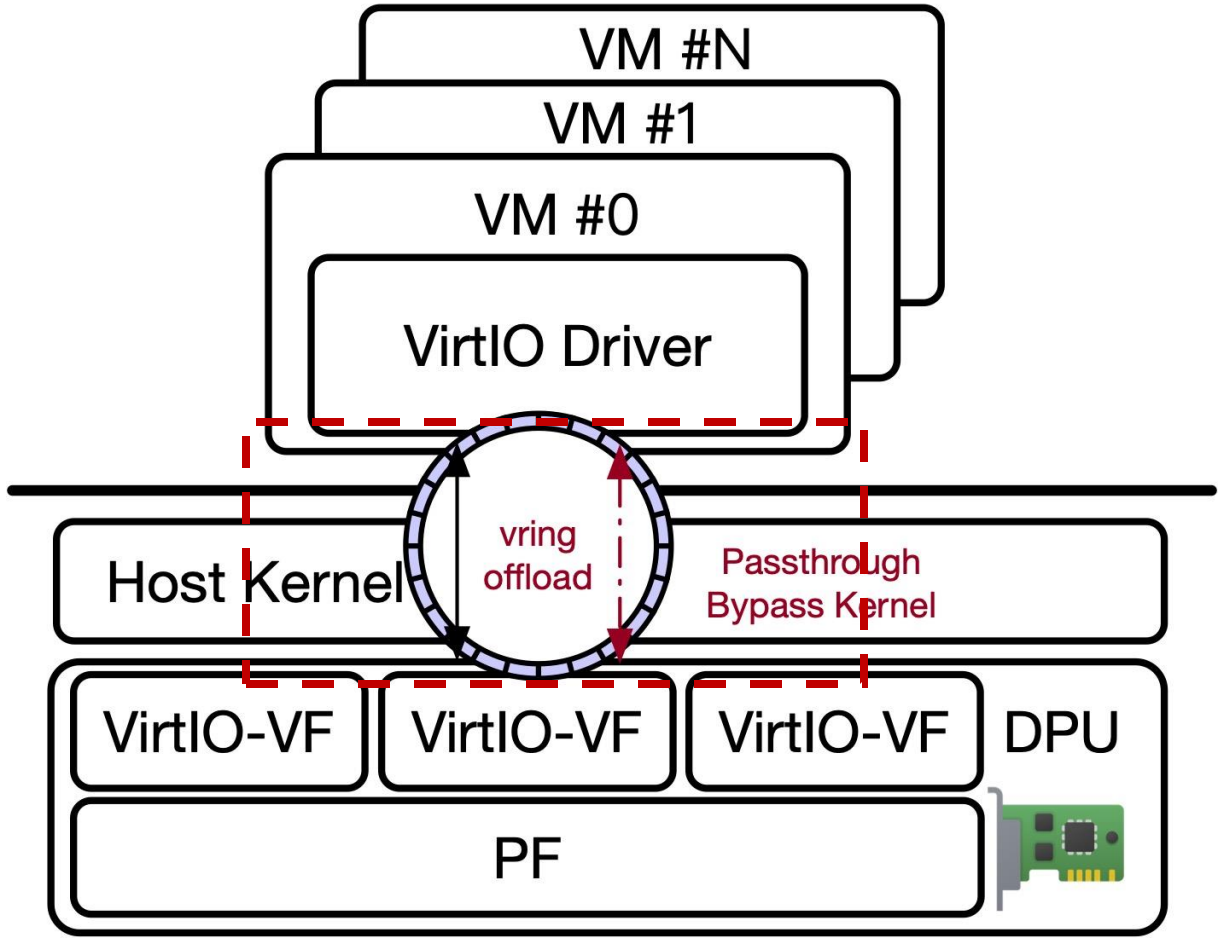
Page Fault Detection

Optimal Performance

IOPFs Reduction

Loose I/O Interposition

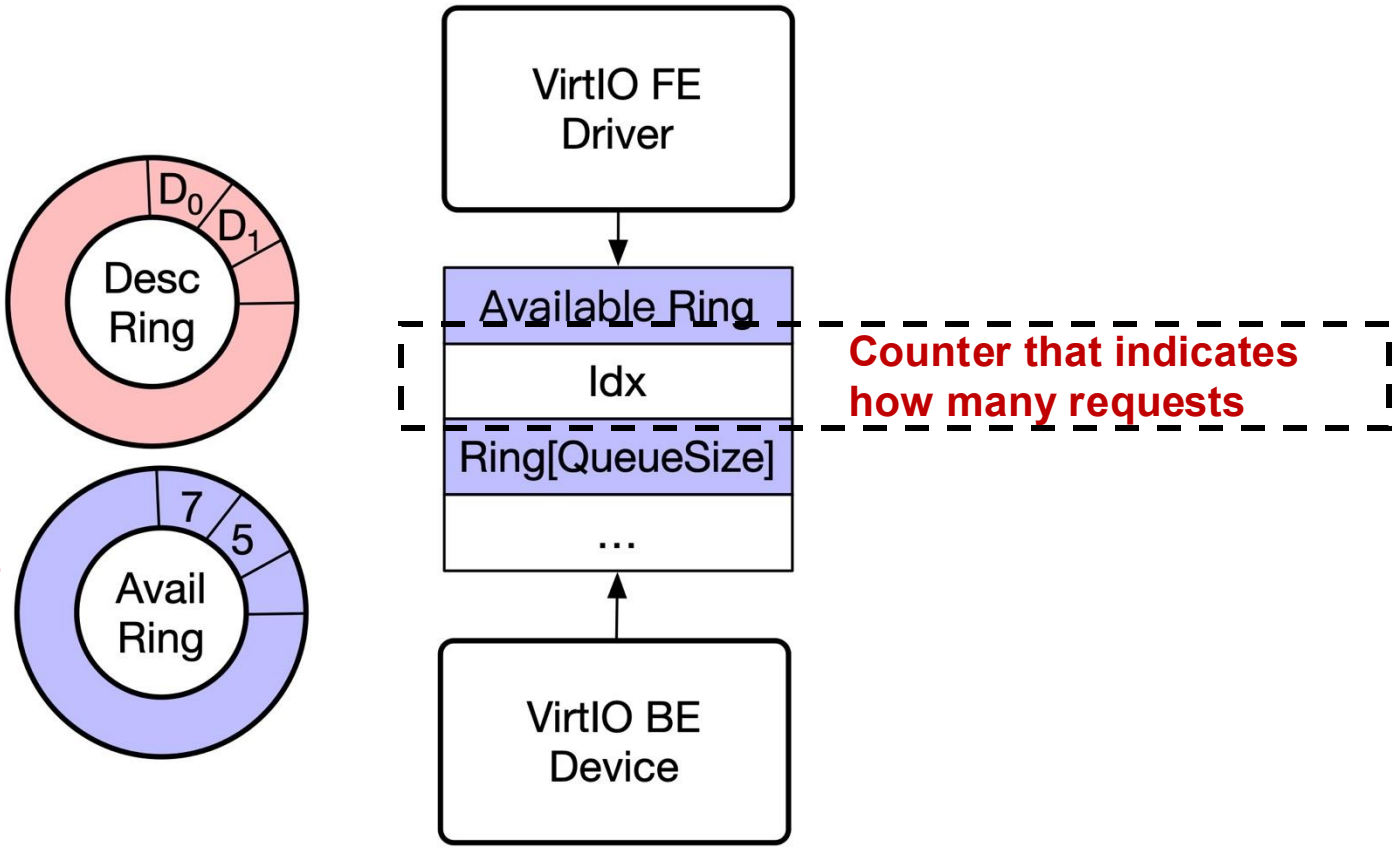
Root Cause: I/O operations **bypass** the hypervisor.



IOPA-Snoop

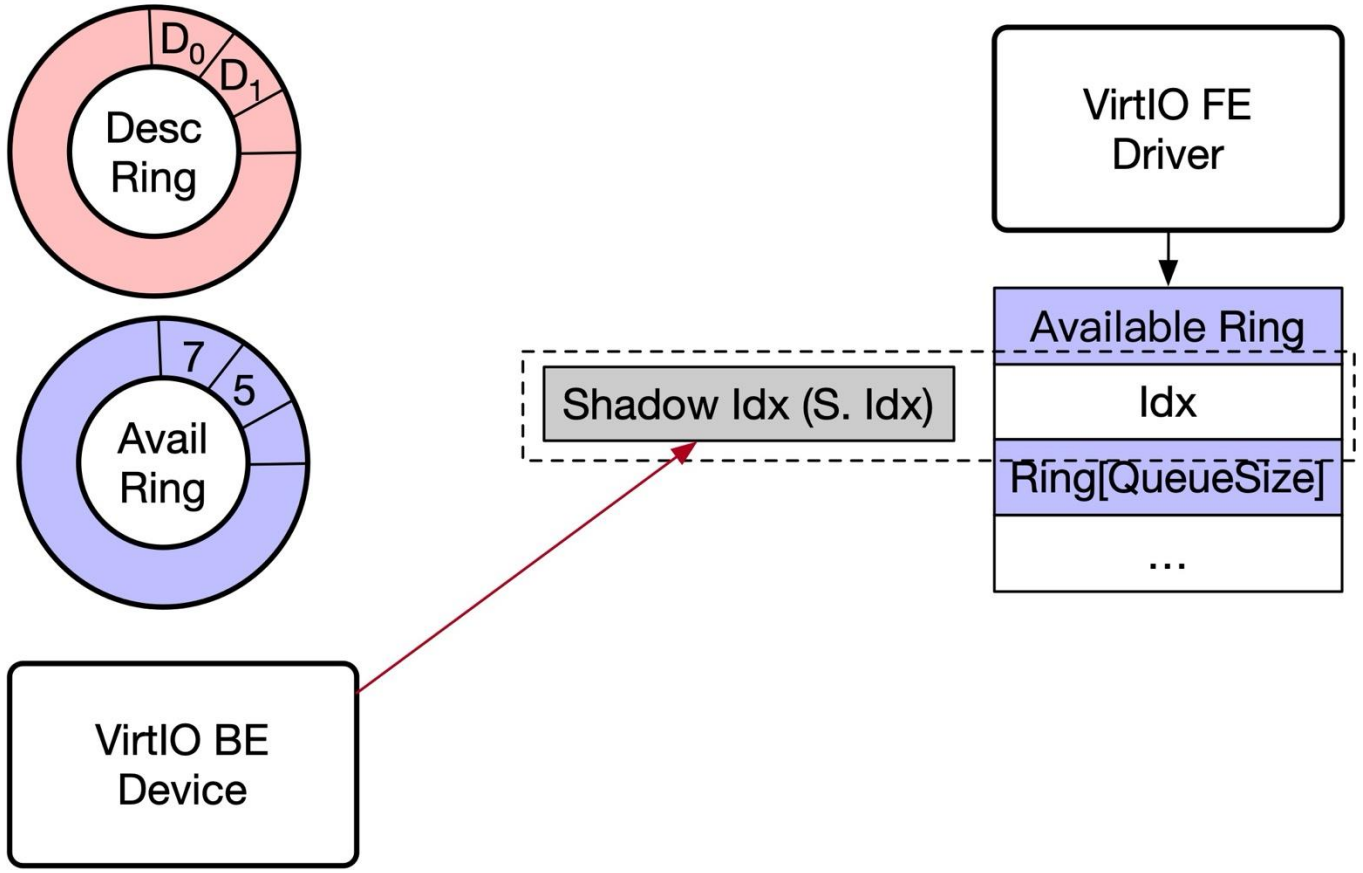
Contains I/O request descriptors

Places descriptor indices to submit requests



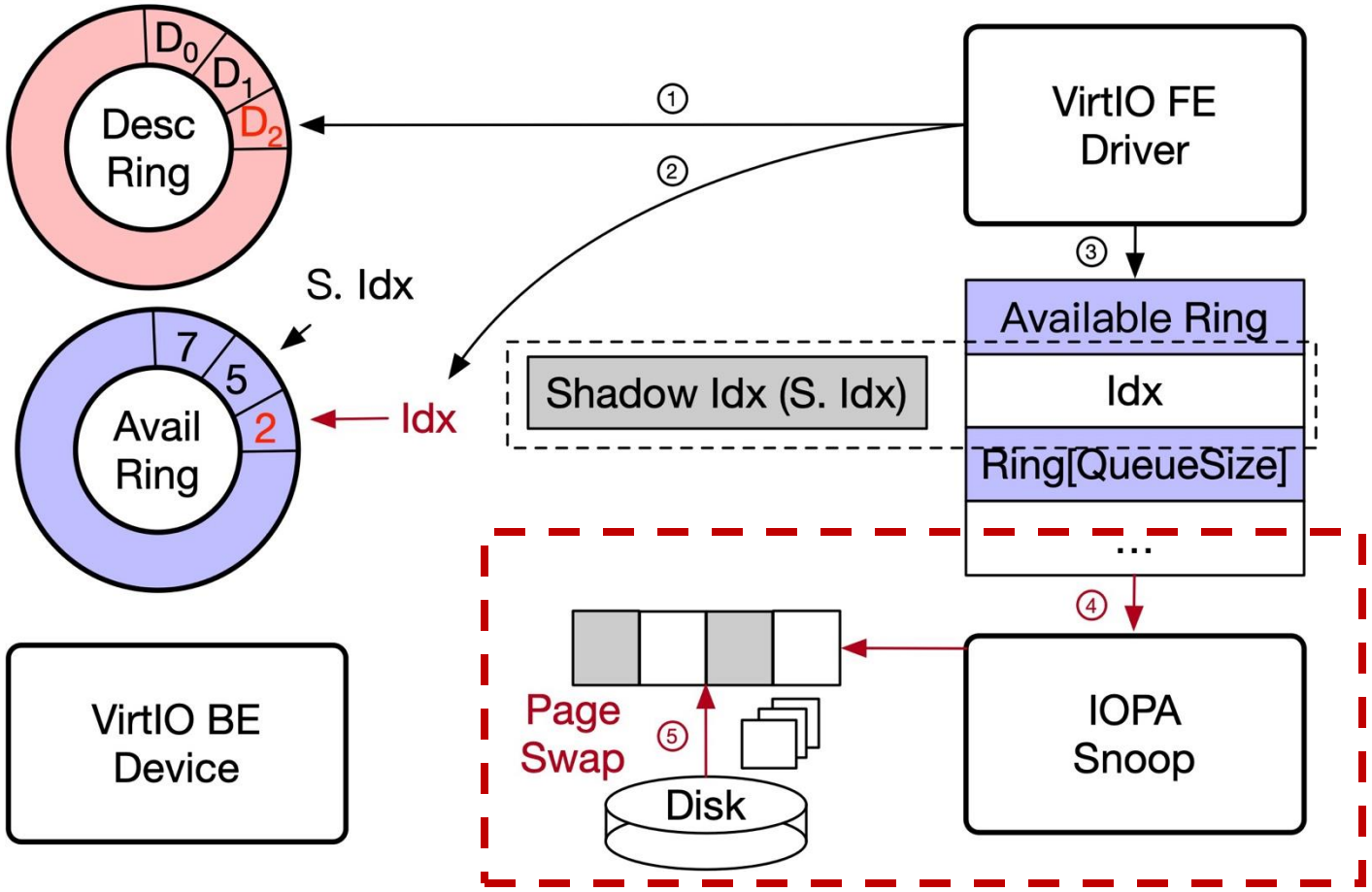
VirtIO Model (Simplified)

IOPA-Snoop (cont.)



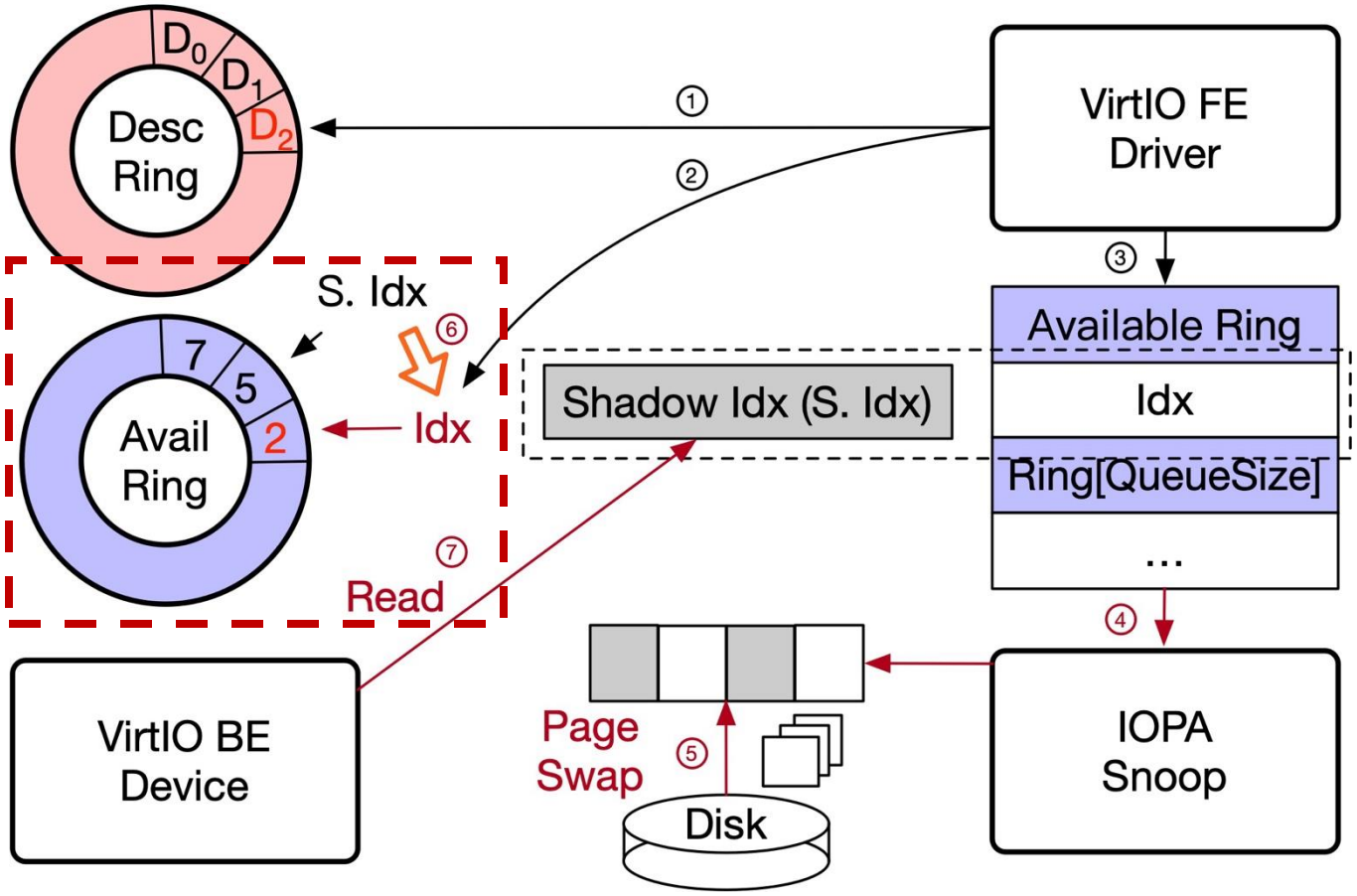
In **VIO**, device reads **Shadow Index** instead of **regular available index**.

IOPA-Snoop (cont.)



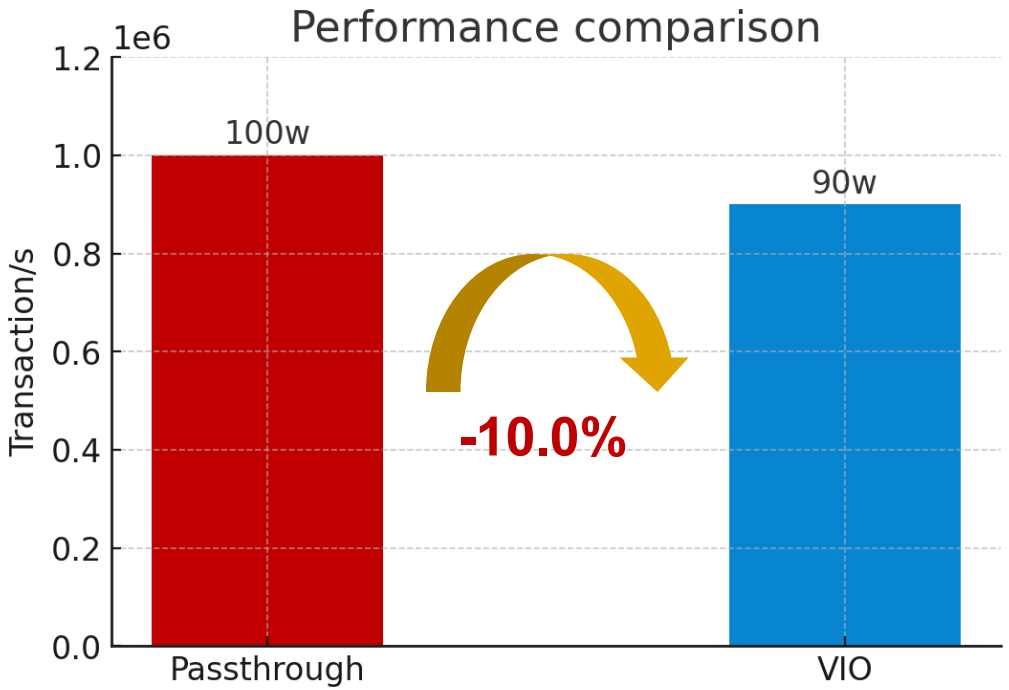
By monitoring DMA addresses in the vring, the hypervisor handles page faults on the CPU side.

IOPA-Snoop (cont.)

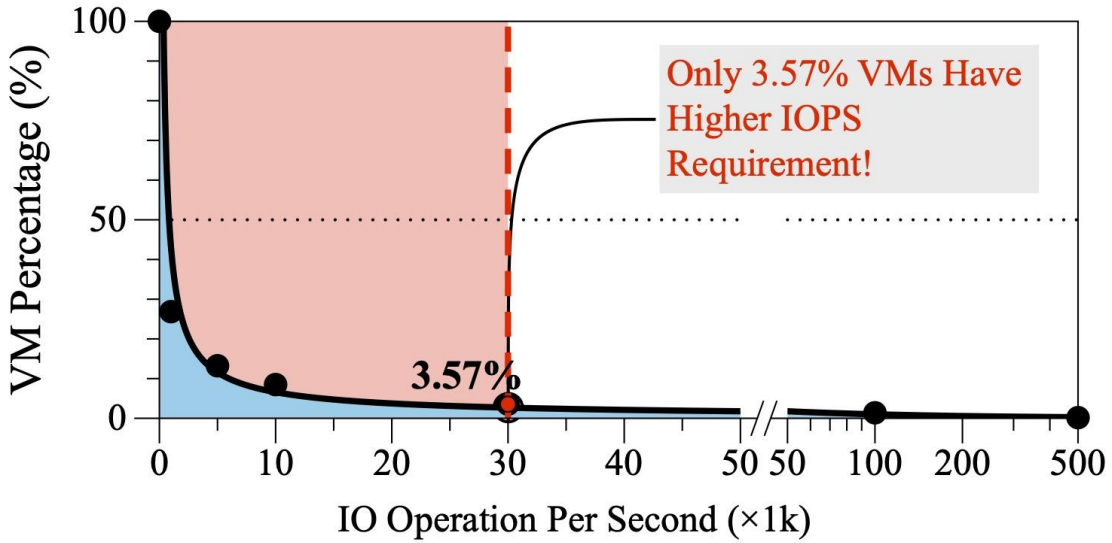


Once the hypervisor updates the **Shadow Index**, the device sees the update and starts the DMA transfer using the descriptor.

I/O-Intensive Workloads



At high IOPS, cumulative snooping overhead causes a **~10% performance drop**.



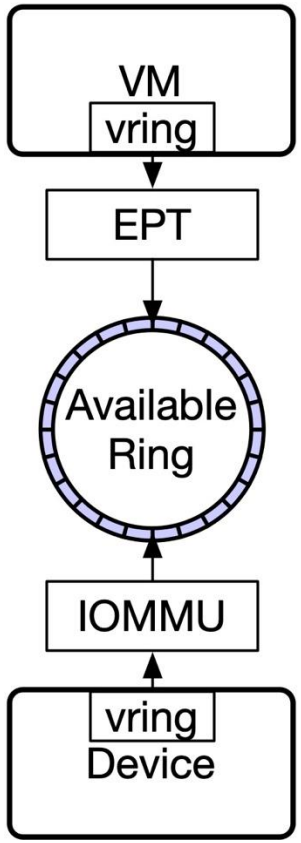
High IOPS is rare in production, affecting **less than 4% of VMs**.

Elastic Passthrough

- IOPS-Aware Elastic Passthrough, dynamic change mode according to **I/O pressure**.
  - Low IOPS (The Common Case): The VM runs in **VIO snooping mode**.
  - High IOPS (The Rare Case): The VM is automatically switched to **native passthrough mode**.

Elastic Passthrough (cont.)

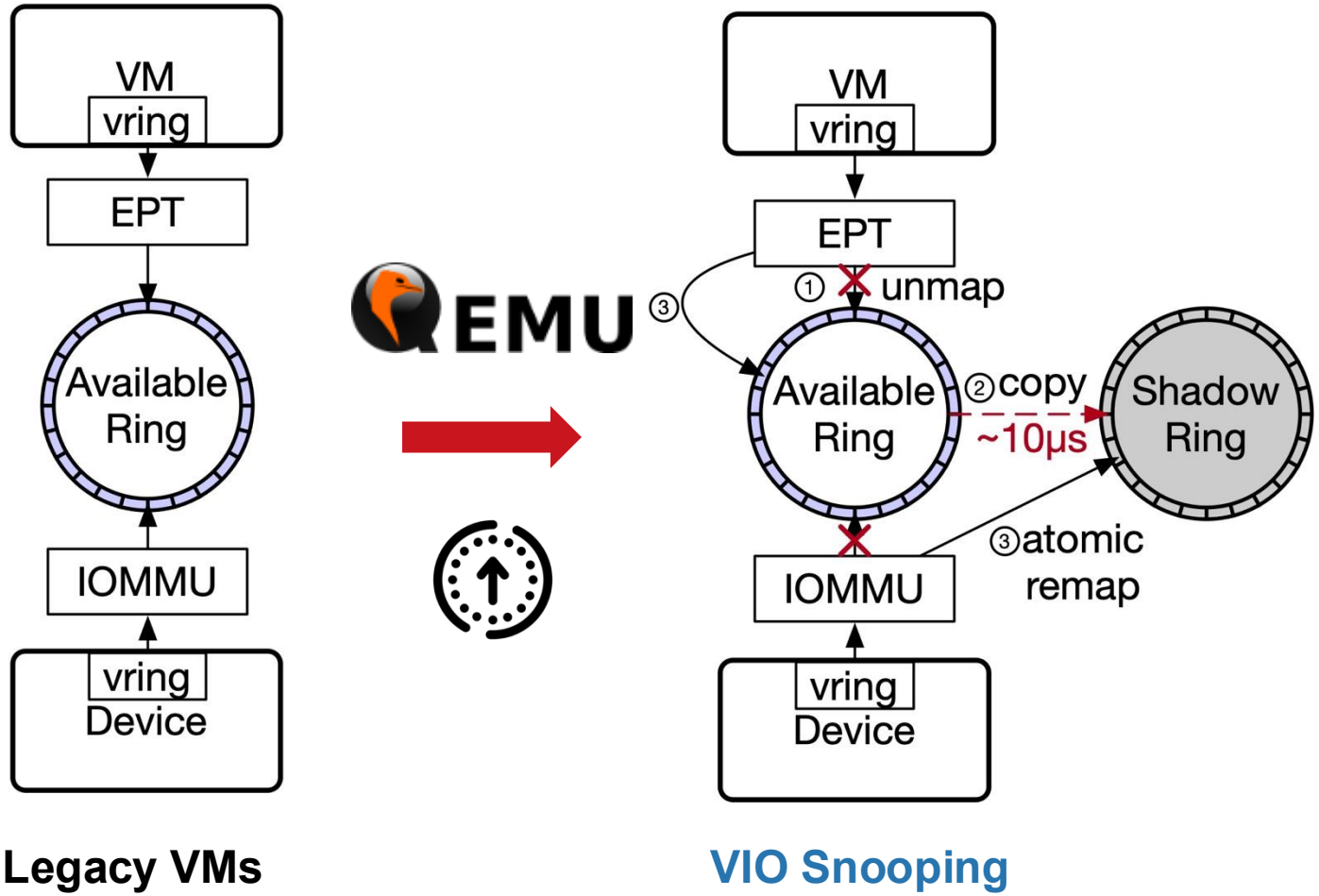
- **Passthrough**



Legacy VMs

Elastic Passthrough (cont.)

- **Passthrough → Snooping (Live Upgrade)***



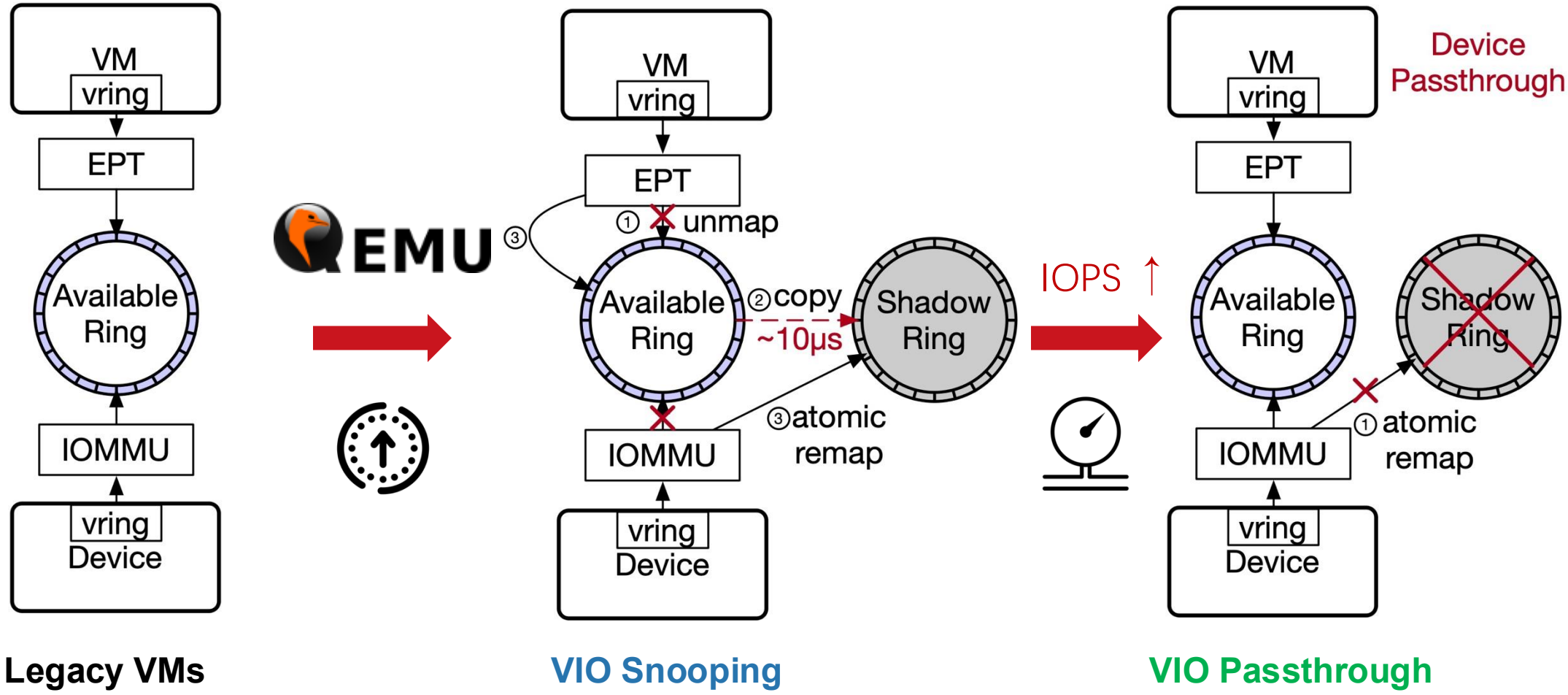
Legacy VMs

VIO Snooping

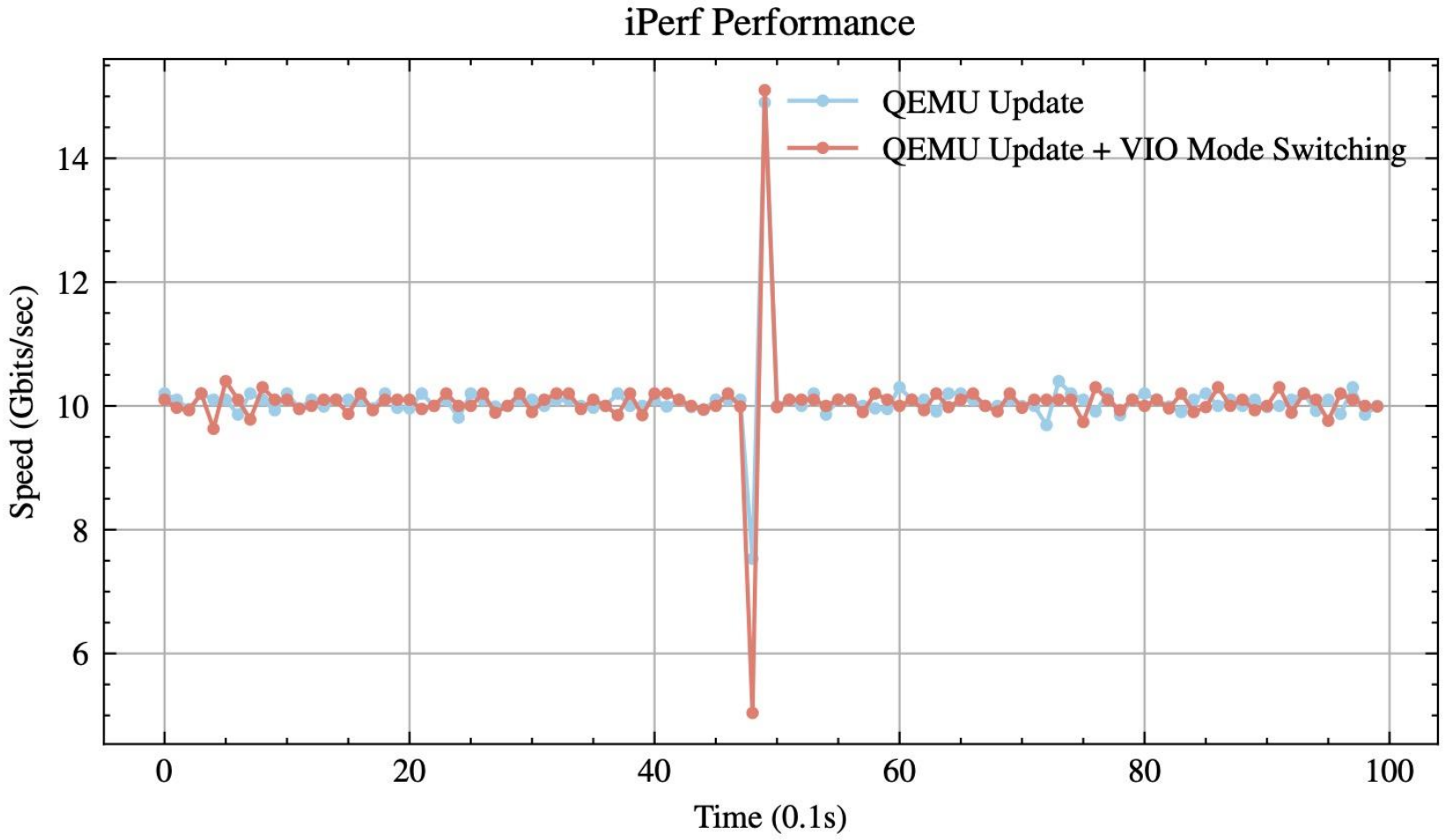
[1]Zhang, Xiantao, et al. "Fast and scalable VMM live upgrade in large cloud infrastructure." *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*. 2019.

Elastic Passthrough (cont.)

- Snooping → Passthrough (IOPS increases)



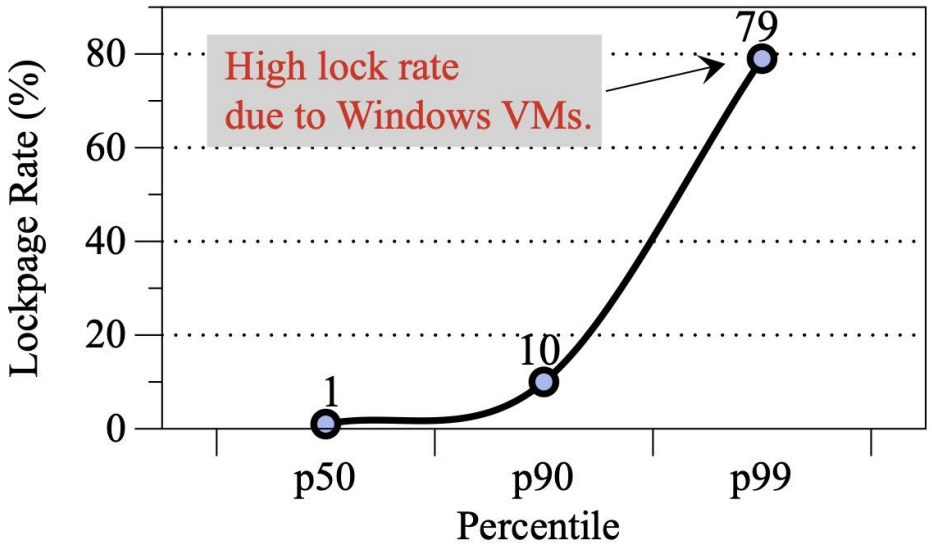
Elastic Passthrough (cont.)



VIO rollout in the Alibaba production

IOPFs Reduction - Lockpage

- Lockpage mechanism
  - Uses IOPA-snoop to log page access patterns and identify these frequently used I/O pages.
  - Lockpage mechanism "locks" or "pins" hot pages in memory.

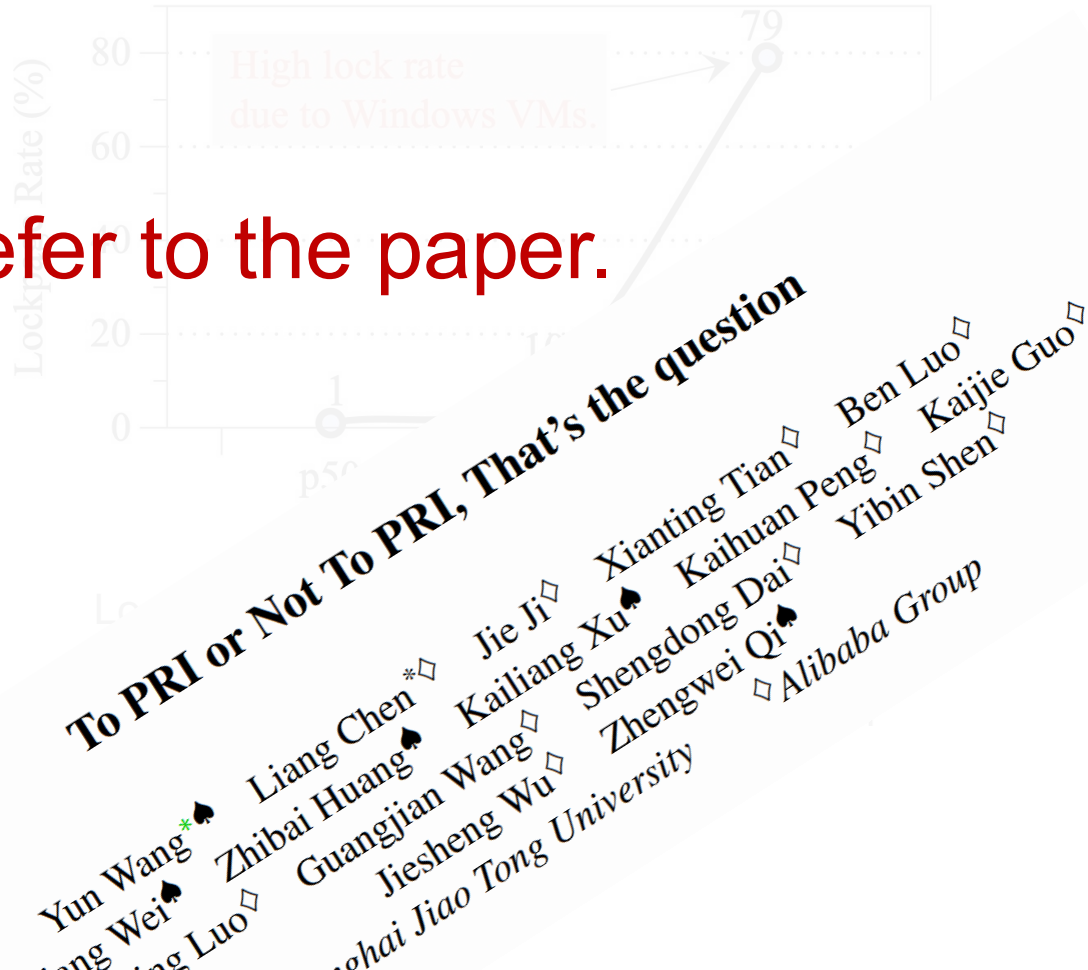


Lockpage rate in a production environment, showing **P50 at 1%, P90 at 10%, and P99 spiking to 79%** due to poor I/O locality in Windows VirtIO driver.

IOPFs Reduction - Lockpage

- Lockpage mechanism
  - Uses IOPA-snoop to log page access patterns and identify the frequently used I/O pages.
  - Lockpage mechanism "locks" or "pins" hot pages in memory.

For more details, please refer to the paper.



To PRI or Not To PRI, That's the question

Yun Wang*♠ Liang Chen*□ Jie Ji□ Xianting Tian□ Ben Luo□
Zhibai Huang♠ Kailiang Xu♠ Kaihuan Peng□ Kaijie Guo□
Guangjian Wang□ Shengdong Dai□ Yibin Shen□
Jiesheng Wu□ Zhengwei Qi♠
Shanghai Jiao Tong University □ *Alibaba Group*

Experimental Setup

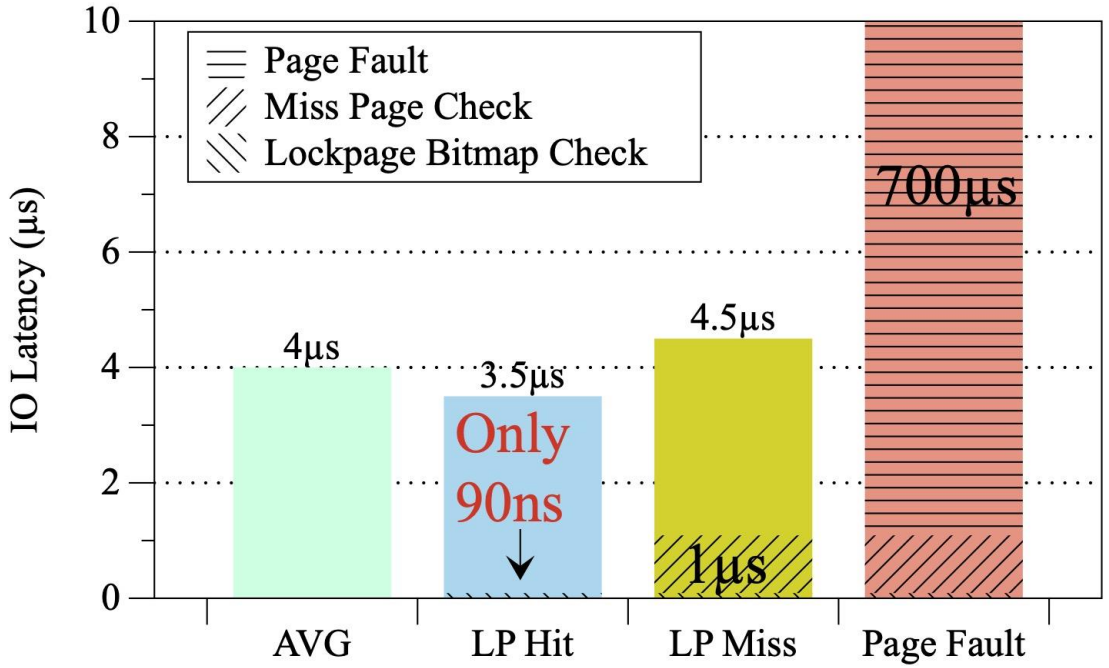
Software

- Applications: iperf/nginx/Redis/Memcached
- Dataset: YCSB
- Baseline: VPRI[SOSP'24]

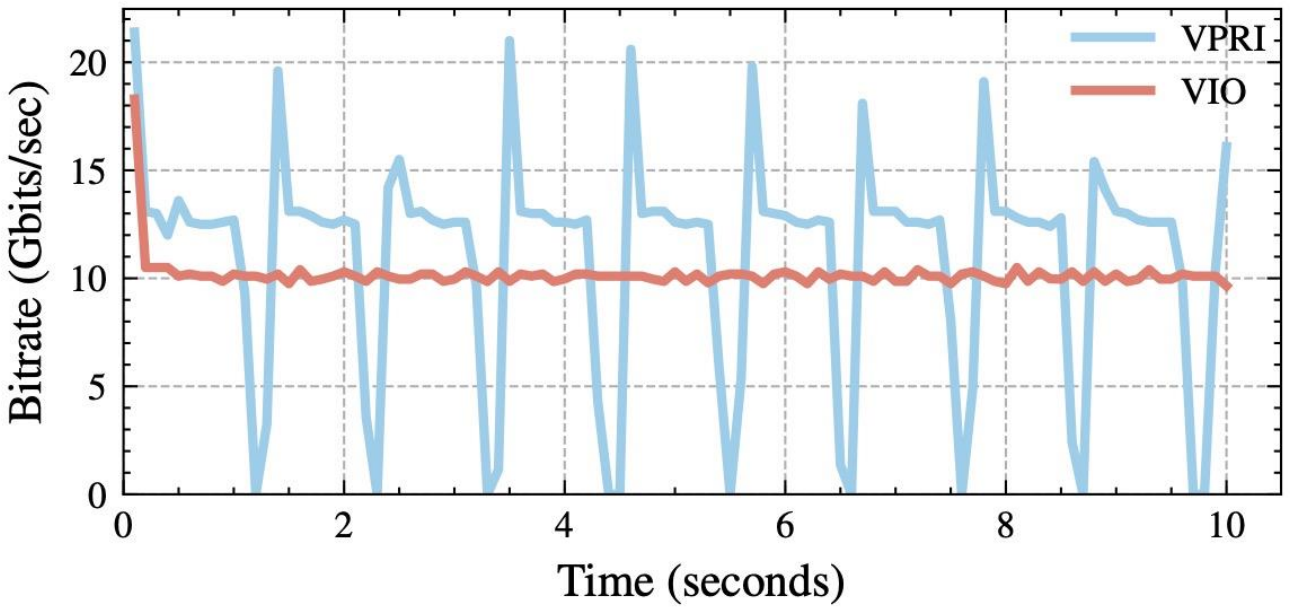
Hardware

Host	2x Intel Xeon Platinum 8269CY (52C/104T), 192GB RAM, 1TB SSD
DPU	200 Gb/s Bandwidth, PCIe Gen3, Supports 2300 VFs
VM	4 vCPUs, 8GB RAM, CentOS 7.9, 10 Gb/s Network Throughput

Evaluation

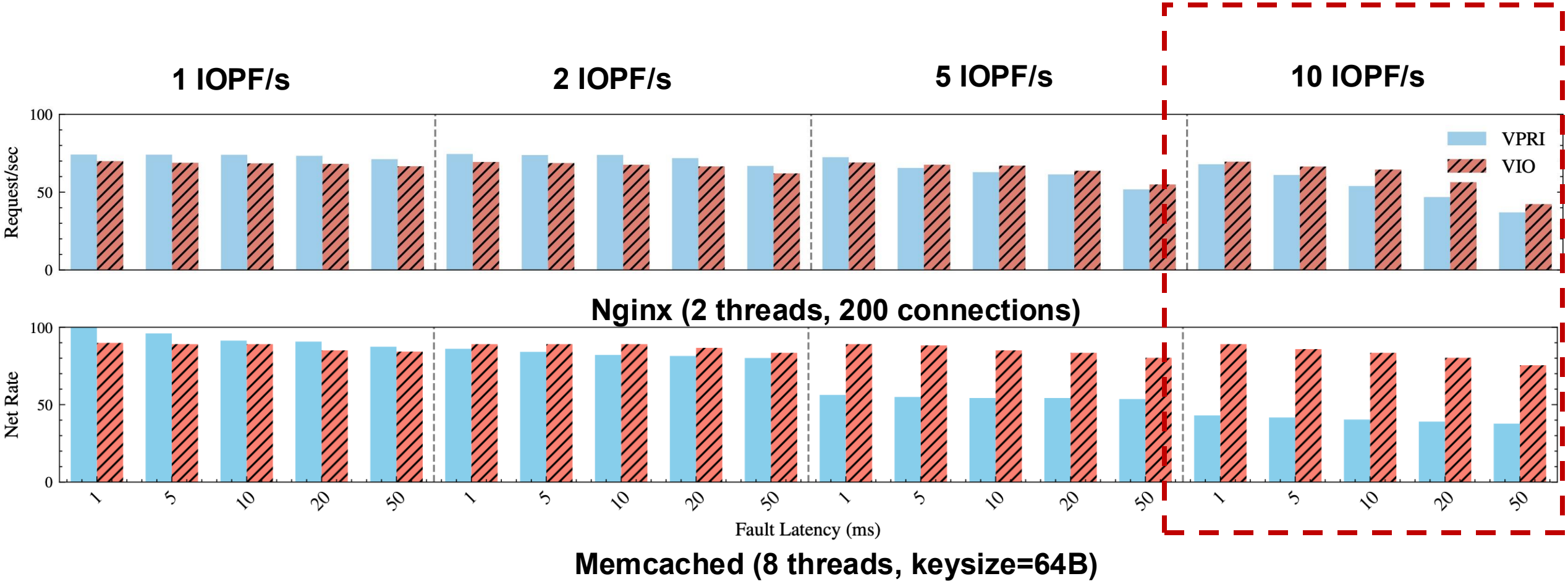


IOPA-Snoop Latency



iperf3 performance
(1 IOPF/s with 5ms latency)

Evaluation



As the rate of I/O page faults per second (IOPF/s) increases, VIO's performance remains consistent, with throughput degradation staying below 10% across all applications.

Conclusion

- VIO is a practical software alternative to hardware-dependent approaches like PRI, offering hardware independence and high performance.
- It uses IOPA-Snooping and Elastic Passthrough to move fault handling off the critical path, achieving both near-native speed and resource efficiency.
- VIO is successfully deployed on 300K VMs in the production, reclaiming significant daily memory while maintaining user SLOs.

Thank you!

Email: yunwang94@sjtu.edu.cn