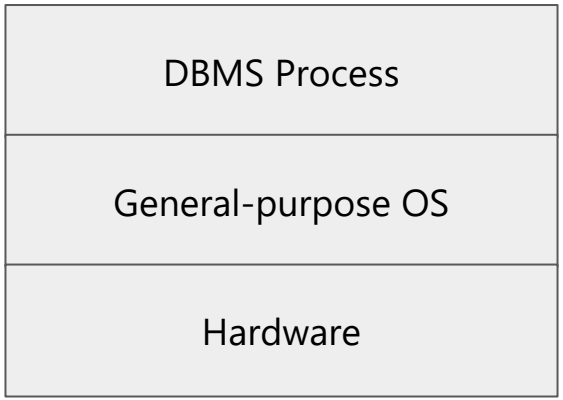
Practical DB-OS Co-design with Privileged Kernel-Bypass

Xinjing Zhou

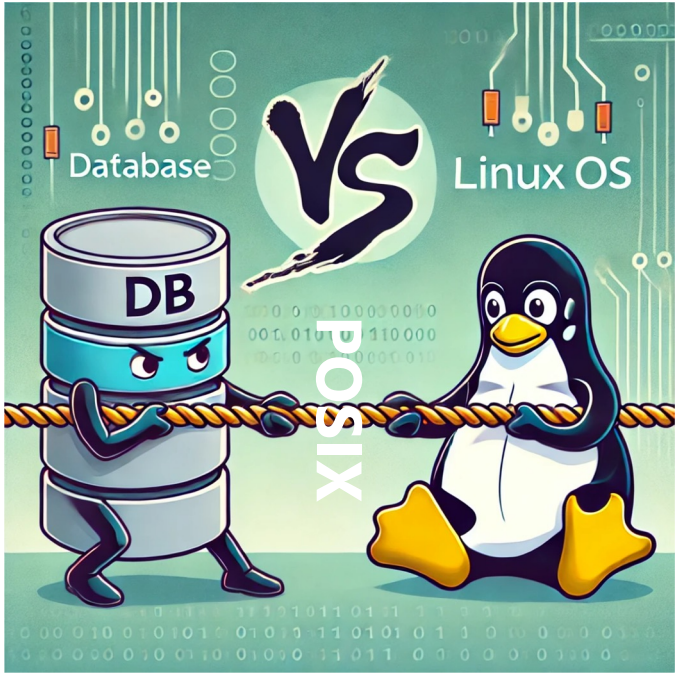
with Viktor Leis, Xiangyao Yu, Michael Stonebraker

DBMS on top of OS



DB-OS Interface Mismatch

Performance
Hardware Control



Security
Resource Efficiency

Are You Sure You Want to Use MMAP in Your Database Management System?

Andrew Crotty
Carnegie Mellon University

Viktor Leis
Friedrich-Alexander-Universität

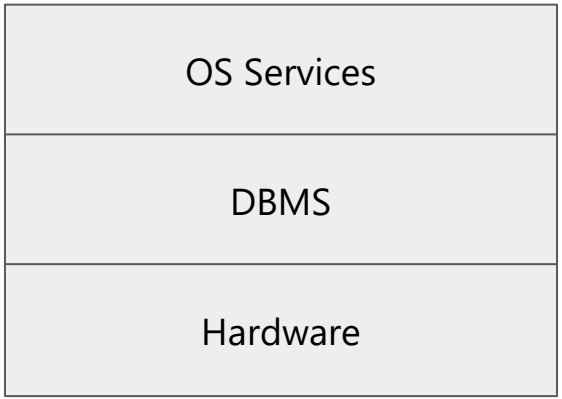
Andy Pavlo
Carnegie Mellon University

AIO is a horrible ad-hoc design, with the main excuse being "other, less gifted people, made that design, and we are implementing it for compatibility because ***database people - who seldom have any shred of taste - actually use it****".*

- Linus Torvalds in 2016

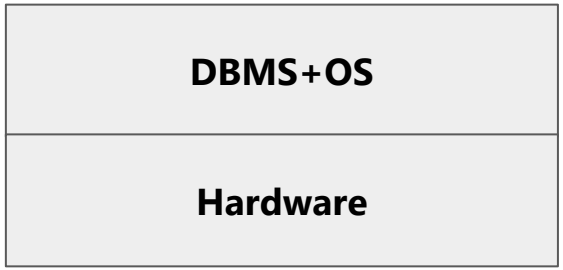
OS on top of DBMS

- The DBOS-project
- Requires a revolution

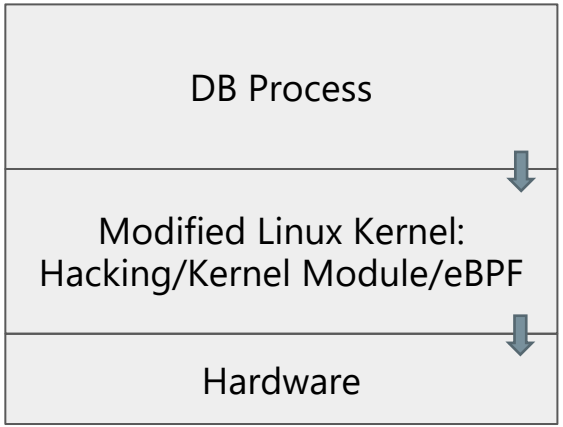


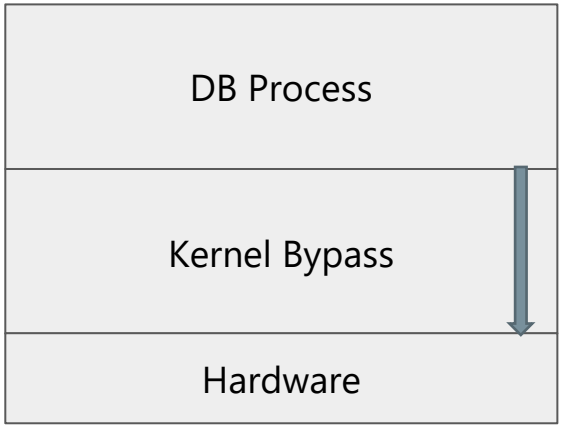
A Middle-ground: Co-design

- Blurring the boundary of DB and OS
- Some pieces in the OS
- Focus of this talk



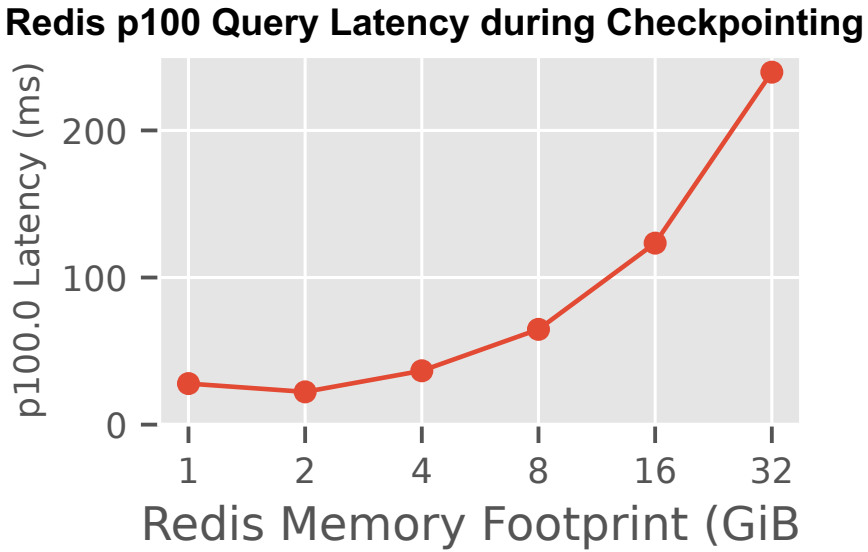
Co-design Paradigms





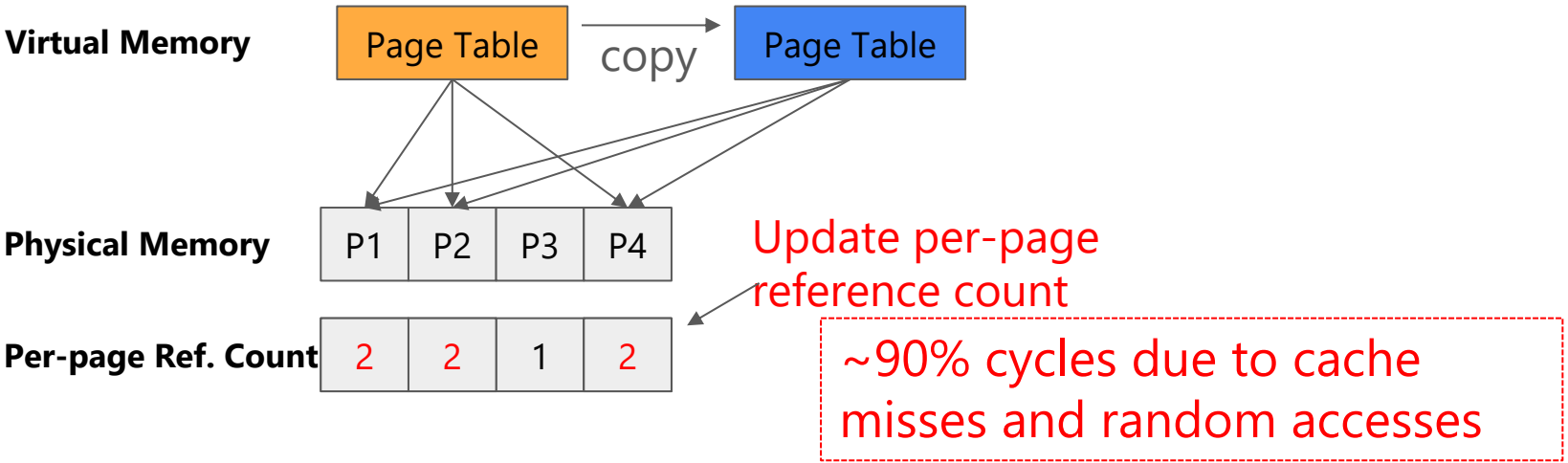
Case Study: Virtual Memory Snapshotting

- Redis uses fork to save process memory as checkpoints for persistence
- fork is **blocking** and requires threads to be paused to get a consistent snapshot

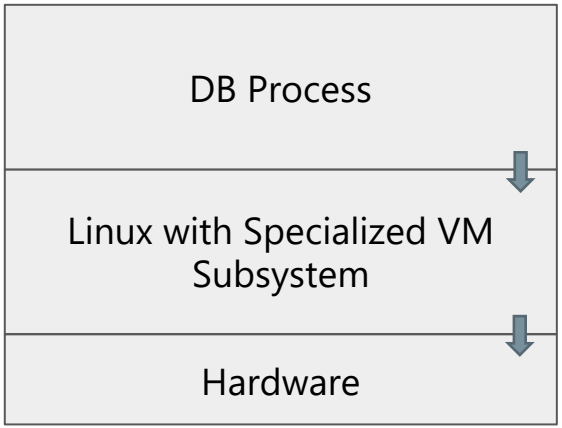


fork Bottleneck Analysis

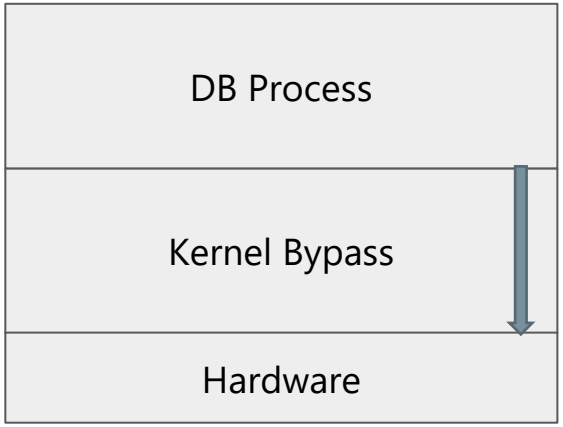
- Linux kernel maintains a per-page reference count for safe page reclamation – a fundamental design decision to support shared-memory, page cache …



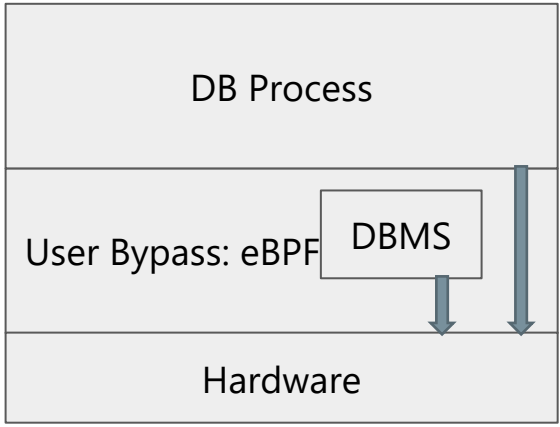
Co-design Paradigms for this Problem



- Security/stability issue
  - CrowdStrike incident
- Fundamental design limitation



- Only works for networking and storage

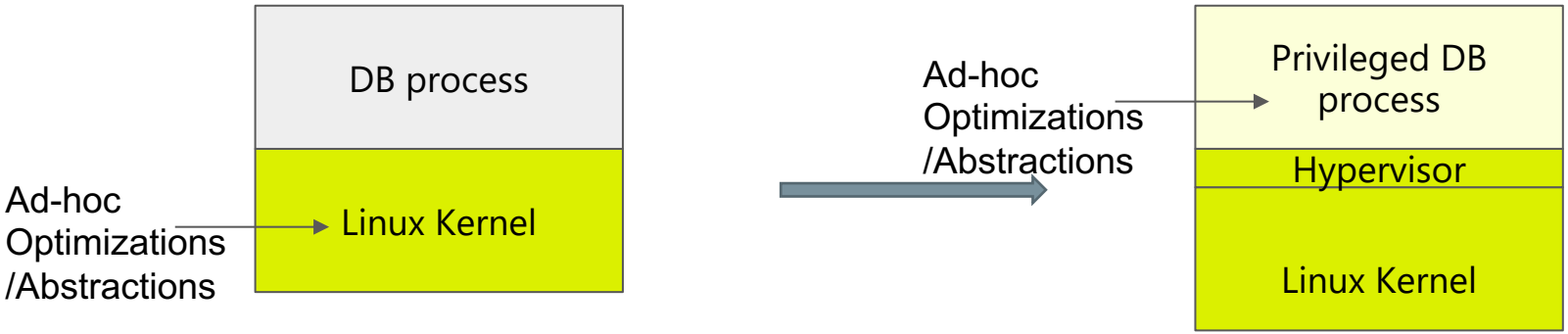


- Limited programmability

Privileged Kernel-Bypass: complete freedom to specialize subsystems while minimizing impact on security, stability, and compatibility

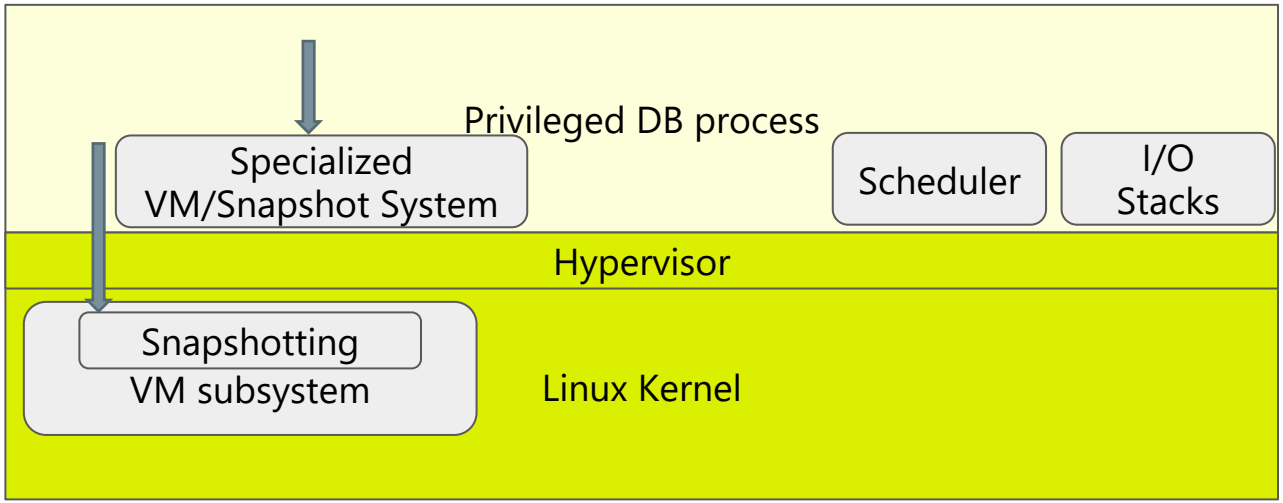
Specialize in an Unconstrained and Safe Place – Virtualized Environment

- There is a class of hypervisors[1] (Dune) that raises the privilege level of a Linux process
  - Runs in **Guest Kernel Space with access to all privileged instructions:** paging, interrupts, rings…
  - Preserves process abstraction to reuse host kernel features



[1] Belay, Adam, et al. "Dune: Safe user-level access to privileged {CPU} features." *OSDI 12*. 2012.

Privileged Kernel-Bypass: Selectively Specialize Data-Intensive Subsystem



Privileged Kernel-Bypass vs. Kernel-Bypass for DBMS

	Kernel-Bypass	**Privileged Kernel-Bypass**
DBMS runs in	User space	Guest Kernel Space
Specializes	Network/Storage	Virtual Memory/Scheduler/Interrupt /Network/Storage

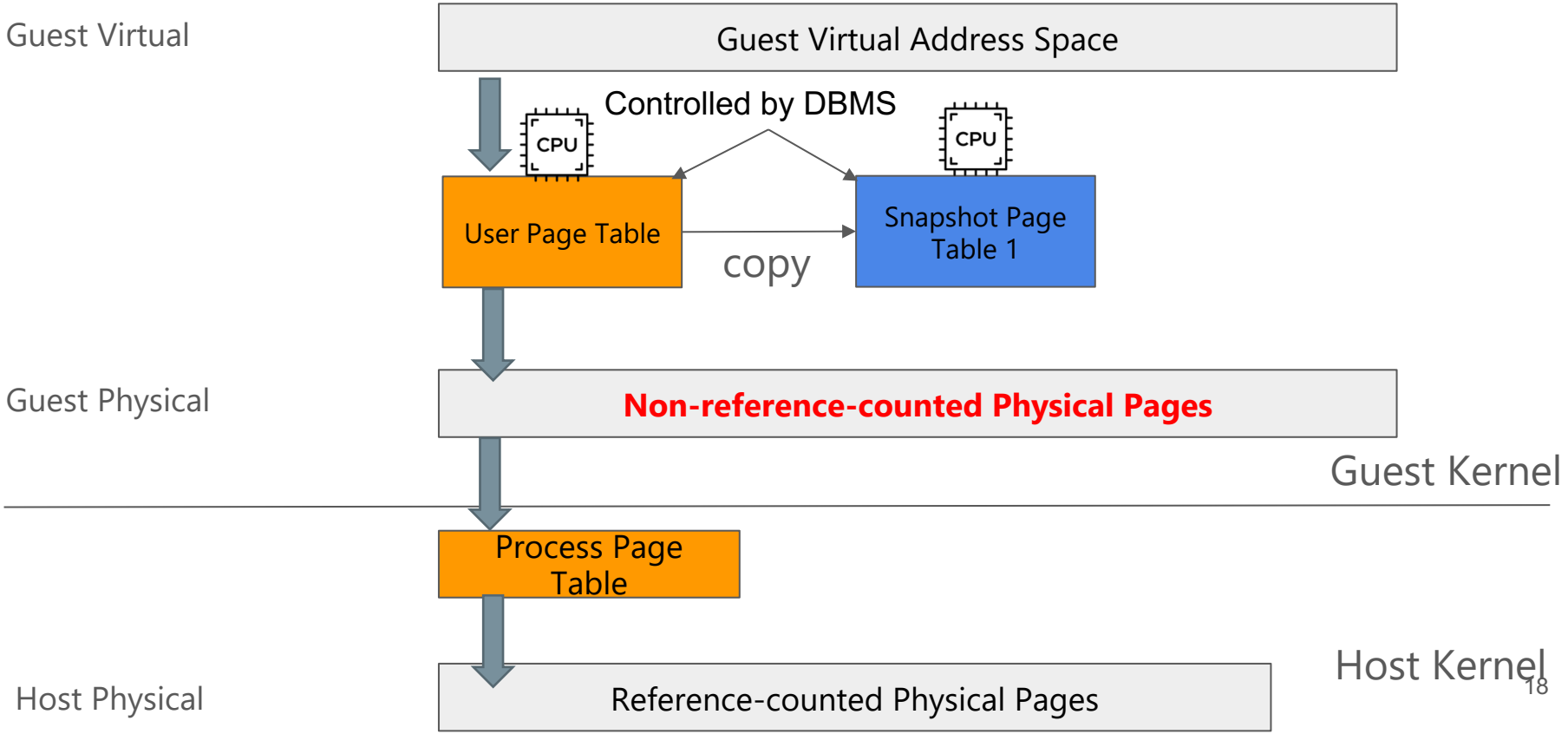
Numerous Possibilities

- **Fast snapshotting. ←**
- A “perfect mmap” buffer manager ?
- Faster memory-rewiring for DBMS applications
- UDF sandboxing: UDF in guest userspace and DBMS in guest kernel space.
- Lightweight Preemptive Scheduler
- Faster memory allocation
- ……

Attacking the fork Problem in Privileged DB process

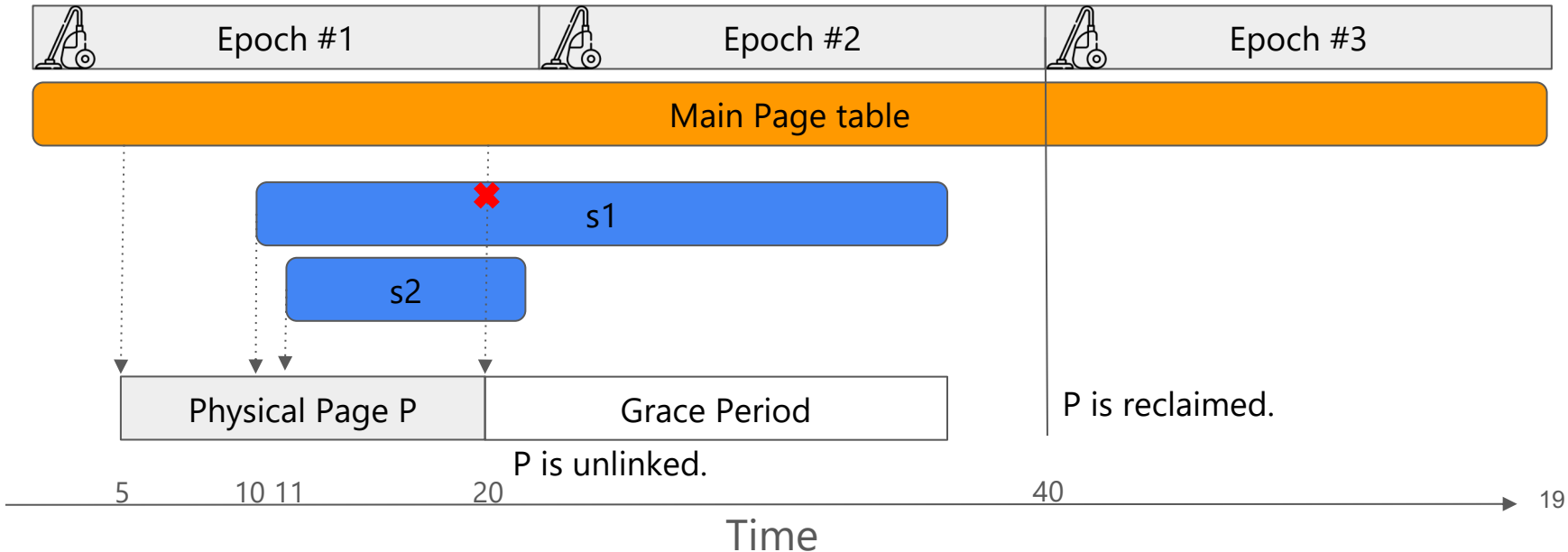
- Specialize an extremely simple VM/snapshotting system in the privileged DB process
- No reference counting for physical pages

Specialized VM Subsystem for Fast Snapshotting



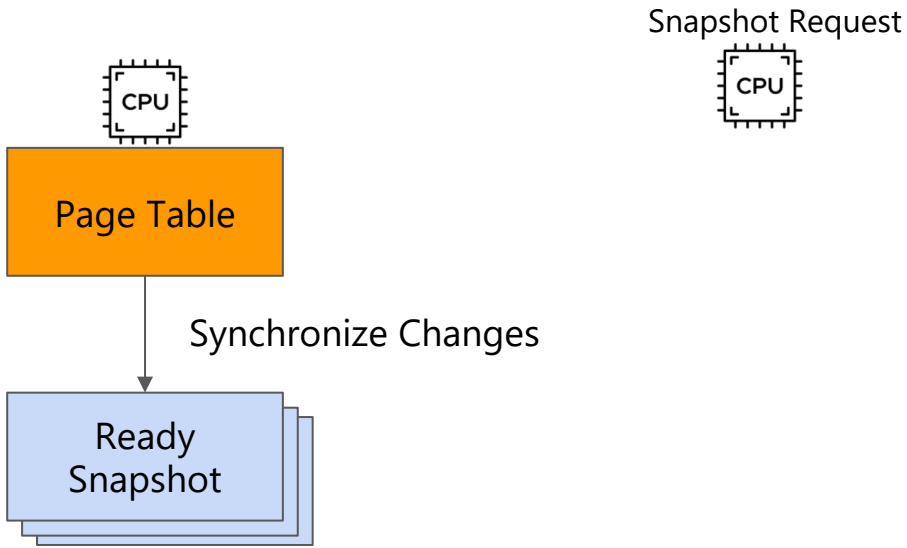
Safely Reclaiming Physical Pages with Epochs

- Assumption: snapshot is only allowed to be created from main page table
- A page is reclaimed at epoch boundary when there is no references from page tables.



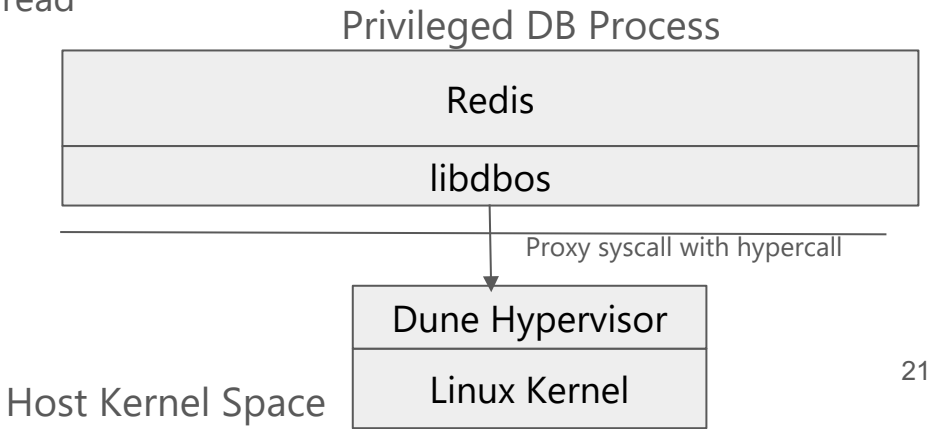
Instant Snapshot Creation via Pre-creation

- Asynchronously pre-create and maintain a set of ready-to-go snapshot page tables
- Completely hide the copy latency, making the snapshot creation appear instant



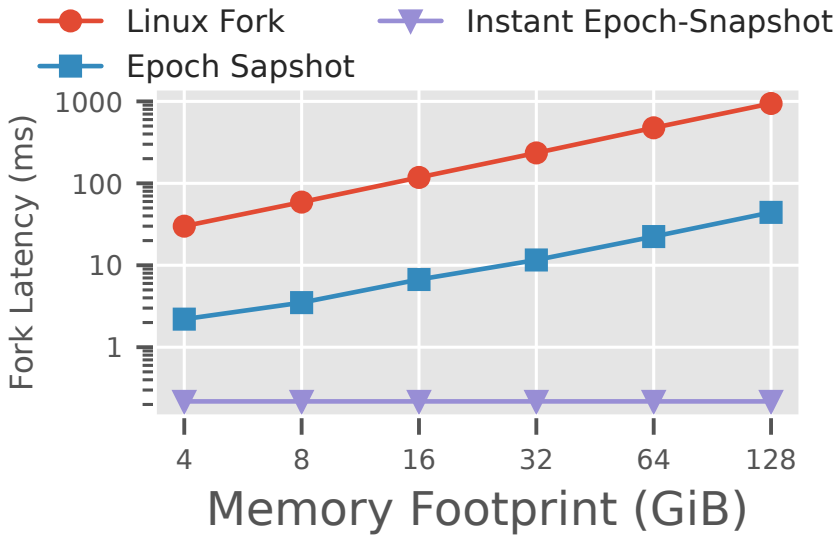
Implementation

- The snapshot mechanism is implemented (~1K LOC) in a guest kernel called **libdbos** on top of Dune hypervisor
- Physical memory backing and system call proxy are done by the hypervisor
- Evaluated on Redis by replacing fork with this snapshot mechanism
  - Checkpoint process runs in a separate thread

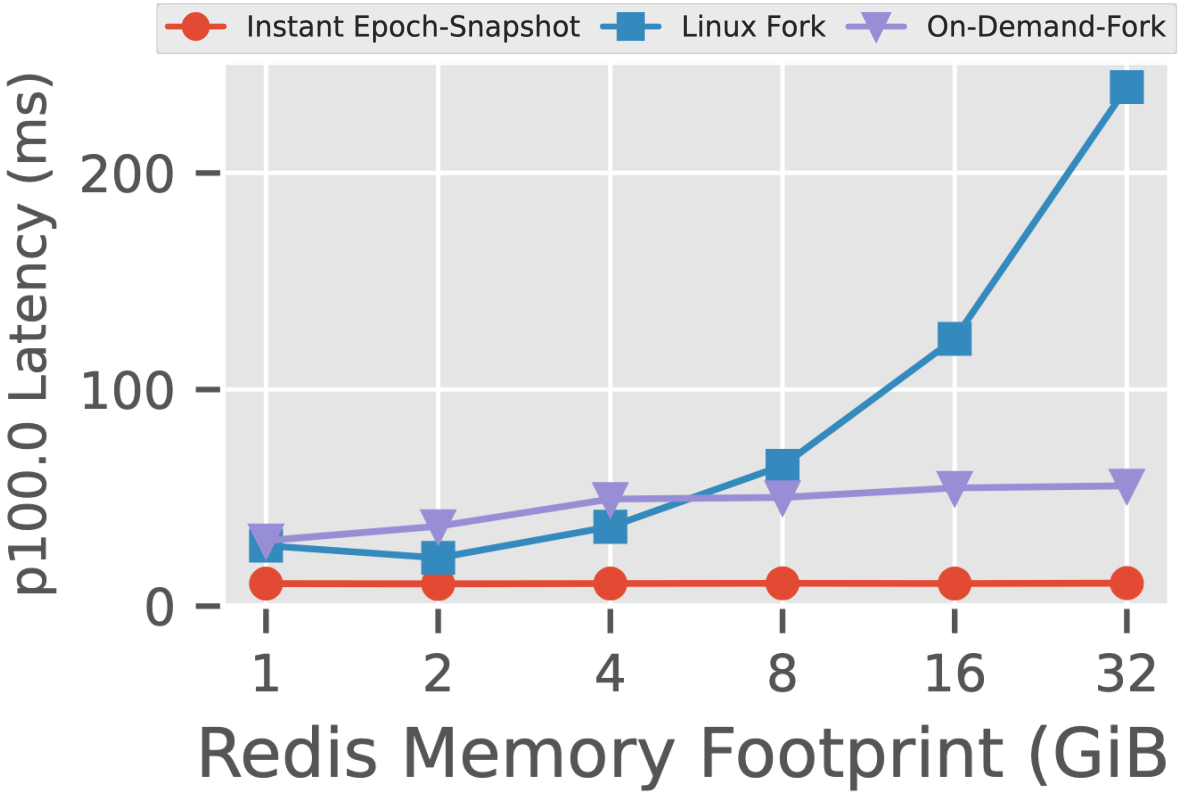


Microbenchmark

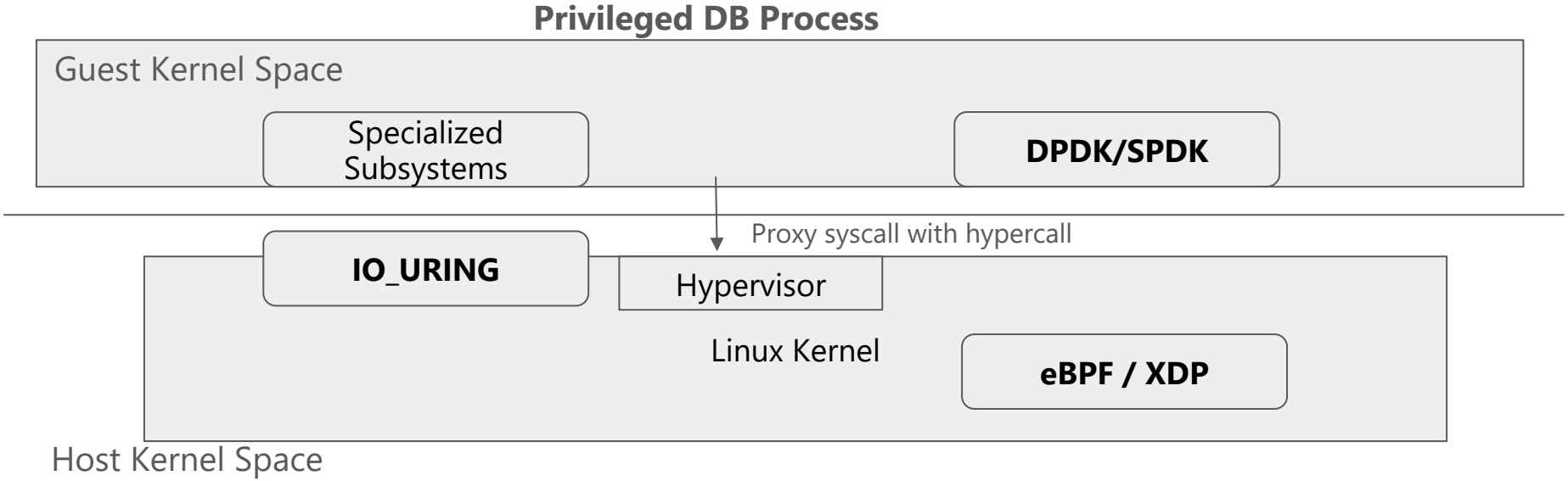
- ~20x reduction in snapshot latency
  - Snapshot 128GB memory in 40ms without parallelization
- Async copy completely hides fork latency if snapshot frequency > page table copy time



Tail Latency of Redis set **Query during Checkpoint**



Orthogonal to Linux Bypass Mechanisms



Conclusions

- With **privileged kernel-bypass**, we can address the mismatch problem while
  - minimizing impact on kernel security and stability
  - providing complete freedom to developers
  - preserving ecosystem
- DBMS deserves to be to **privileged!**
- Contact us at xinjing@mit.edu