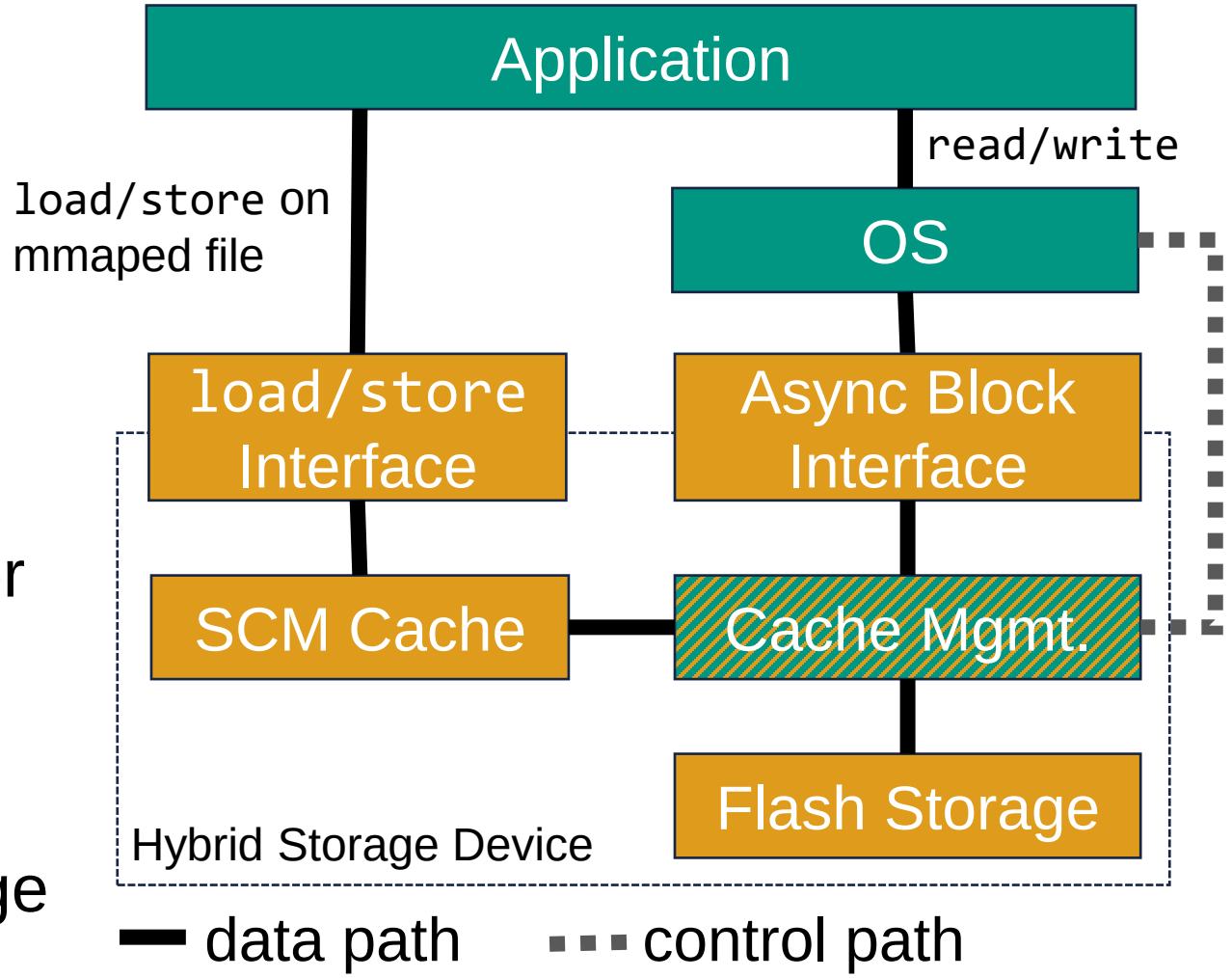
Karlsruhe Institute of Technology

Towards Hybrid Storage Devices with Block and DAX Interface

Daniel Habicht, Yussuf Khalil, Lukas Werling, Frank Bellosa

(Re)defining Hybrid Storage

- Dual-Paradigm:
  - Asynchronous Block I/O
  - Synchronous `load/store`
- Flash for backing storage
- Storage-Class Memory (SCM) for `load/store` access on storage
- Our contribution:
  OS abstractions for hybrid storage



Hybrid Storage: Use Cases & Prior Research

- Advantages of SCM at price close to Flash

- Memory Tiering
  - Cheap DRAM replacement
  - Hybrid storage device for slow tier memory
  - Not discussed here further

- Hybrid storage for I/O
  - File systems (journaling, cross-media fs)
  - Apps with strong persistence requirements
    e.g., write-ahead logging (WAL) in DBMS
  - Transparent use of SCM in unmodified applications

"2B-SSD"
D.-H. Bae et al. (ISCA '18)

"FlatFlash"
A. Abulila et al. (ASPLOS '19)

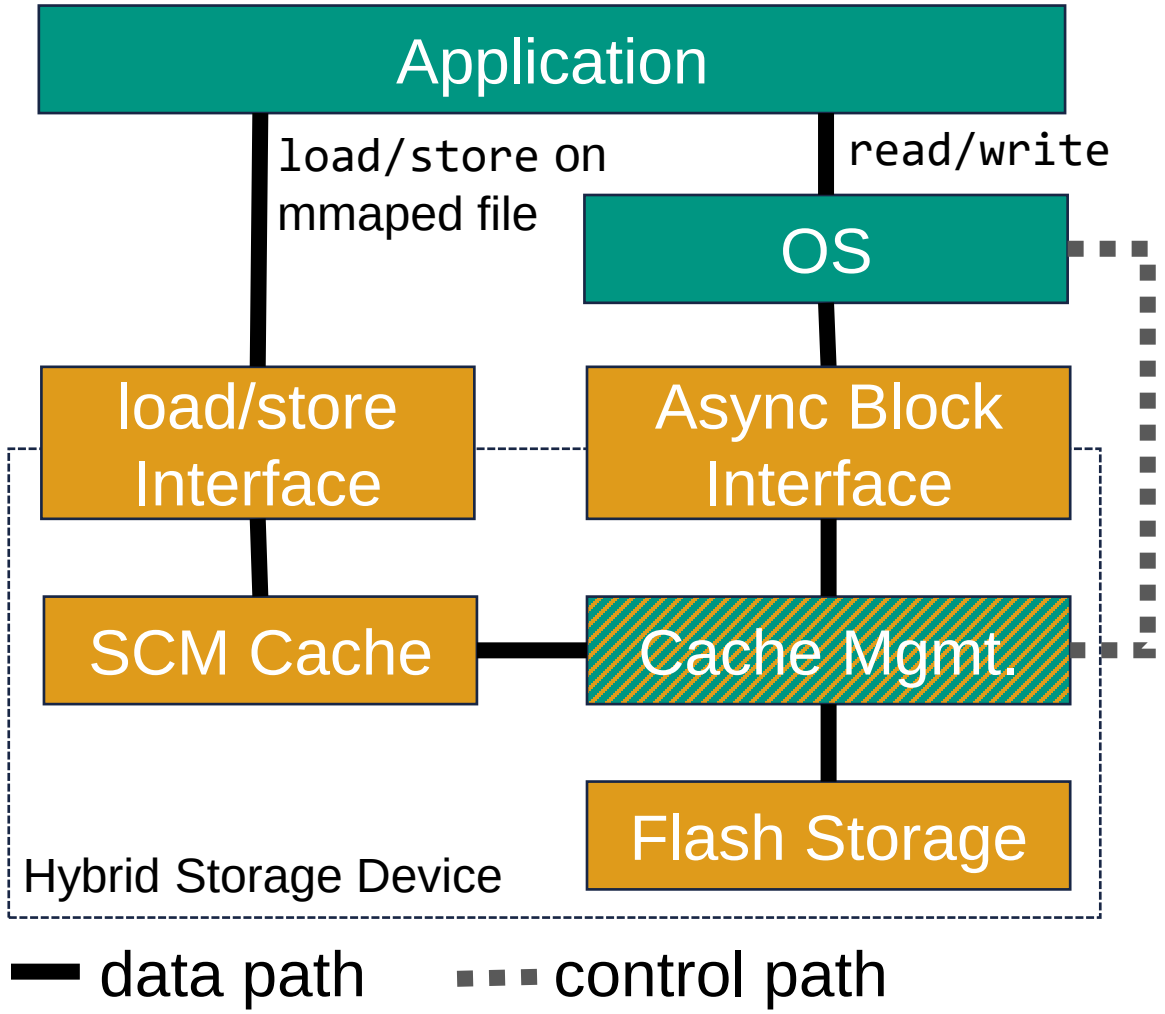
"Hello bytes, bye blocks"
M. Jung (HotStorage '22)

Hybrid Storage: Why Now?

- SCMs available for years but no commercial hybrid storage
- PCIe unfit for hybrid storage
  - read/write transaction not optimized for low-latency operation
  - Host cannot cache device-attached memory
- Compute Express Link (CXL)
  - Growing availability of CXL-capable hardware
  - CXL.mem for low-latency load/store semantics on device-attached memory
  - Global Persistent Flush (GPF)
- First commercial offerings on horizon (Samsung’s CMM-H)

Hybrid Storage: Challenges

- Abstraction for hybrid storage
- Coherency of interfaces
- Limited SCM capacity
  - Fairness
  - Performance guarantees
- Transparent use of SCM



Linux Direct-Access (DAX)

- DAX bypasses the page cache (zero-copy access)
- Currently supported by ext2, ext4 and XFS

- Per-inode DAX flag → no fine-granular control
- Assumes non-blocking access at all times
  →cannot use swapping mechanism for SCM cache

Linux Direct-Access (DAX)

- DAX bypasses the page cache (zero-copy access)
- Currently supported by ext2, ext4 and XFS

- Per-inode DAX flag → no fine-granular control
- Assumes non-blocking access at all times
  →cannot use swapping mechanism for SCM cache

→ Existing DAX support unsuitable for hybrid storage

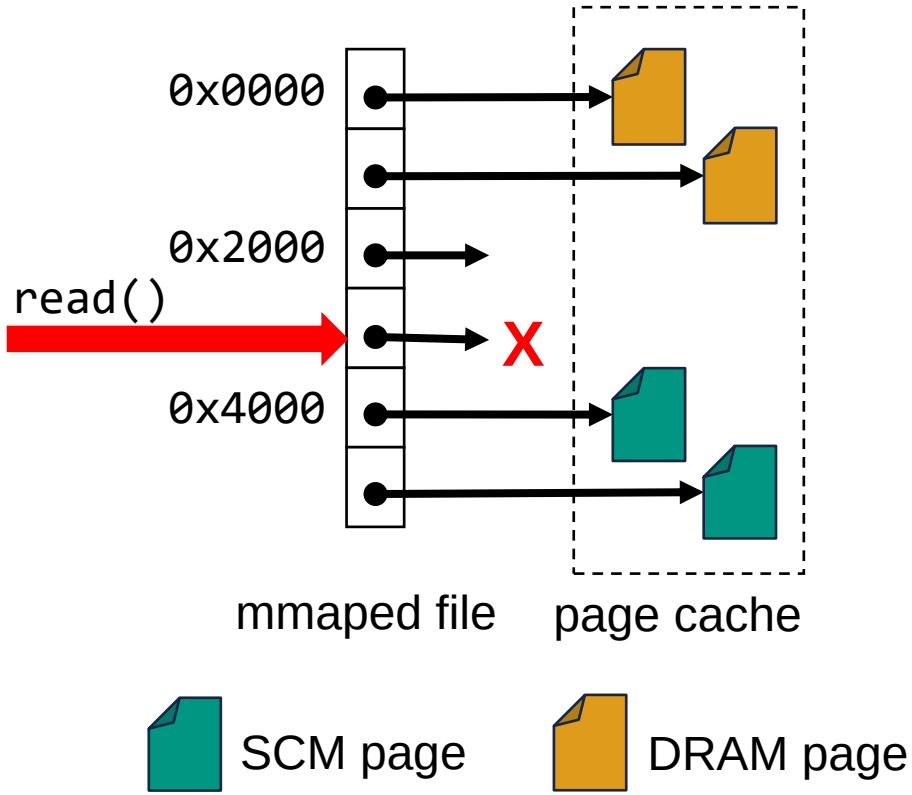
Supporting Hybrid Storage in Linux

- No hardware development platform for hybrid storage
  - →Emulate hybrid storage with PMEM + NVMe SSD
  - →Implement cache management inside OS
- First approach: build indirection on top of existing DAX support
  - Requires reimplementation of many core mm components
  - Too much complexity, prone to errors
- Persistency-aware Page Cache
  - Reuse existing mm functionality
  - Few changes in FS required

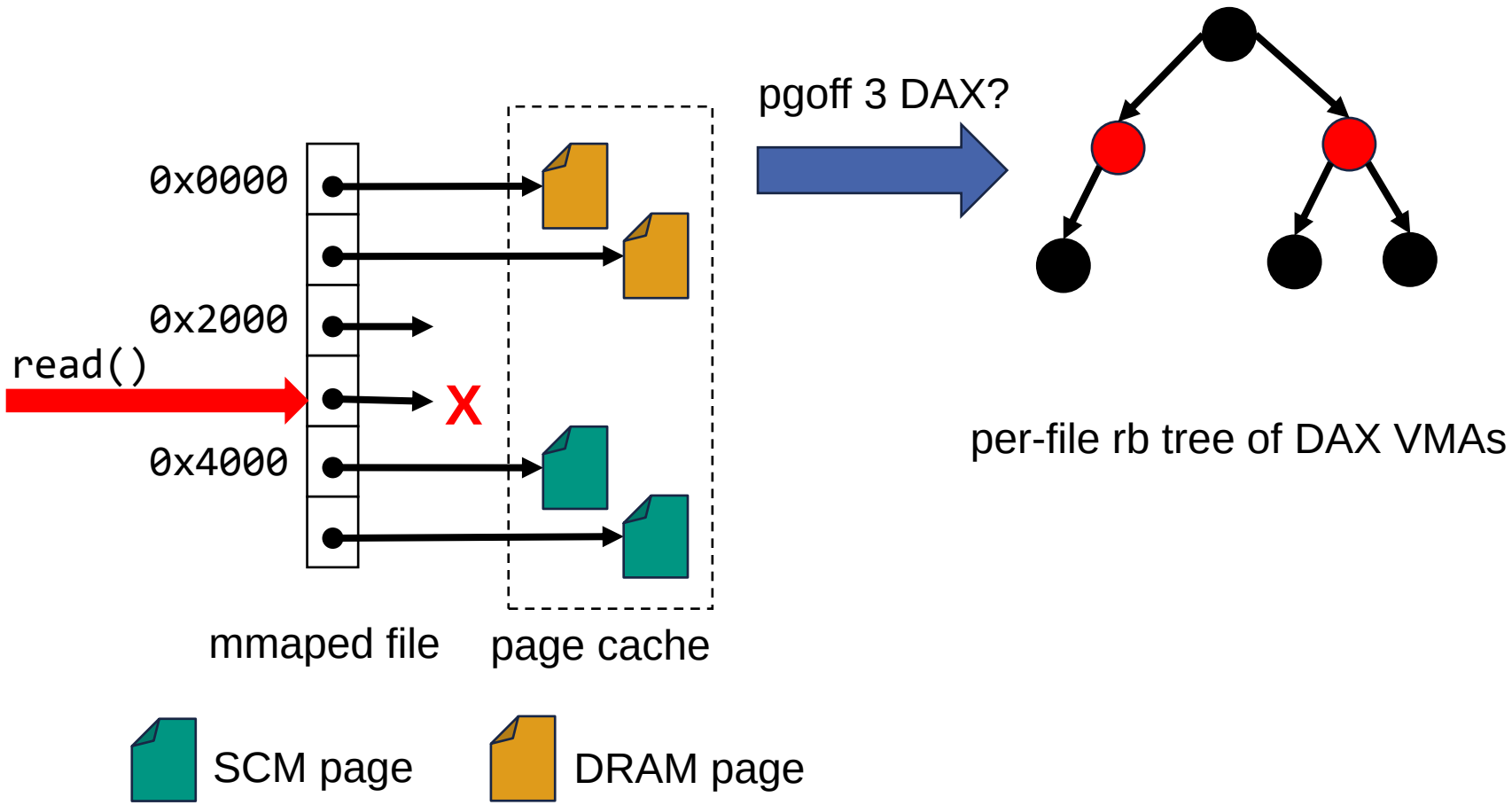
User Space API for Hybrid Storage

- `mmap` with `MAP_DAX` flag for requesting DAX mapping
  - Must be used with `MAP_SHARED_VALIDATE`
- `mlock` for pinning page to SCM cache
  - Guarantees absence of major faults
  - New `rlimit` for controlling amount of pinned DAX pages
- Global limit for total amount of pinned DAX pages
- Direct I/O directly on SCM cache

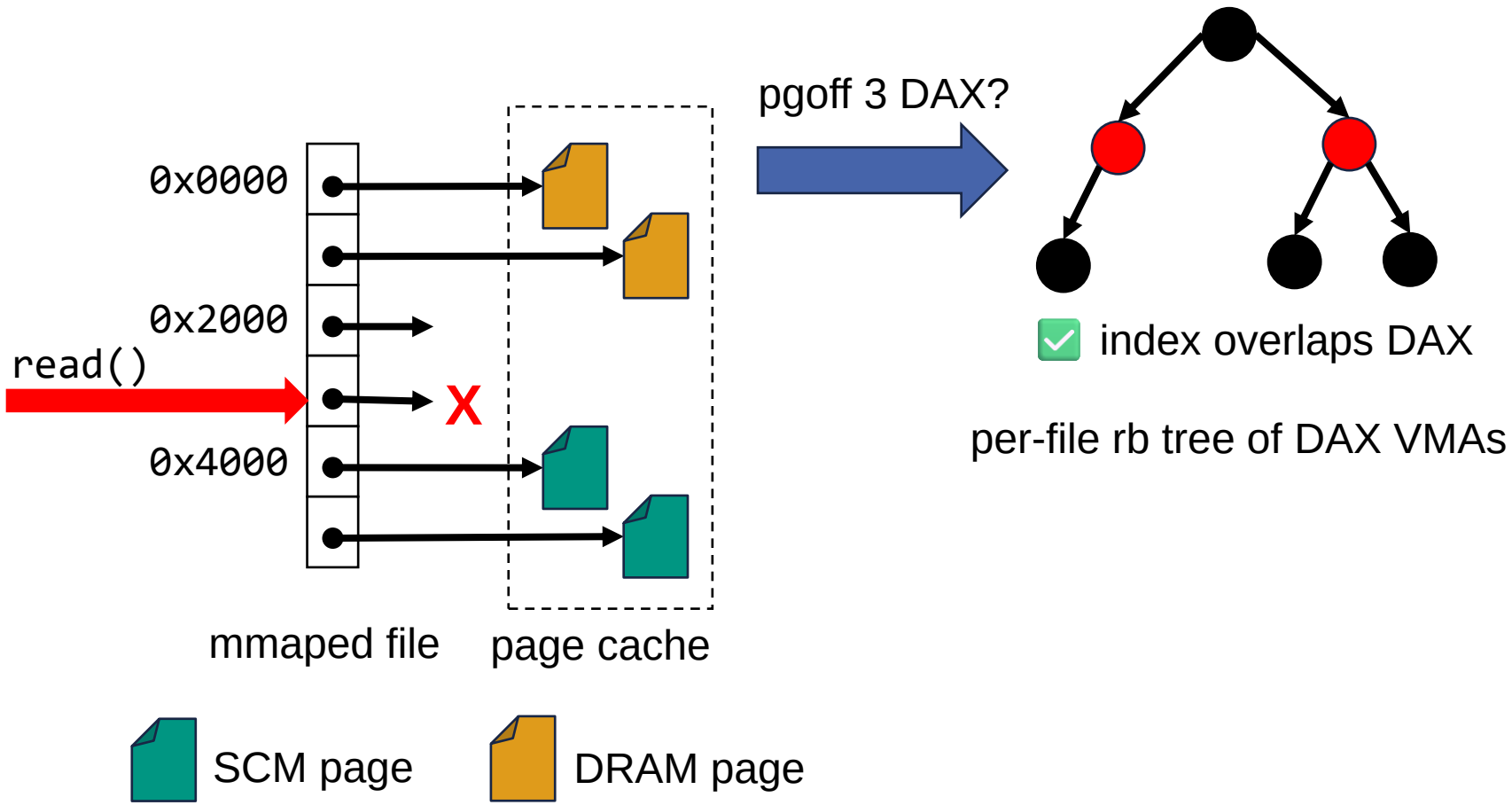
Persistency-aware Page Cache



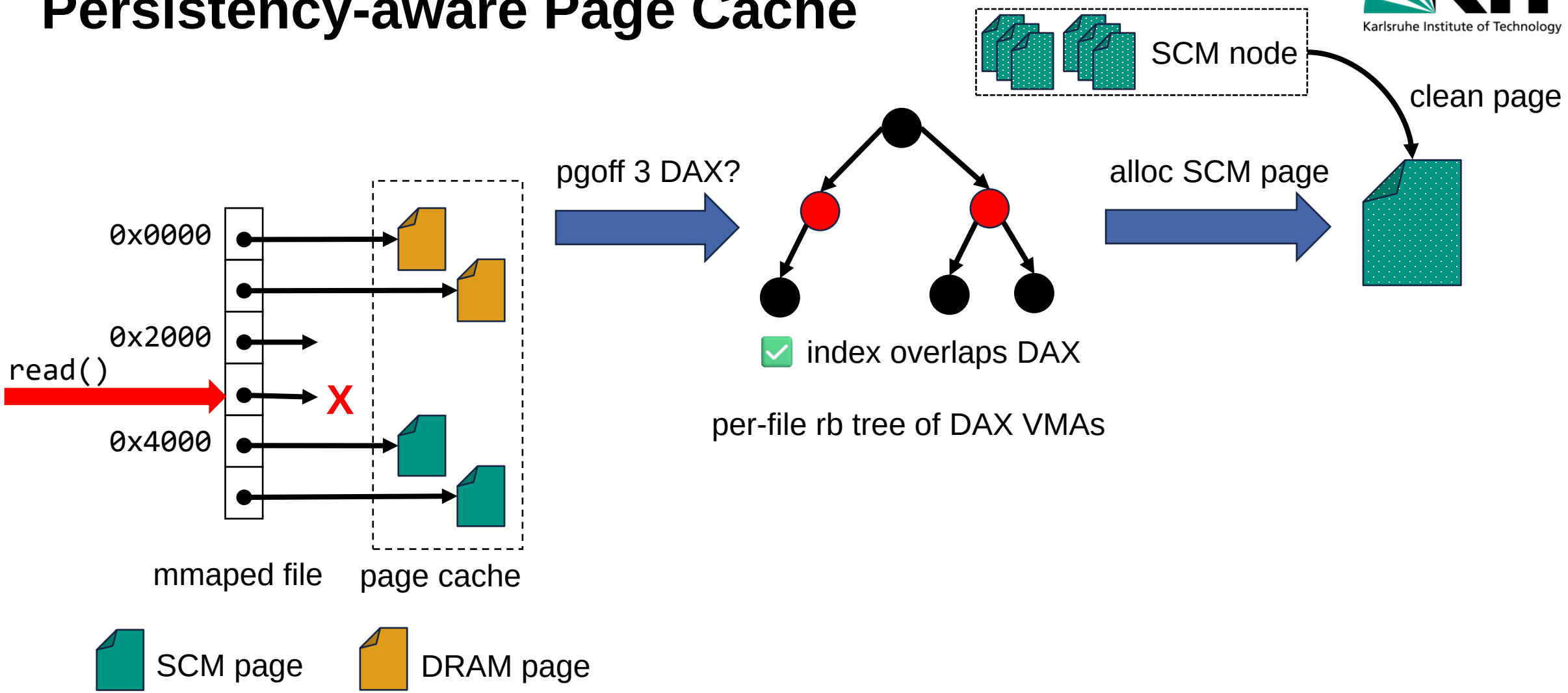
Persistency-aware Page Cache



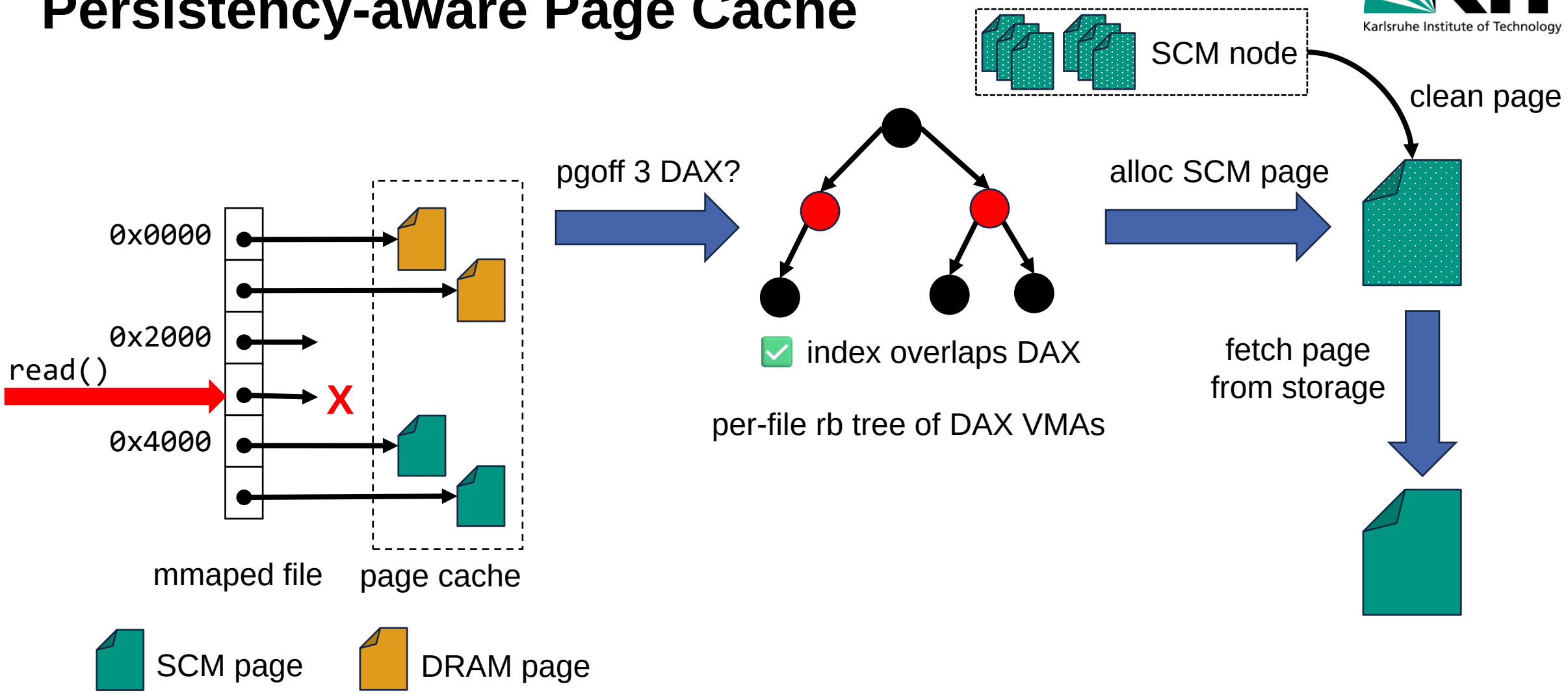
Persistency-aware Page Cache



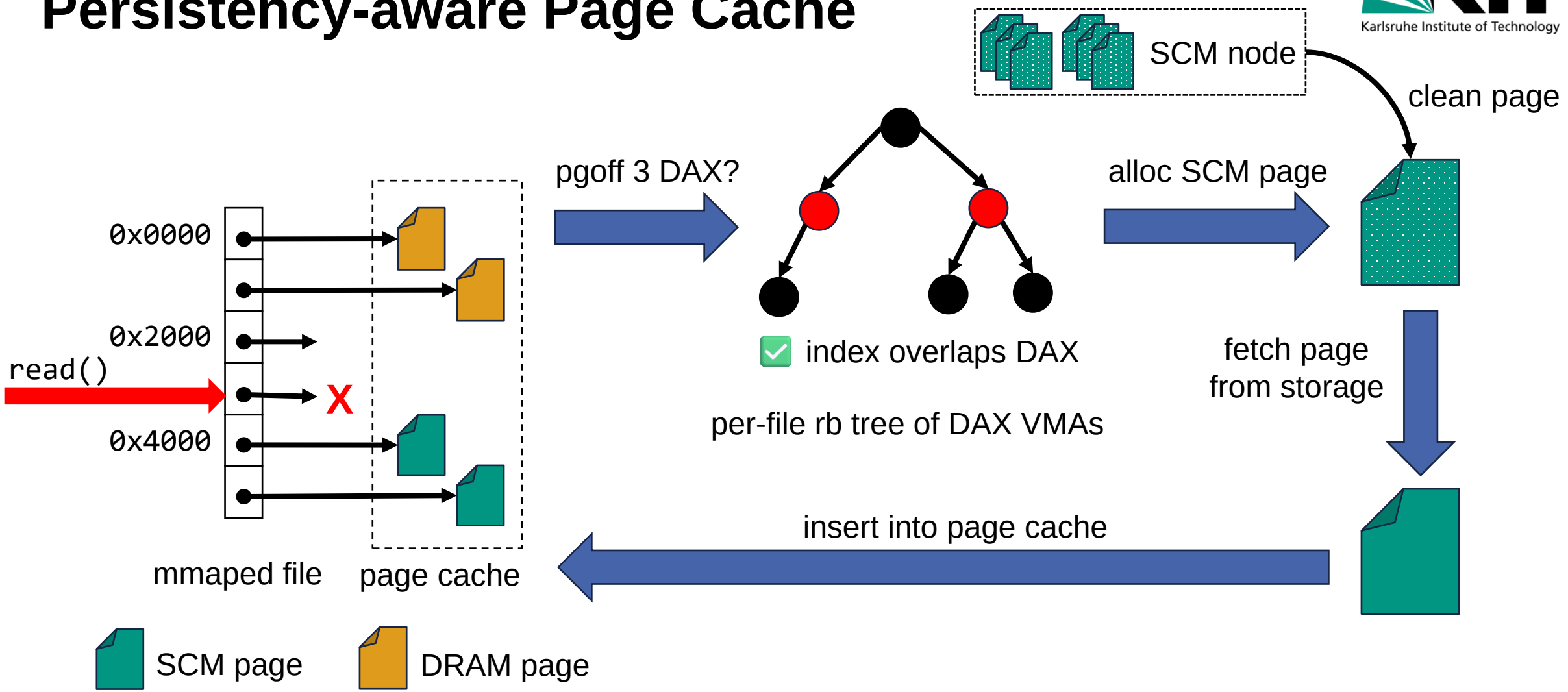
Persistency-aware Page Cache



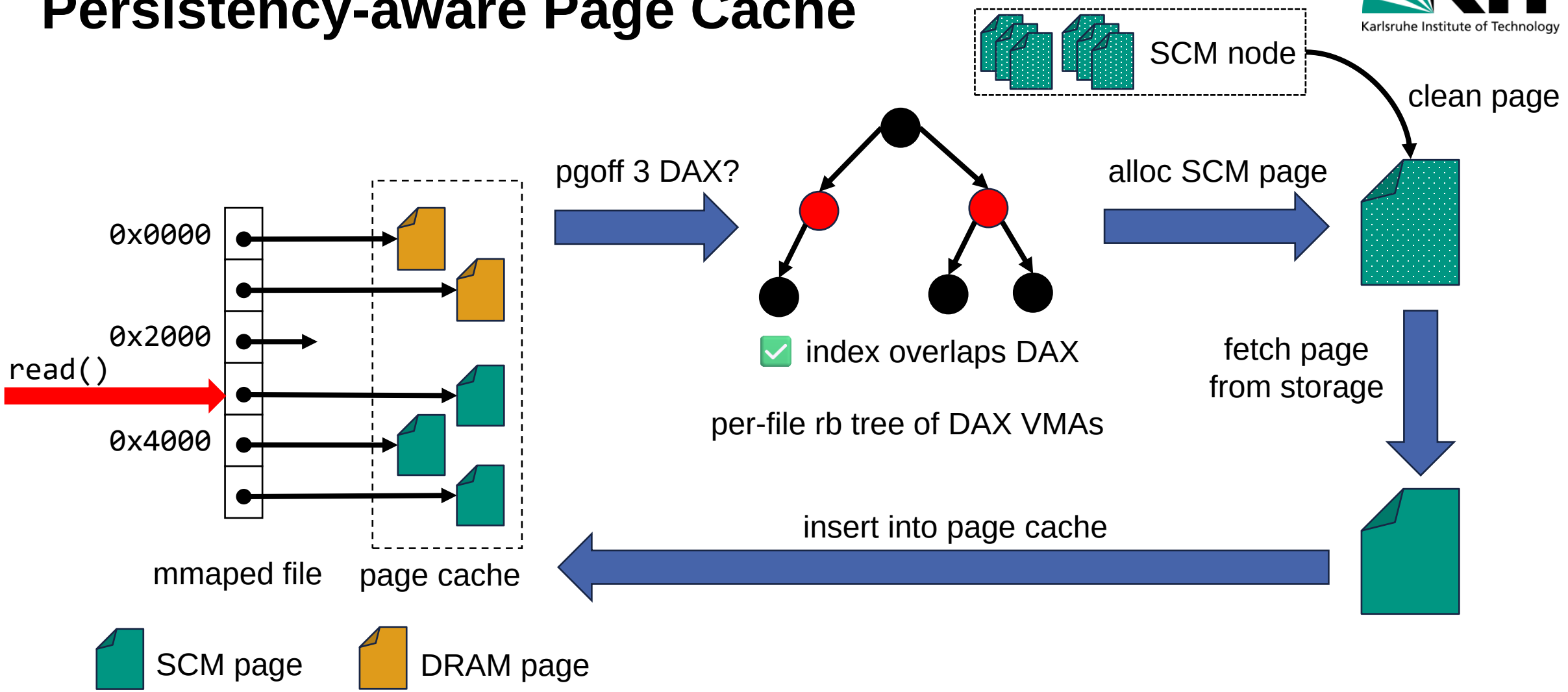
Persistency-aware Page Cache



Persistency-aware Page Cache

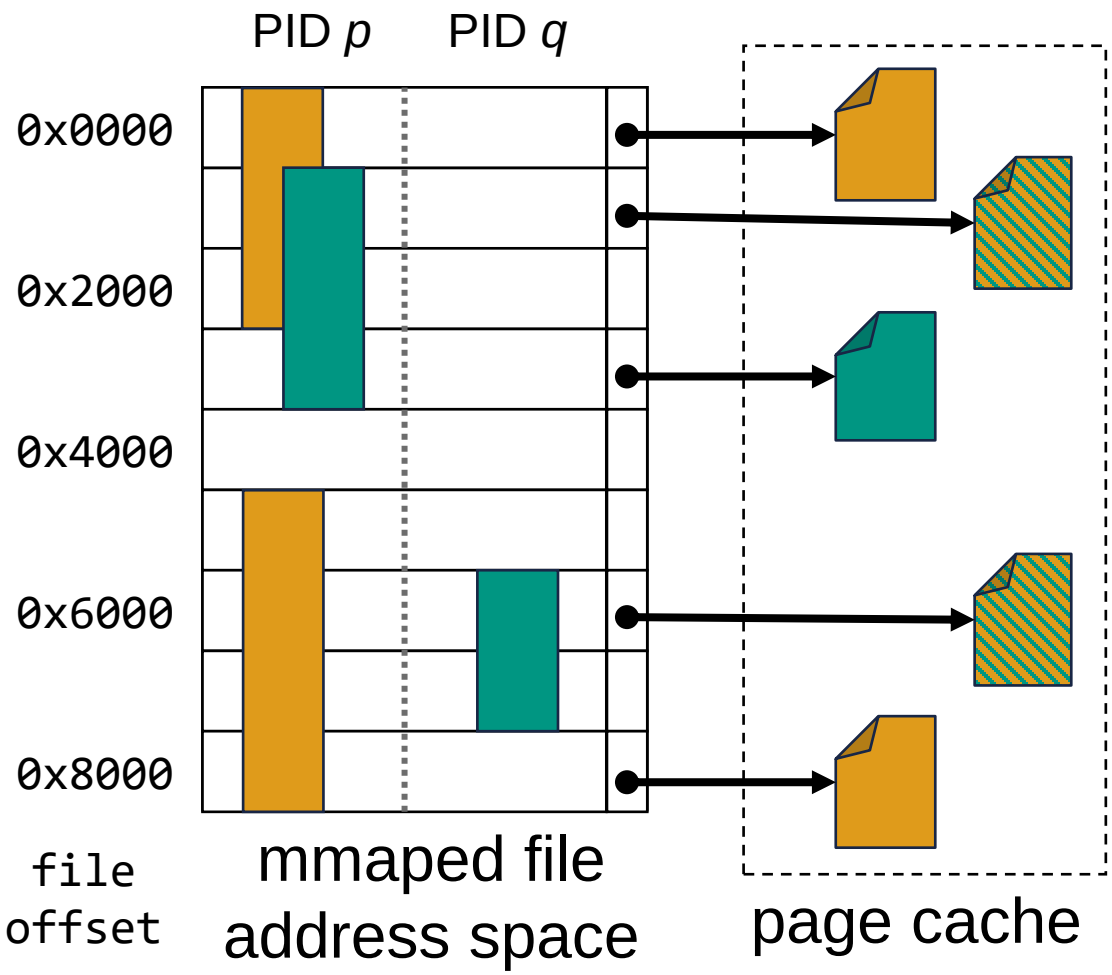


Persistency-aware Page Cache



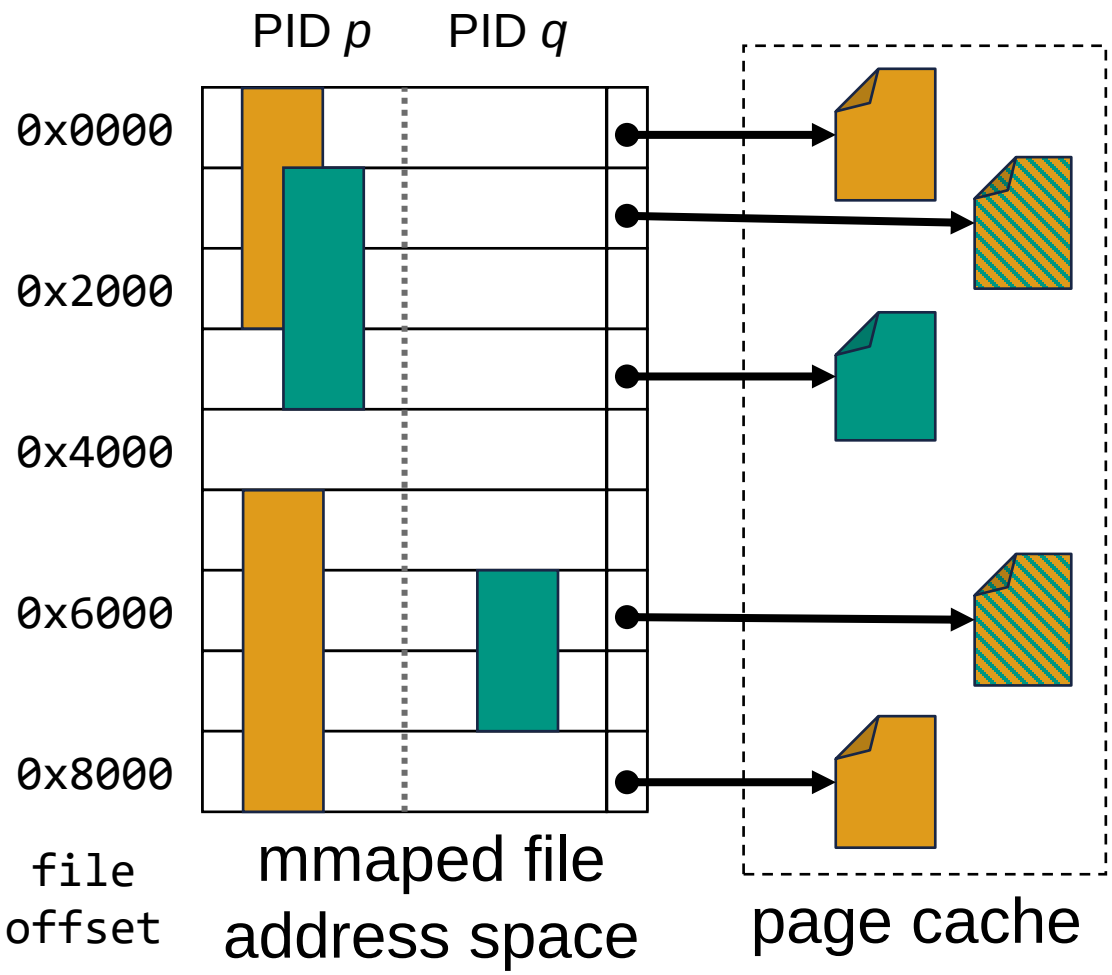
Retaining Coherency of Page Cache

- **DAX** and **non-DAX** VMAs might overlap



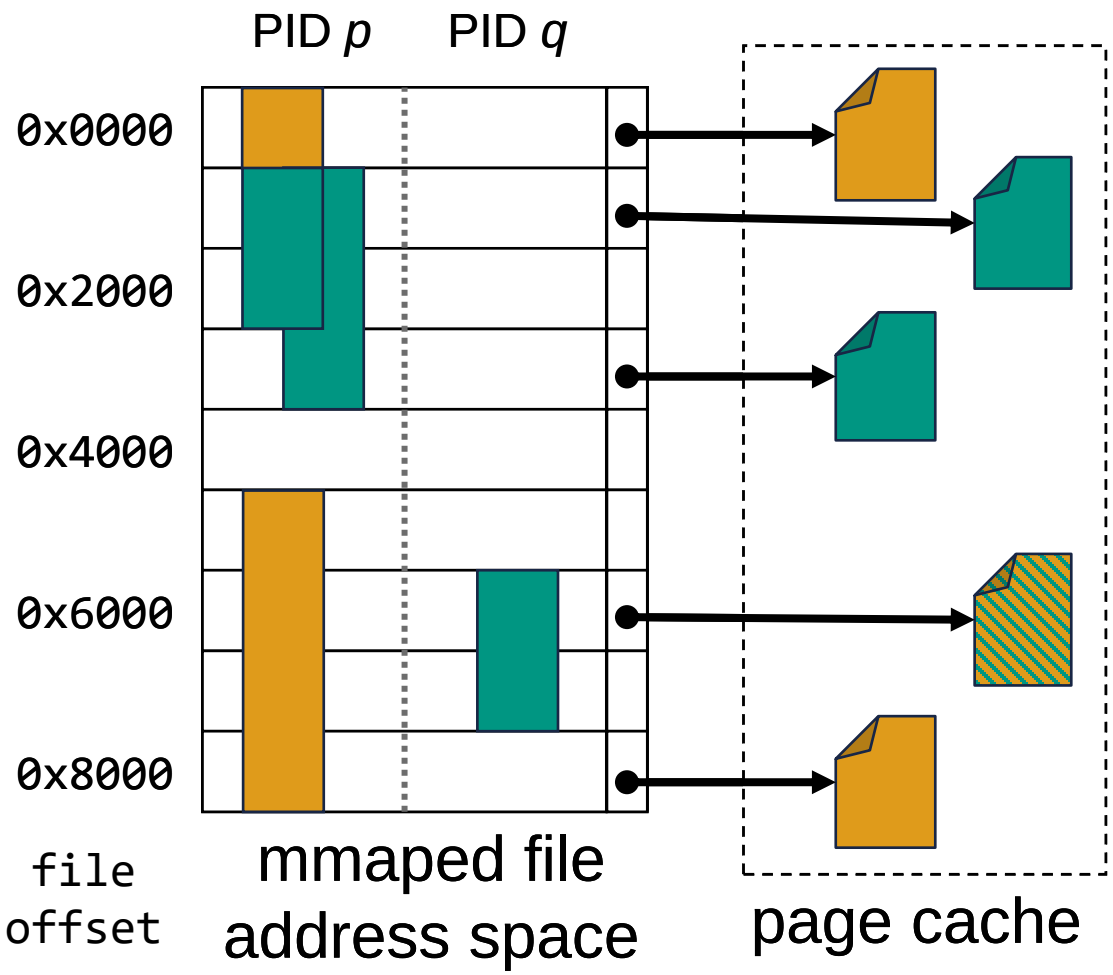
Retaining Coherency of Page Cache

- **DAX** and **non-DAX** VMAs might overlap
- Split VMAs and upgrade to DAX



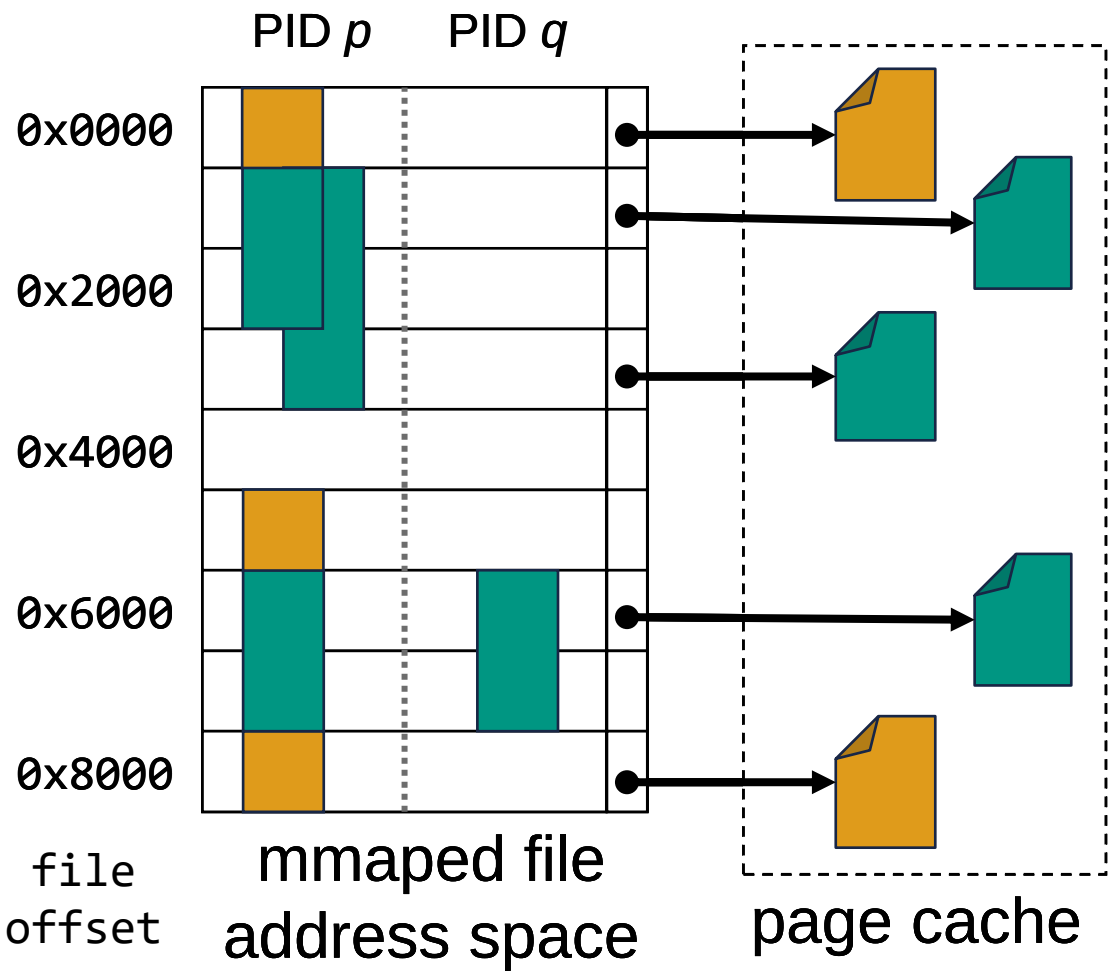
Retaining Coherency of Page Cache

- **DAX** and **non-DAX** VMAs might overlap
- Split VMAs and upgrade to DAX



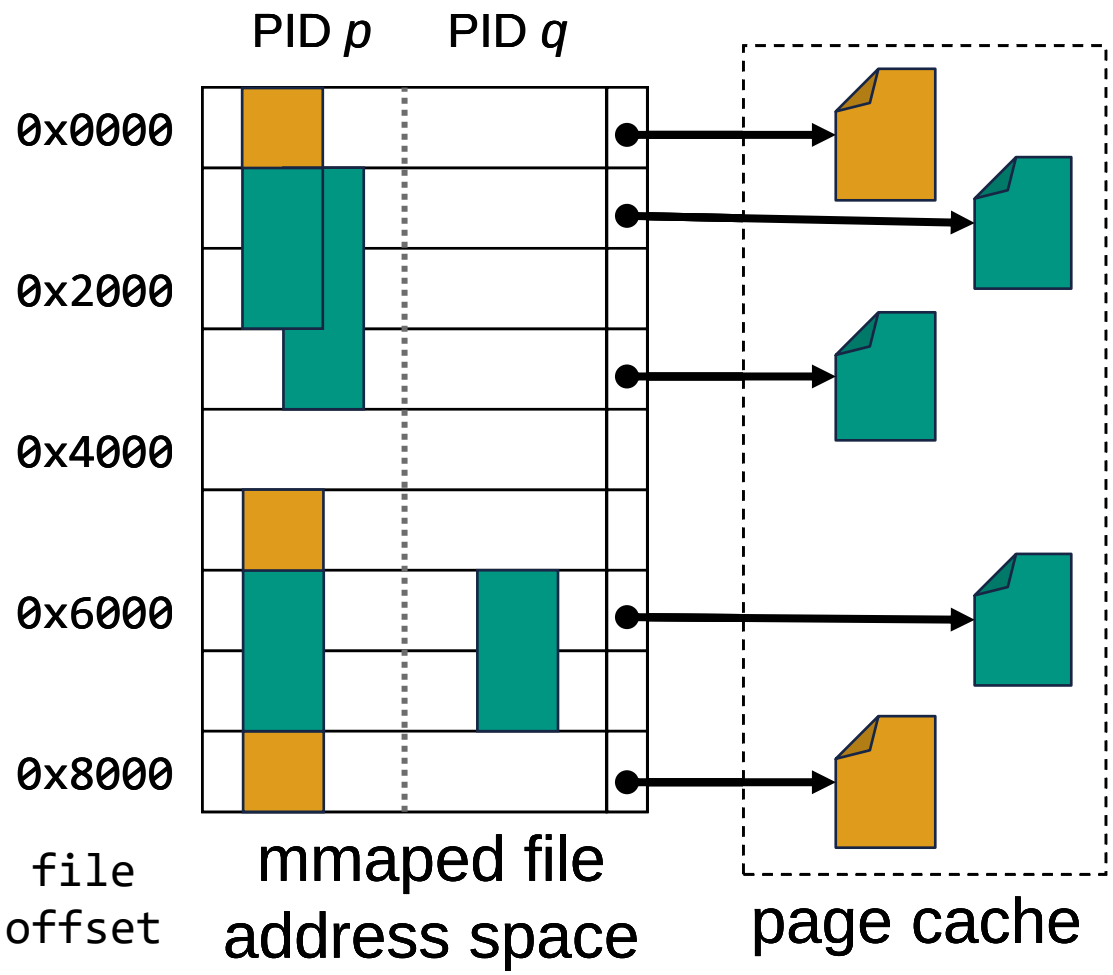
Retaining Coherency of Page Cache

- **DAX** and **non-DAX** VMAs might overlap
- Split VMAs and upgrade to DAX
- DAX upgrade affects other processes
  - VMA splitting might fail (`mmap` limit)
  - Locking difficult (cyclic dependency)



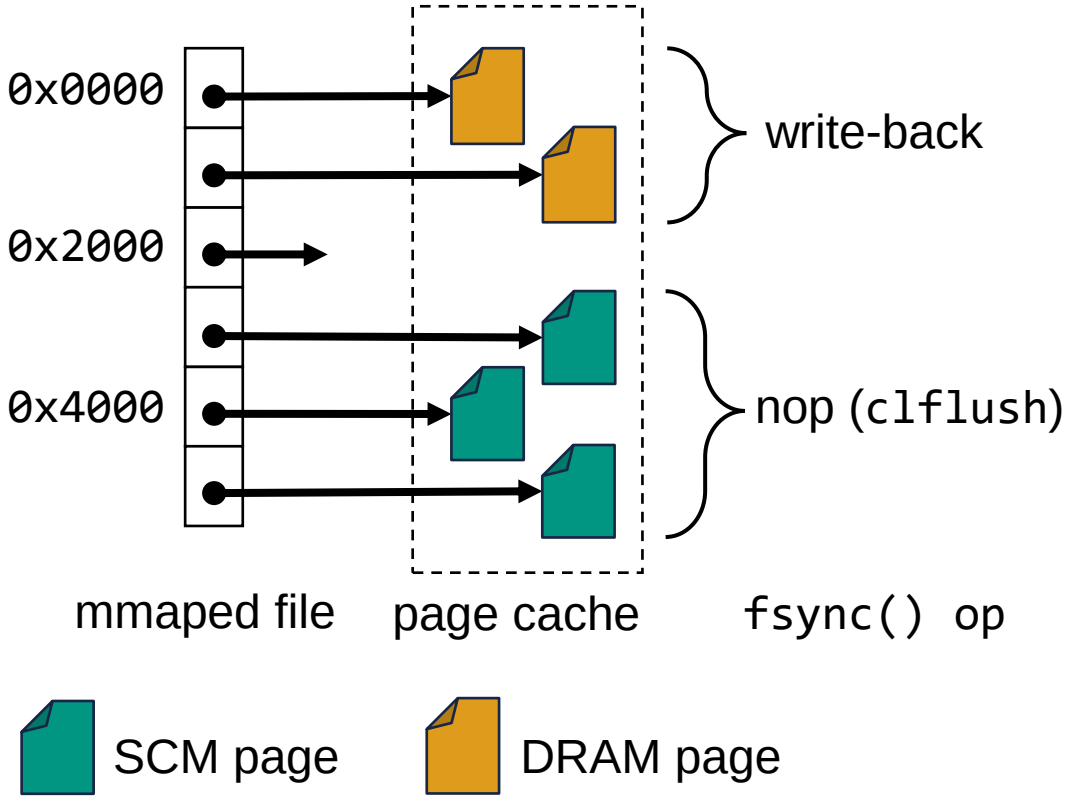
Retaining Coherency of Page Cache

- **DAX** and **non-DAX** VMAs might overlap
- Split VMAs and upgrade to DAX
- DAX upgrade affects other processes
  - VMA splitting might fail (`mmap` limit)
  - Locking difficult (cyclic dependency)
- Undo upgrade when possible?



Bypassing Synchronous Write-Back

- Synchronous writeback critical for performance
  - →skip write-back of SCM pages
    - SCM pages remain dirty
    - SCM guarantees persistence
- Asynchronous write-back unchanged
  - Performance not critical
  - Clean pages beneficial for reclaim
- Dynamically upgrade frequently synced file ranges to SCM



What's Next

- Transparent DAX Mappings (TDM)
  - Kernel dynamically maps SCM to user space
  - TDM-aware libc implements zero-copy read/write in user space
  - Improve performance and power usage

- CXL hybrid storage prototype
  - FPGA-based
  - OpenExpress (NVMe development platform)
  - Comparison to Samsung's CMM-H

"OpenExpress"
M. Jung (ATC '20)

Summary

- Upcoming CXL hybrid storage
  - Asynchronous block I/O
  - Synchronous load/store
- Existing OS support lacking
- Linux support for hybrid storage
  - Persistency-aware page cache
  - Bypass synchronous write-back
  - TDMs for better SCM utilization