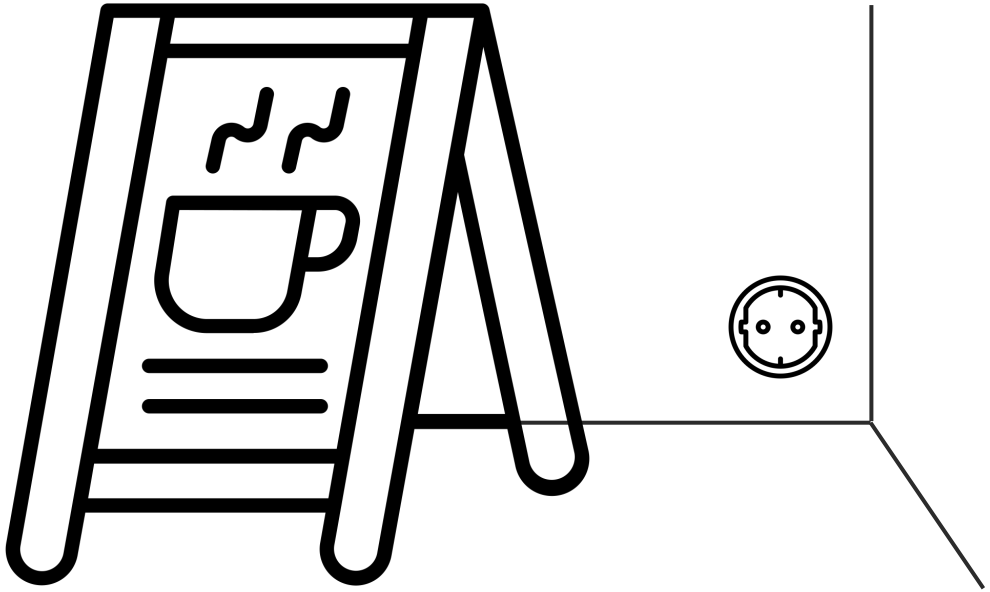
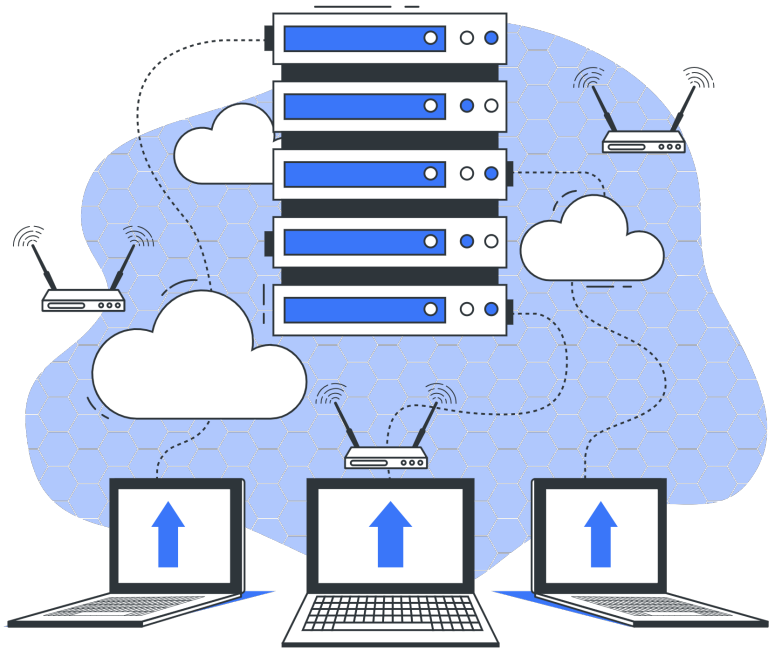
Taming the Linux Memory Allocator for Rapid Prototyping

Ruiyi Zhang, *Tristan Hornetz, Lukas Gerlach, Michael Schwarz*

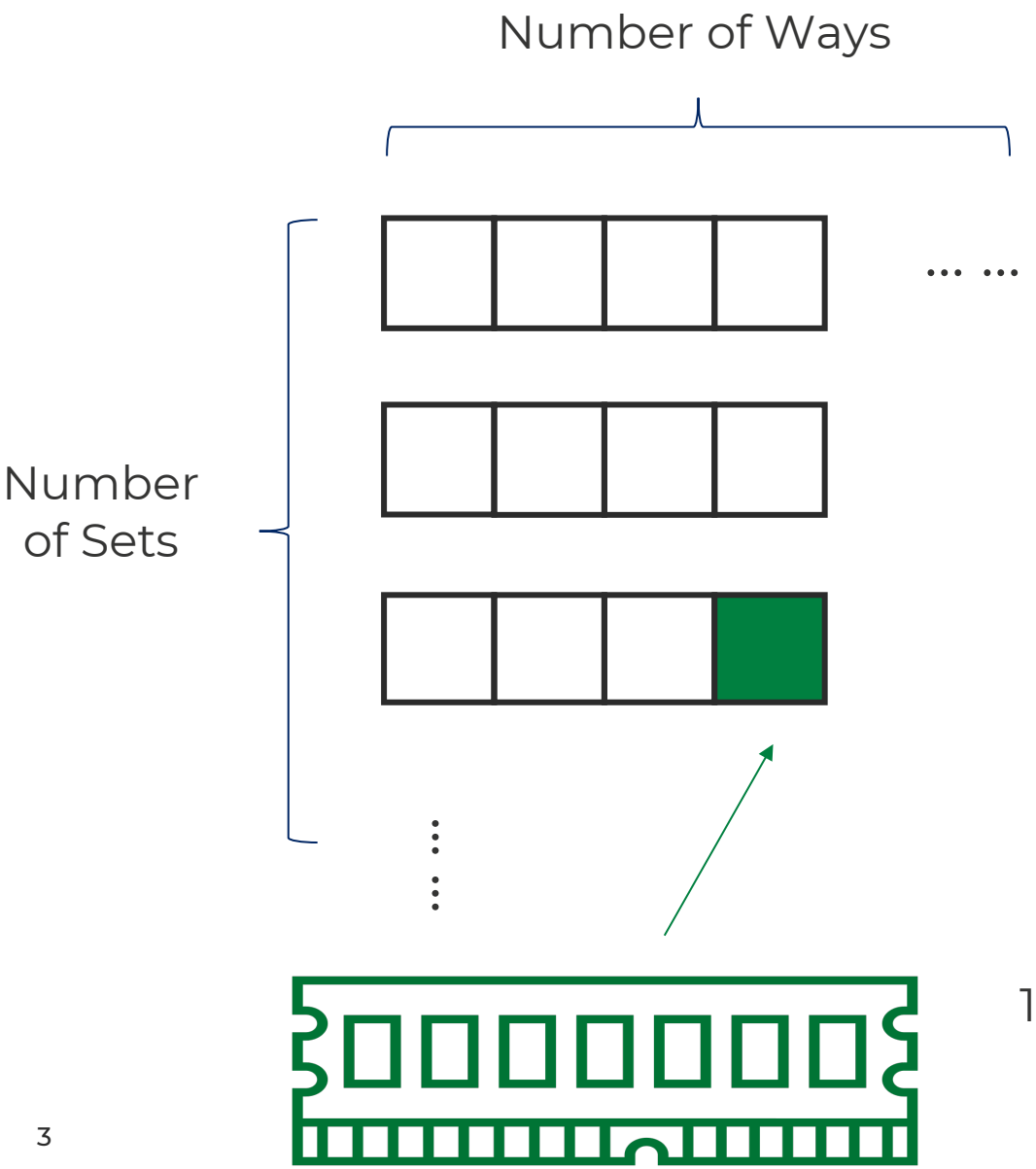
DIMVA, July 2025

Contention in Shared Spaces

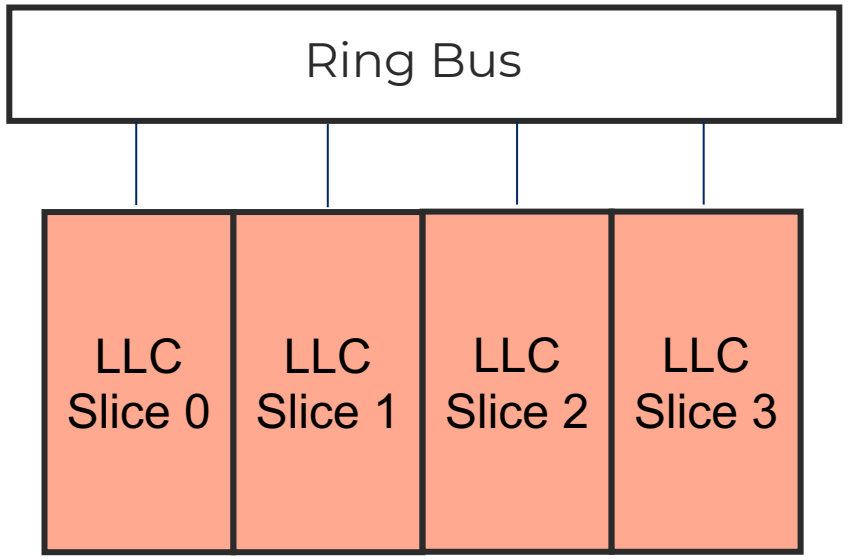




CPU Cache



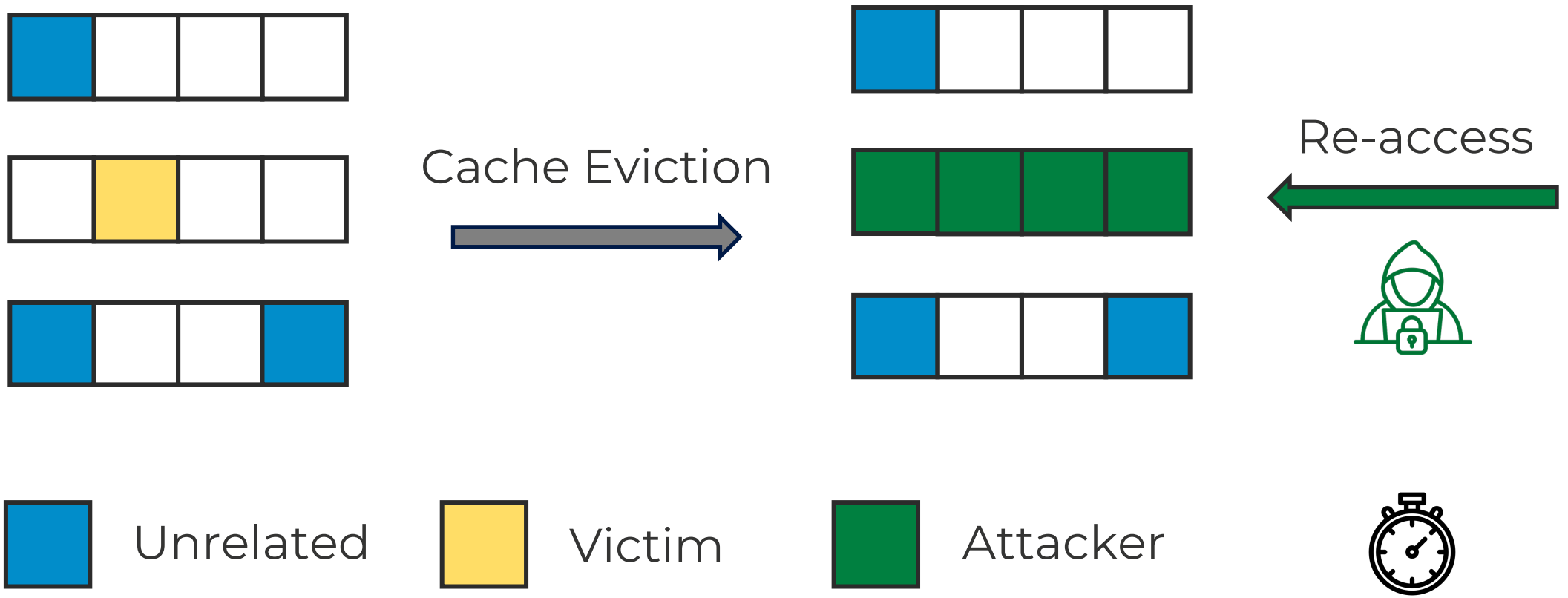
: 64 Byte



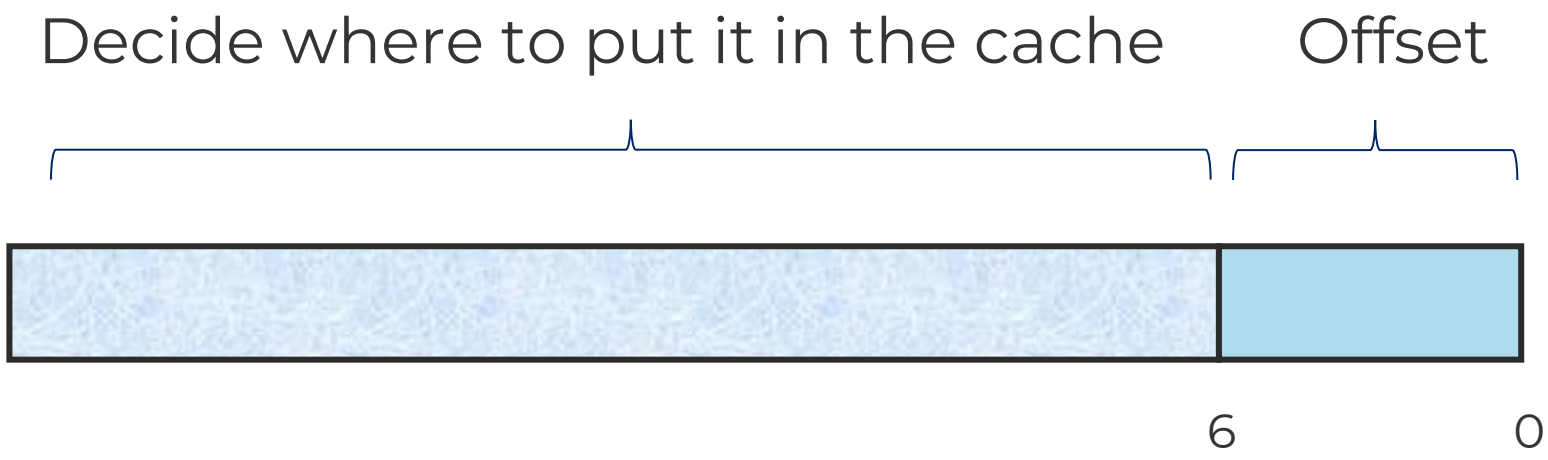
16MB Cache = 8 Slices ✕ 2048 Sets ✕ 16 Ways ✕ 64 Byte

- AMD EPYC 9124

Cache Attacks

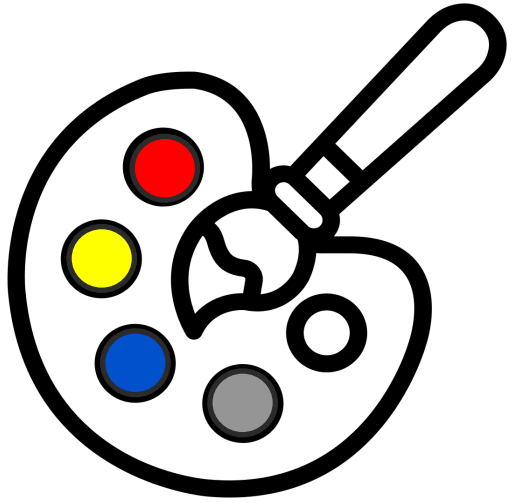


Cache – Hash Function



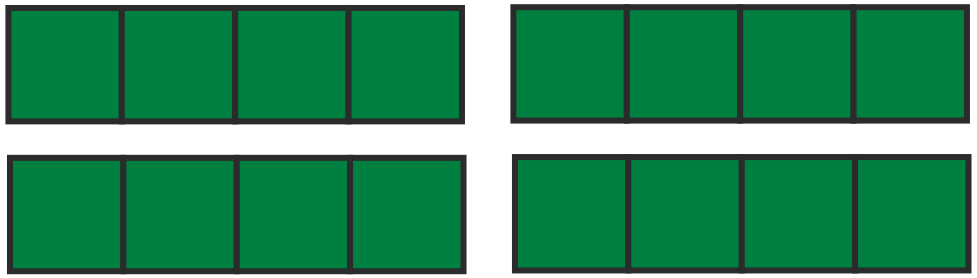
Physical Memory Address

Memory-allocation-based Defense

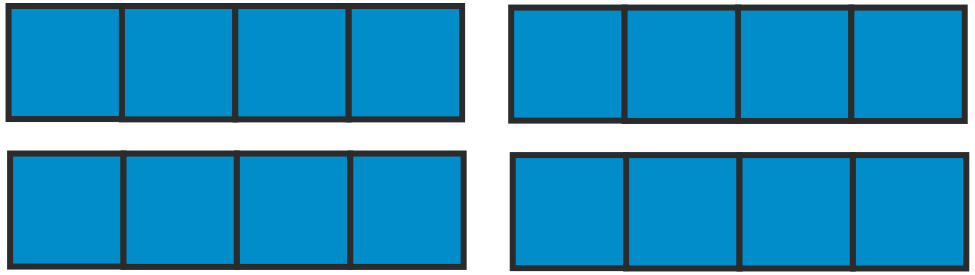


- Page Coloring

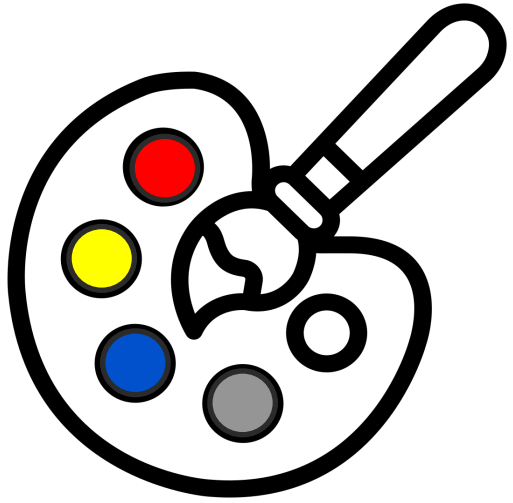
Process A



Process B

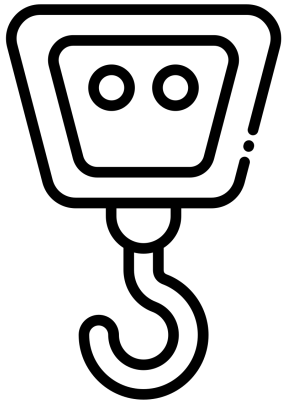


Memory-allocation-based Defense



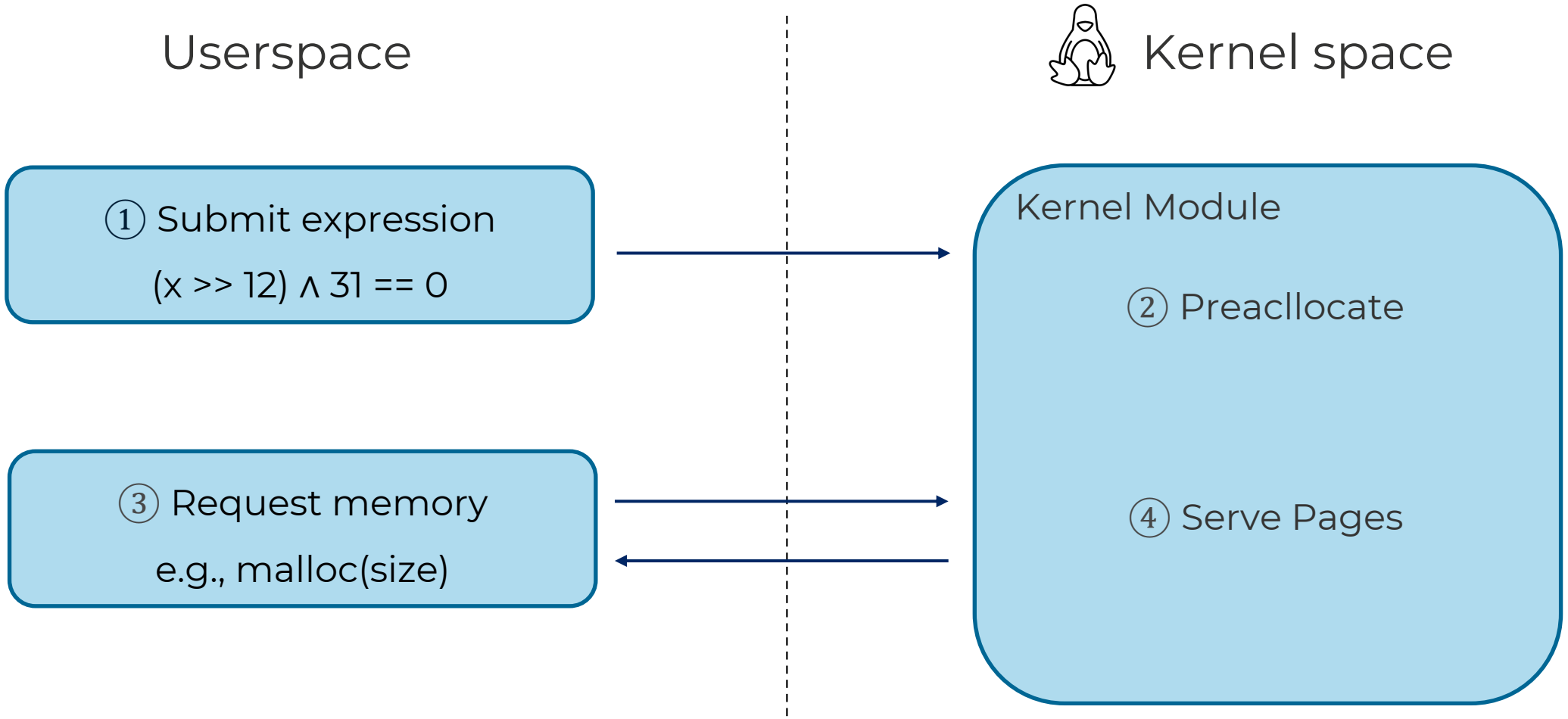
- Page Coloring
- Require complex changes to the kernel
- Hardware modification takes very long

MAPAlloc Framework



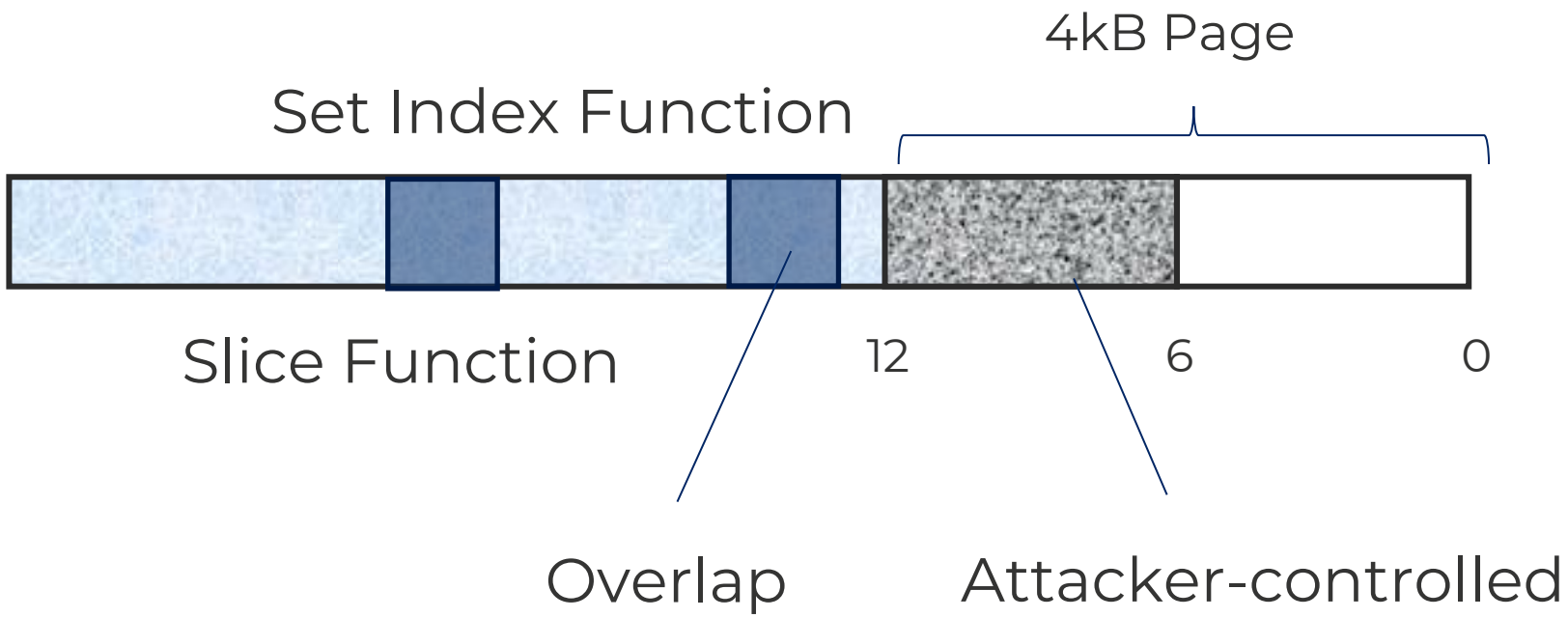
- A flexible, lightweight framework:
  - Users specify color constraints in DSL
  - Hooks into Linux's memory allocator
    - `*Kretprobe*`
    - `*__alloc_pages*`, `*vm_mmap_pgoff*`
  - Cross-architecture support (x86, ARMv8, RISC-V)

How MAPAlloc Works



Layered Coloring

- Last-level cache is also divided into slices

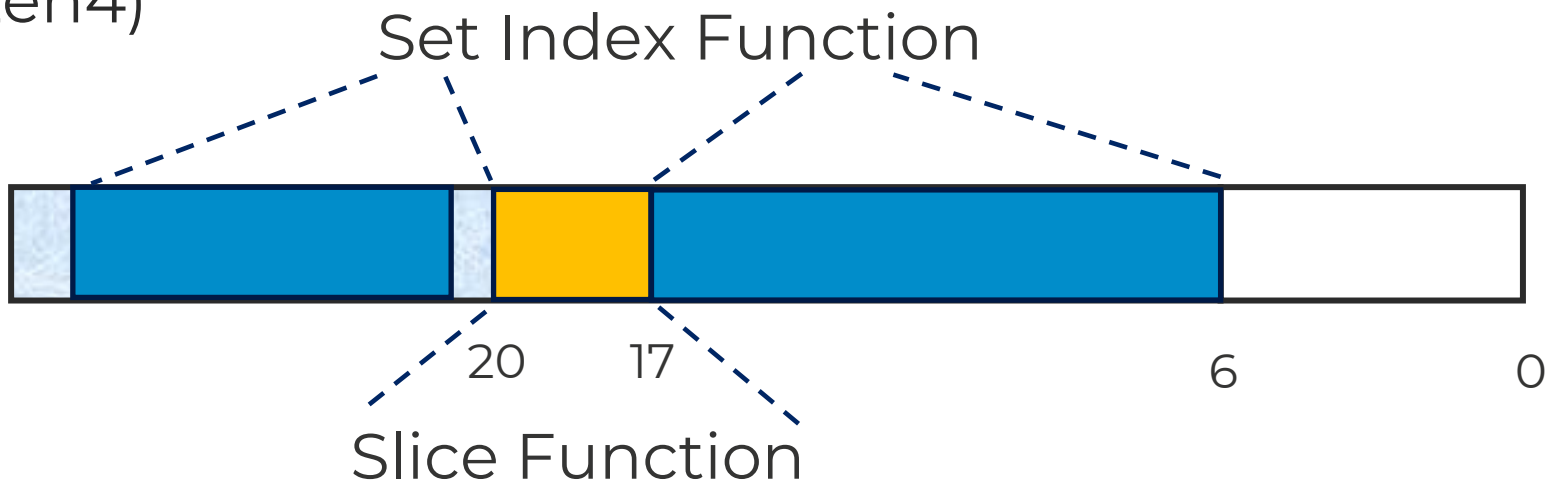


Layered Coloring

AMD EPYC 9124 (Zen4)

- 8 slices

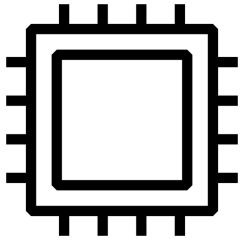
- 2048 sets



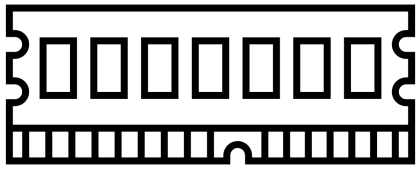
$$i(a) = a_6,\ a_7,\ a_8,\ a_9 \oplus a_{28} \oplus a_{29},\ a_{10} \oplus a_{27} \oplus a_{30},\ a_{11} \oplus a_{26} \oplus a_{31}$$
$$a_{12} \oplus a_{25} \oplus a_{32},\ a_{13} \oplus a_{24} \oplus a_{33},\ a_{14} \oplus a_{23} \oplus a_{34},\ a_{15} \oplus a_{22} \oplus a_{35},\ a_{16} \oplus a_{21} \oplus a_{36}$$
$$h(a) = a_{17},\ a_{18},\ a_{19}$$

$2^5 \times 2^3 = 256$ layerd colors

Case Study



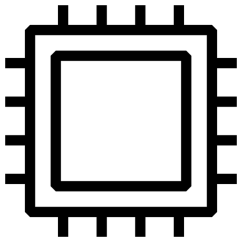
Mitigate
Prime+Probe



Mitigate DRAMA

Isolation

Incomplete Non-linear
Slice Function



Prototype
Prime+Probe

Memory Restriction

Performance Overhead

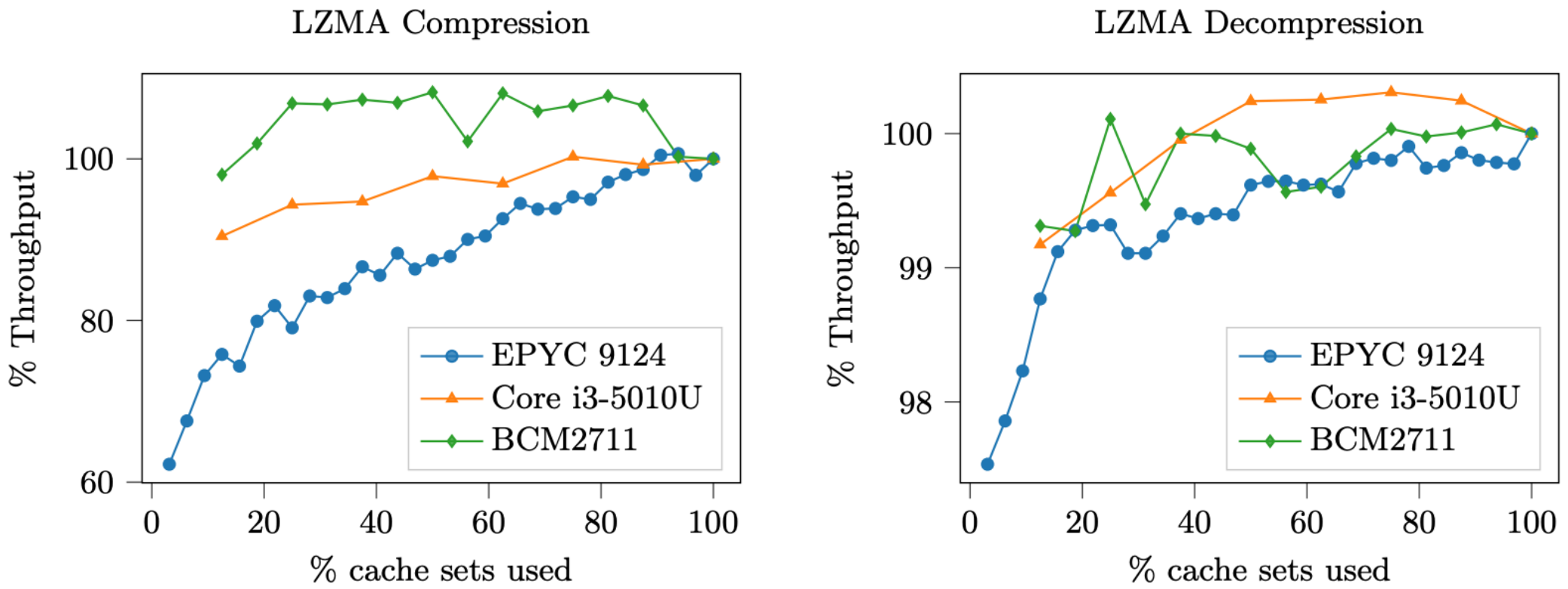


Fig. 4: Mean LZMA de/compression throughput with limited cache access, normalized by baseline performance (7-zip v23.01, 1 thread)

Performance Overhead

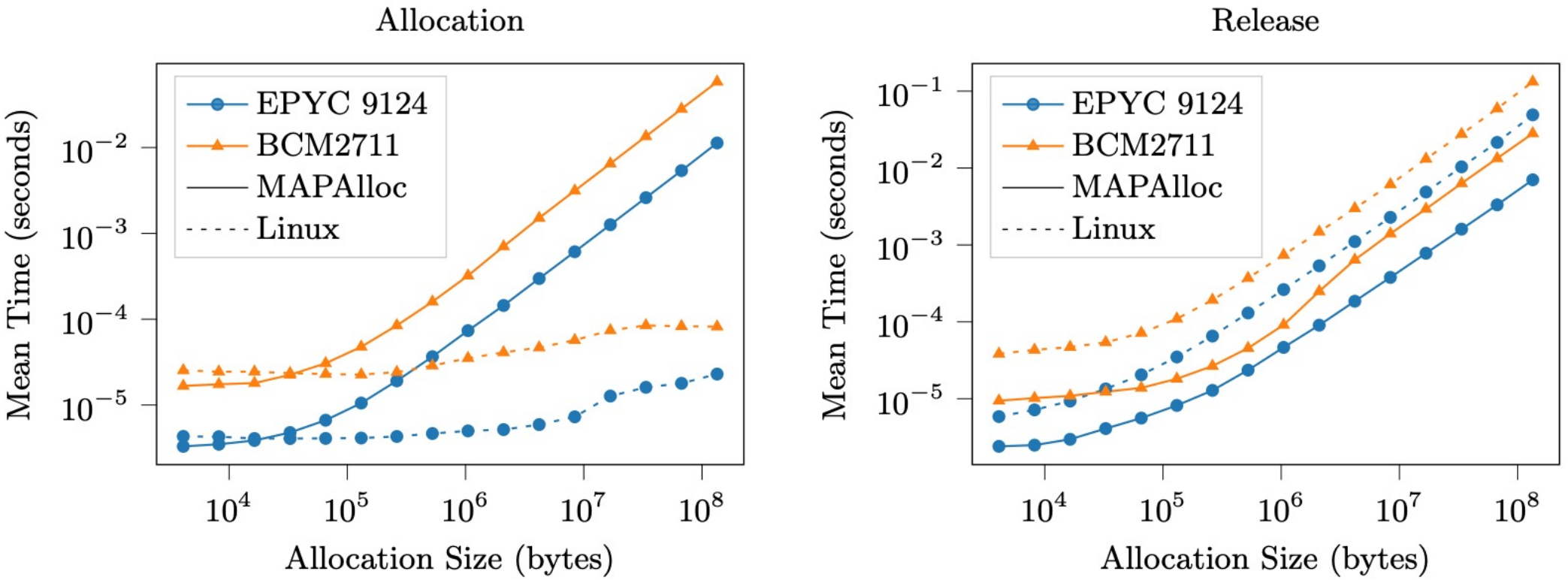


Fig. 3: Mean time to allocate and release memory blocks of different sizes with MAPAlloc and Linux ($n = 4096$ per sample, Linux v6.8.0). Less is better.

Takeaways

We present MAPAlloc for rapid prototyping μarch security research:

– Prototyping memory allocation-based defenses and attacks

– Introducing layered page coloring

– Cross-architecture support (x86, ARMv8, RISC-V)

Contact:
ruiyi.zhang@cispa.de

Github:
https://github.com/cispa/MAPAlloc