ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE

ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL

ARTIFACT EVALUATED usenix ASSOCIATION REPRODUCED

EMT: An OS Framework for New Memory Translation Architectures

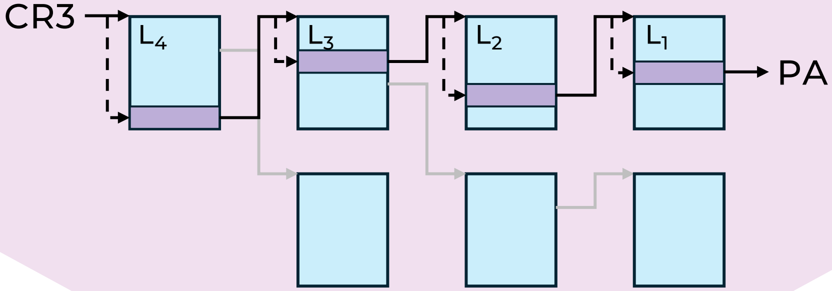
Siyuan Chai, Jiyuan Zhang, Jongyul Kim, Alan Wang,
Fan Chung, Jovan Stojkovic, Weiwei Jia, Dimitrios Skarlatos,
Josep Torrellas, Tianyin Xu

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

THE UNIVERSITY OF RHODE ISLAND

Carnegie Mellon University

Radix tree was the de facto translation design

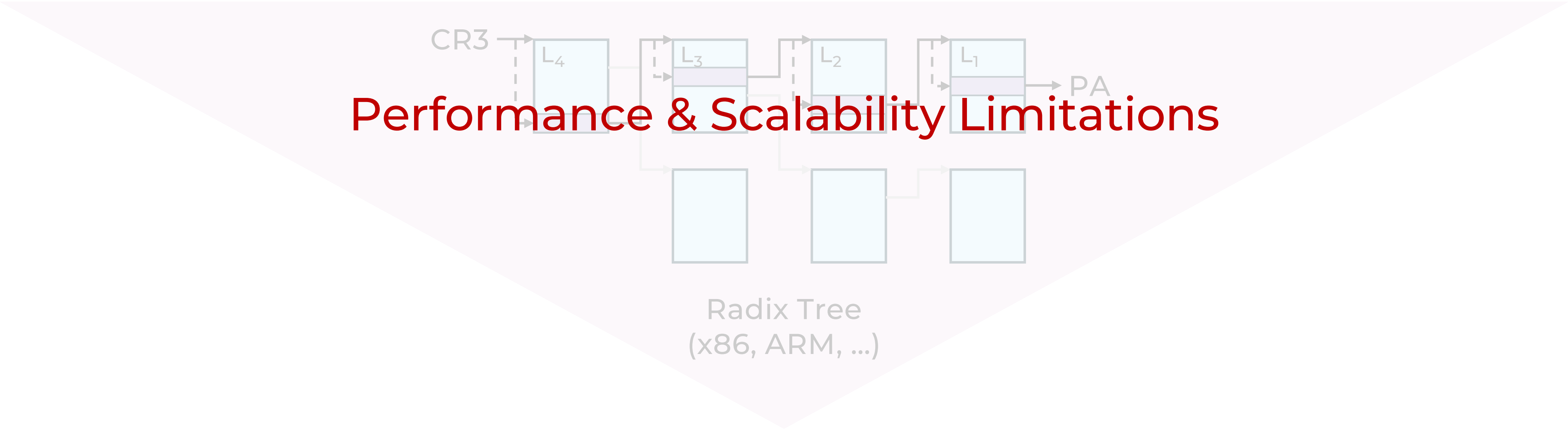


Radix Tree
(x86, ARM, …)

Today most commercial architectures
exclusively uses radix tree design.

x86, ARM64, RISC-V, LoongArch, s390, …

Radix tree was the de facto translation design

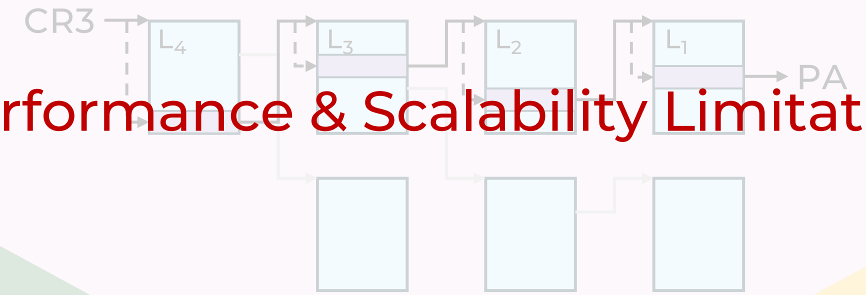


Performance & Scalability Limitations

Today most commercial architectures
exclusively uses radix tree design.

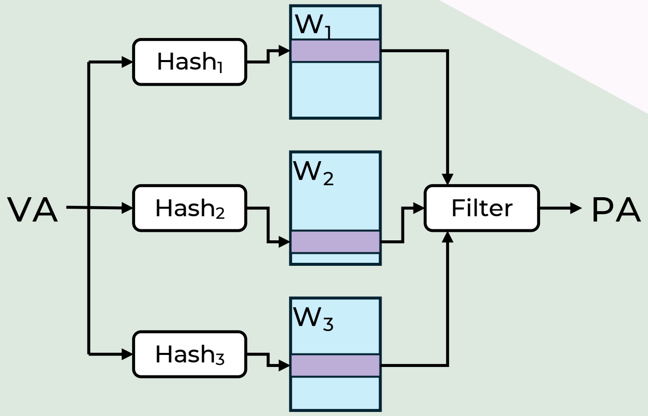
x86, ARM64, RISC-V, LoongArch, s390, ...

New translation architectures are emerging

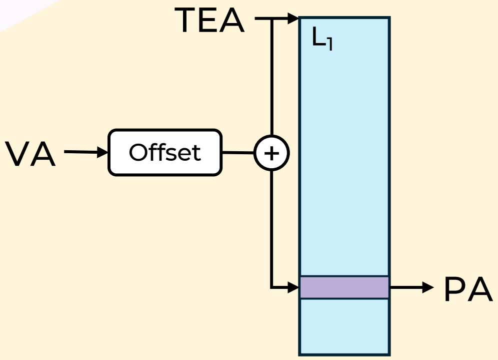


Performance & Scalability Limitations

Radix Tree
(x86, ARM, ...)

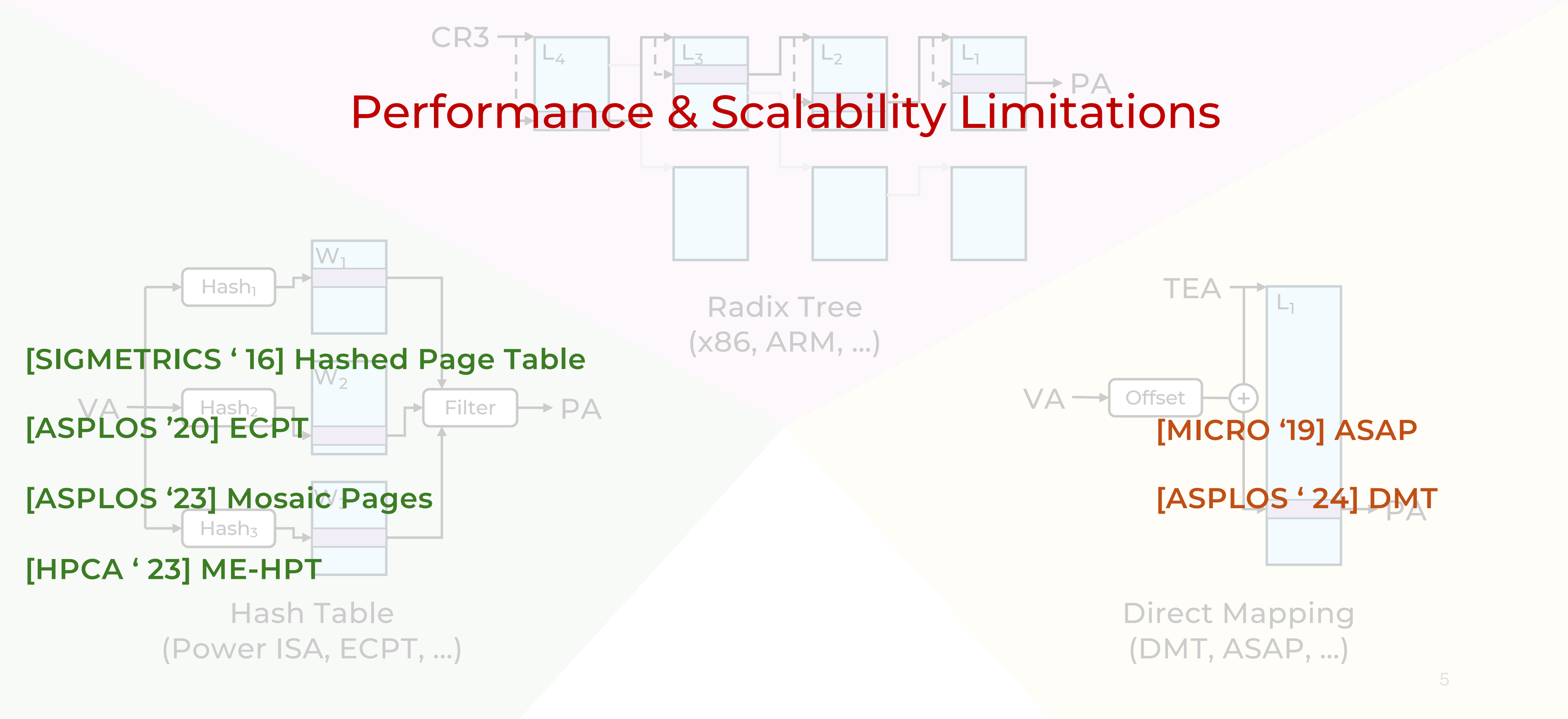


Hash Table
(Power ISA, ECPT, ...)



Direct Mapping
(DMT, ASAP, ...)

New translation architectures are emerging



Performance & Scalability Limitations

[SIGMETRICS ‘ 16] Hashed Page Table

[ASPLOS ’20] ECPT

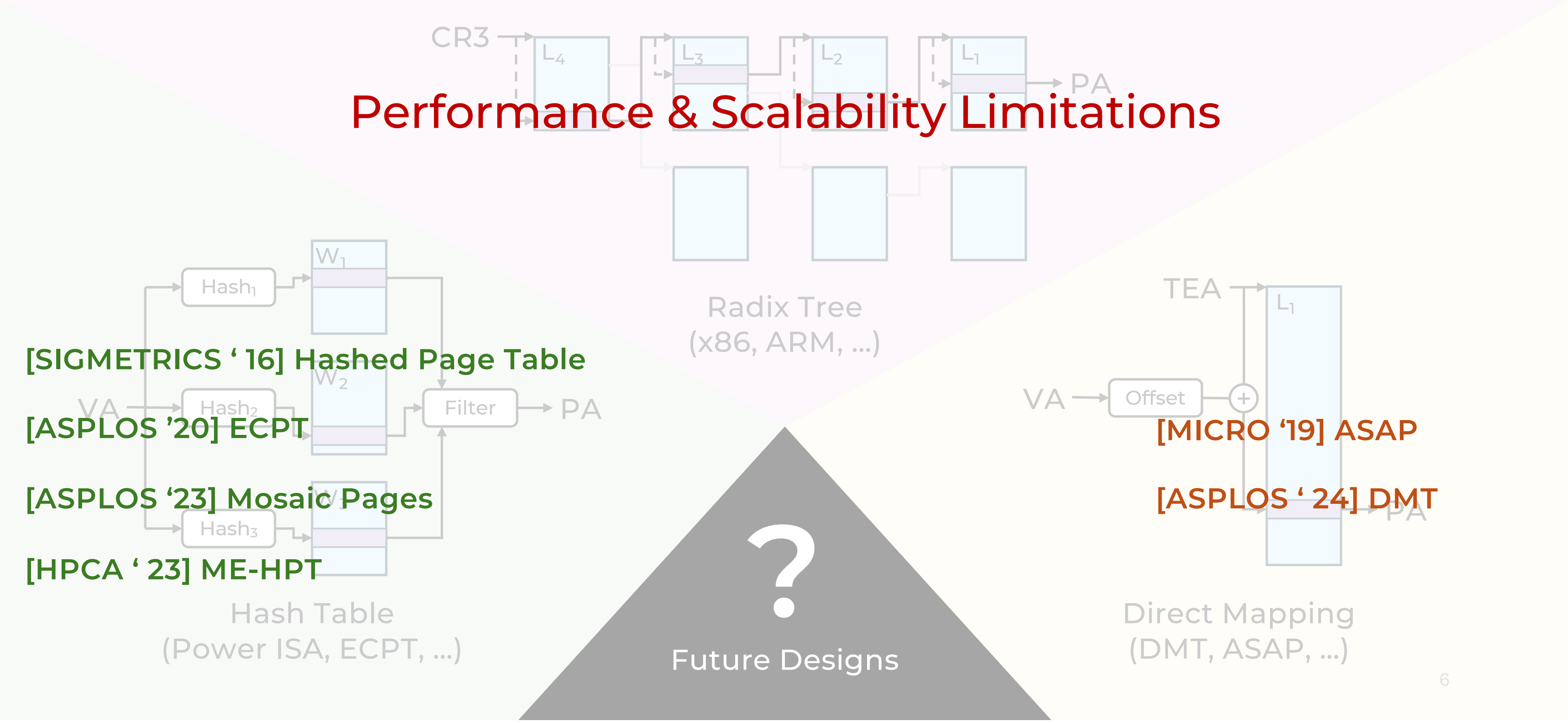
[ASPLOS ‘23] Mosaic Pages

[HPCA ‘ 23] ME-HPT

[MICRO ‘19] ASAP

[ASPLOS ‘ 24] DMT

New translation architectures are emerging



Performance & Scalability Limitations

[SIGMETRICS ‘ 16] Hashed Page Table

[ASPLOS ’20] ECPT

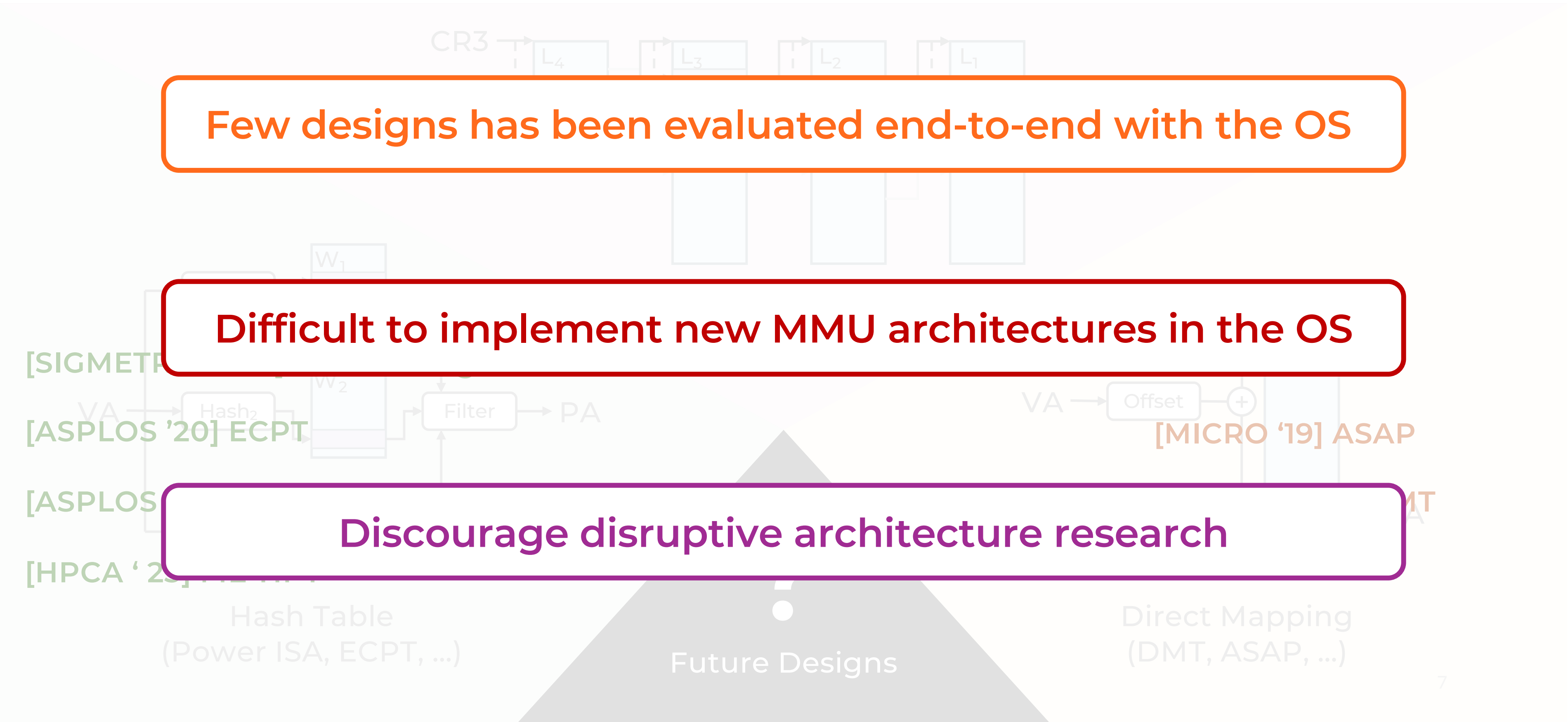
[ASPLOS ‘23] Mosaic Pages

[HPCA ‘ 23] ME-HPT

[MICRO ‘19] ASAP

[ASPLOS ‘ 24] DMT

The missed evaluation of new architectures

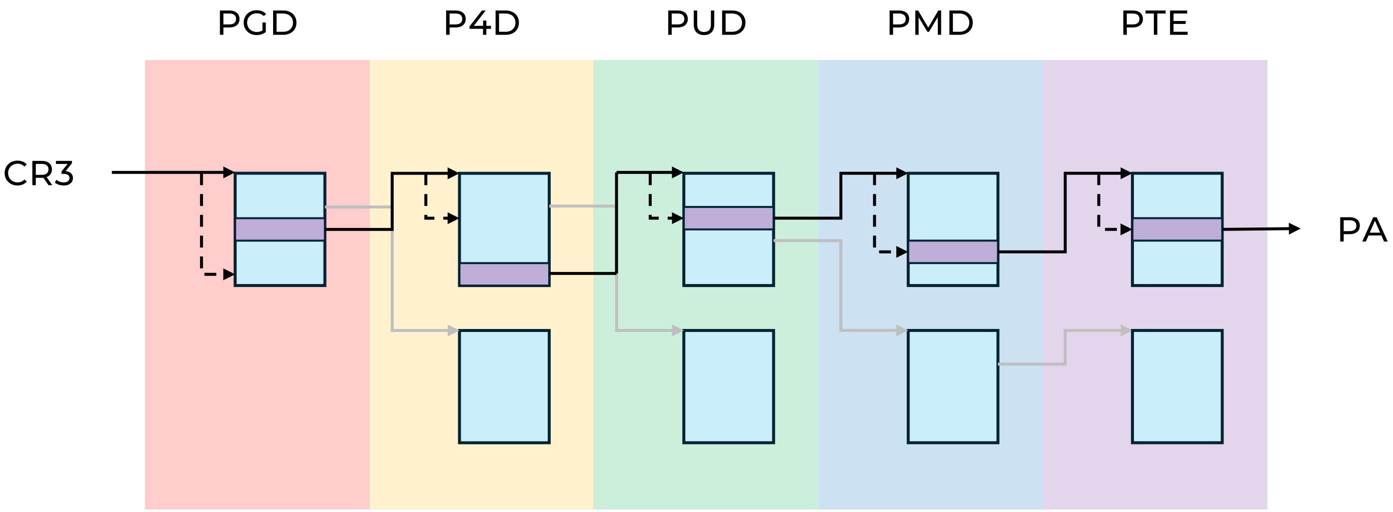


Few designs has been evaluated end-to-end with the OS

Difficult to implement new MMU architectures in the OS

Discourage disruptive architecture research

The Linux kernel assumes radix design

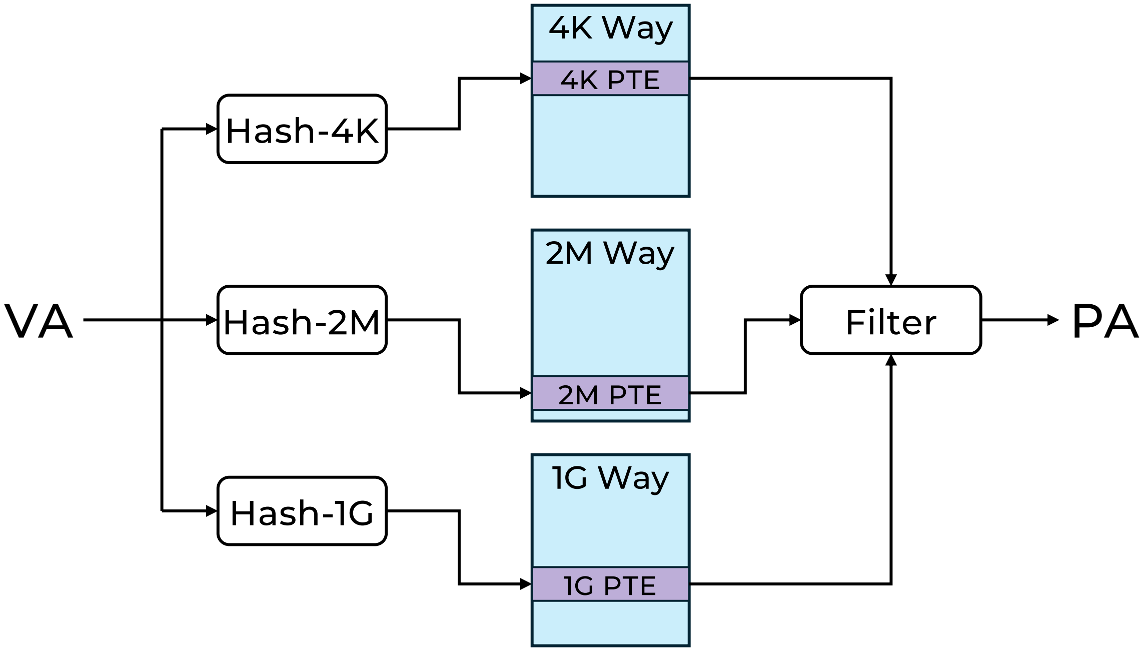


“It so happens that a tree format is the only sane format…”

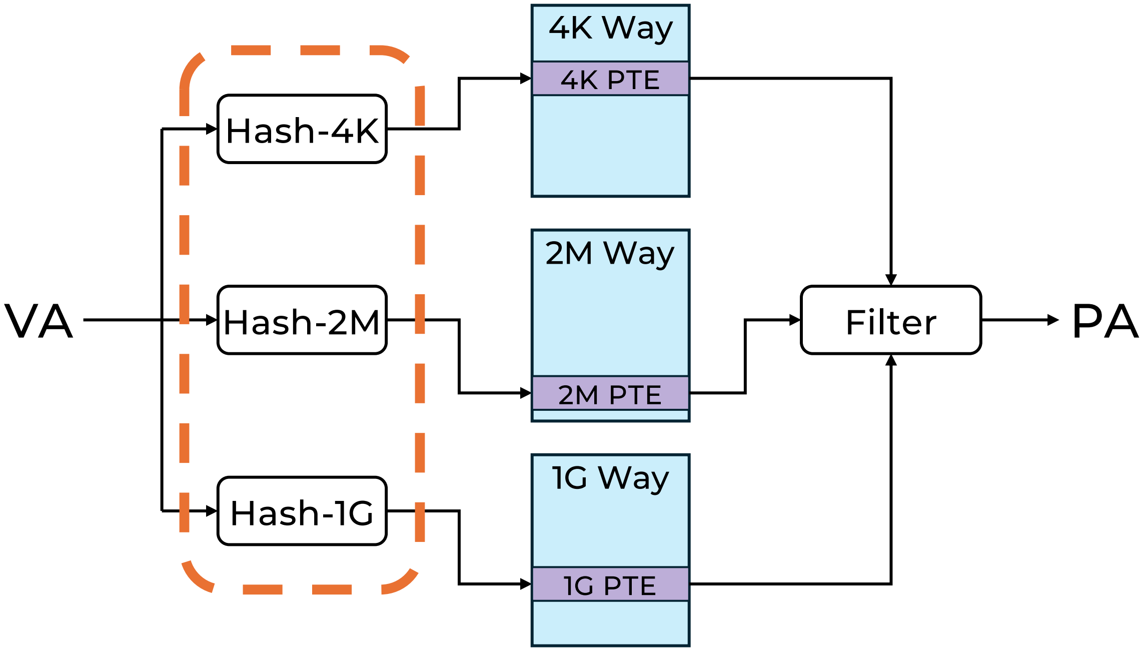
— Linus Torvalds, 2002

🤔

ECPT: A different design from radix schemes



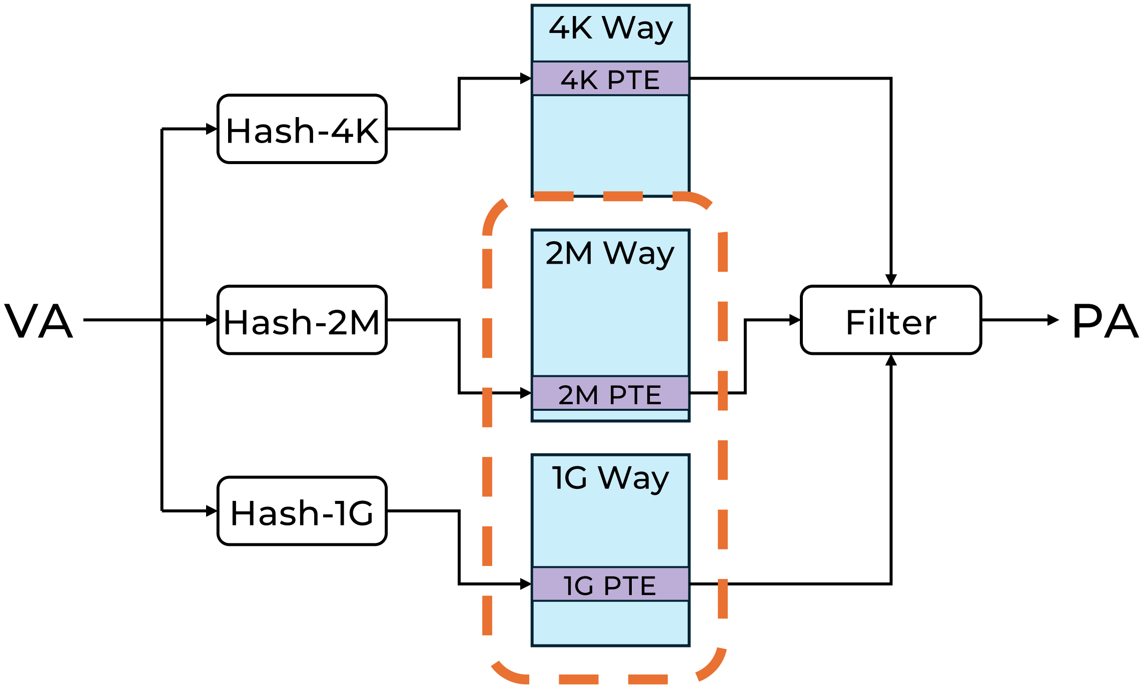
ECPT: A different design from radix schemes



+ Index with hash function values
- Index with bits in virt. address

Elastic Cuckoo Page Table (ECPT) vs. Radix-Tree Page Table

ECPT: A different design from radix schemes

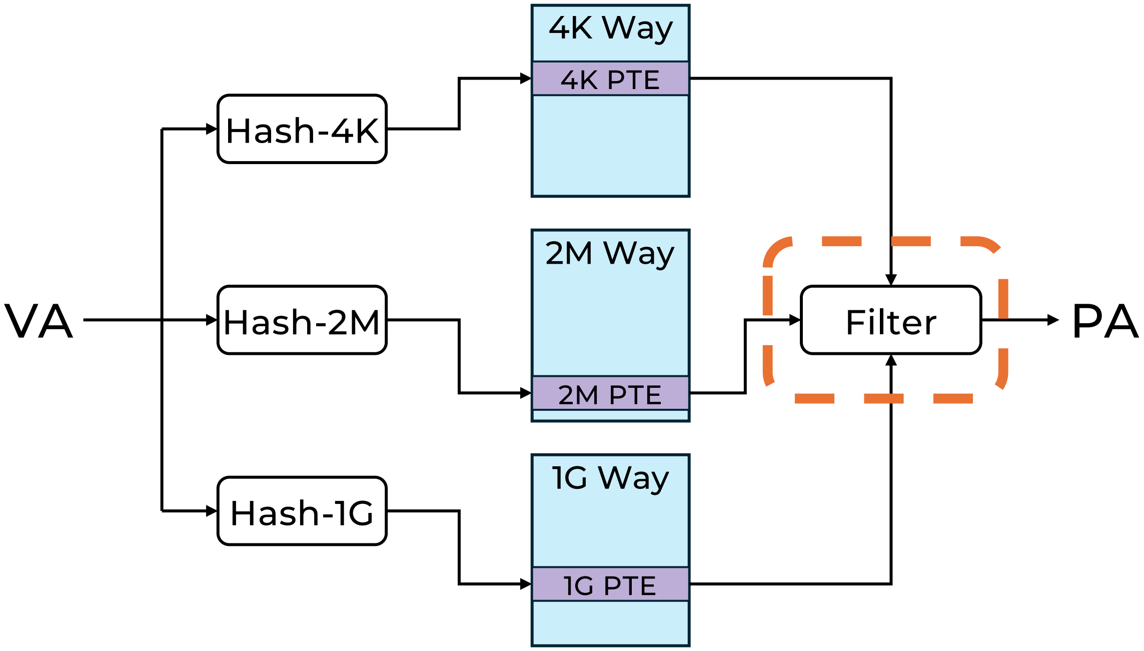


+ Index with hash function values
- Index with bits in virt. address

+ PTEs points to the actual page
- PTEs points to the next level PT

Elastic Cuckoo Page Table (ECPT) vs. Radix-Tree Page Table

ECPT: A different design from radix schemes



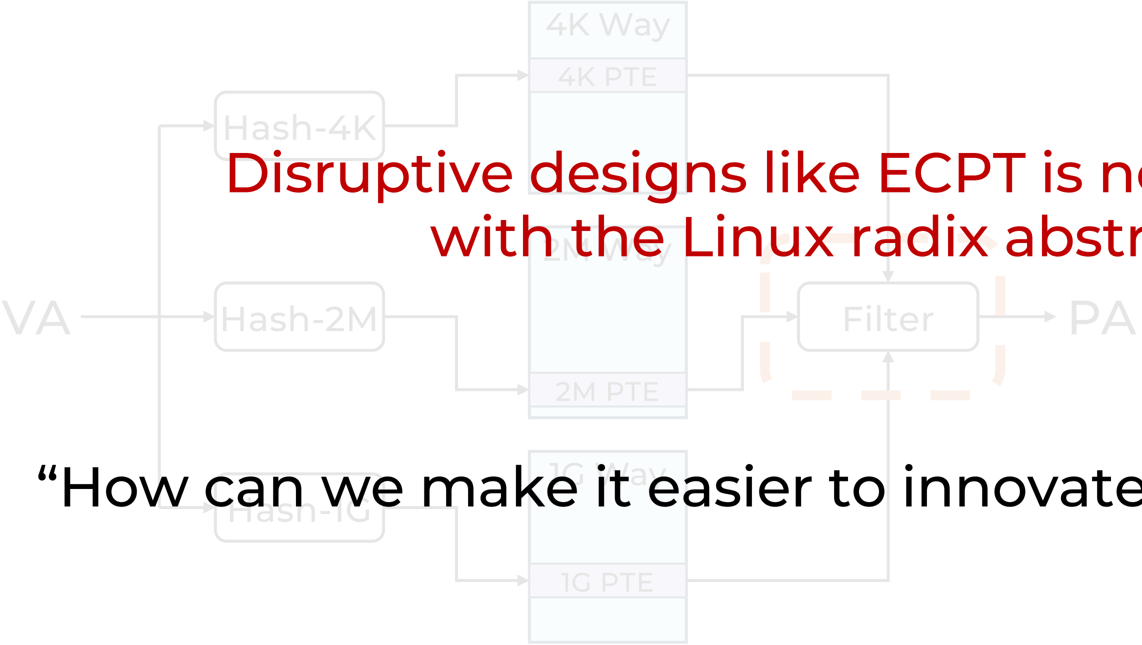
+ Index with hash function values
- Index with bits in virt. address

+ PTEs points to the actual page
- PTEs points to the next level PT

+ Page walk returns many PTEs
- Page walk returns one final PTE

Elastic Cuckoo Page Table (ECPT) vs. Radix-Tree Page Table

ECPT: A different design from radix schemes



Disruptive designs like ECPT is not compatible with the Linux radix abstraction

“How can we make it easier to innovate in virtual memory?”

+ Index with hash function values
- Index with bits in virt. address

+ PTEs points to the actual page
- PTEs points to the next level PT

+ Page walk returns many PTEs
- Page walk returns one final PTE

Elastic Cuckoo Page Table (ECPT) vs. Radix-Tree Page Table

Contributions

ARTIFACT

ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE

ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL

ARTIFACT EVALUATED usenix ASSOCIATION REPRODUCED

EMT: an OS framework for new memory translation architectures

Hardware neutral design with no assumption on page table structures

Extensible interface that enables hardware-specific optimizations

Accurate profiling with near-zero (<0.2%) performance overhead

Contributions

ARTIFACT

ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE

ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL

ARTIFACT EVALUATED usenix ASSOCIATION REPRODUCED

EMT: an OS framework for new memory translation architectures

- Hardware neutral design with no assumption on page table structures
- Extensible interface that enables hardware-specific optimizations
- Accurate profiling with near-zero (<0.2%) performance overhead

An open platform for memory translation research

- Research ready for full system prototyping, development, and evaluation
- Open source available at https://github.com/xlab-uiuc/emt

Contributions

ARTIFACT

ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE

ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL

ARTIFACT EVALUATED usenix ASSOCIATION REPRODUCED

EMT: an OS framework for new memory translation architectures

- Hardware neutral design with no assumption on page table structures
- Extensible interface that enables hardware-specific optimizations
- Accurate profiling with near-zero (<0.2%) performance overhead

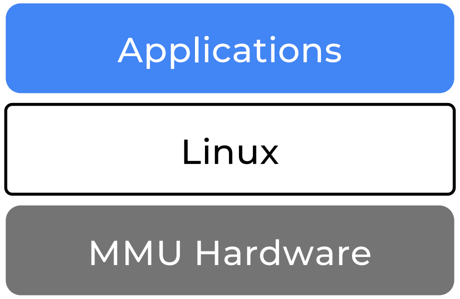
An open platform for memory translation research

- Research ready for full system prototyping, development, and evaluation
- Open source available at https://github.com/xlab-uiuc/emt

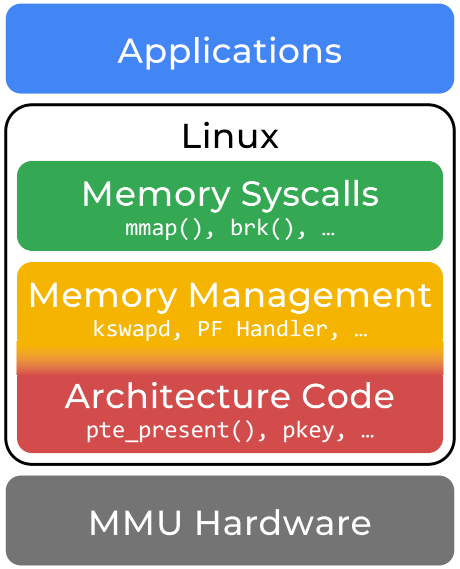
New insights on hashing-based designs from the OS perspective

- New challenges previously undiscovered regarding their OS implications
- New solutions to these challenges evaluated in our ECPT implementation

EMT Overview

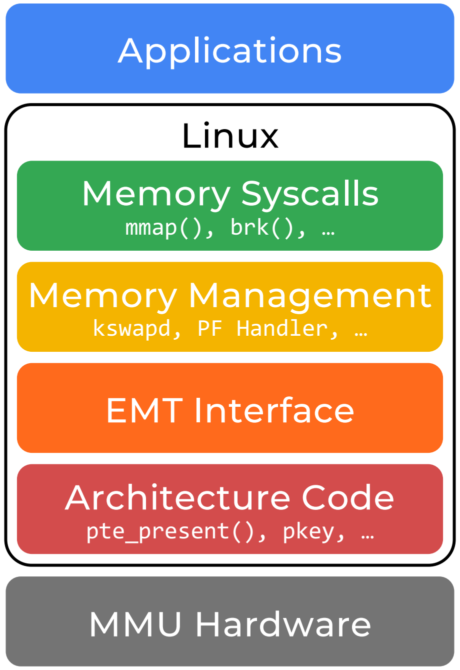


EMT Overview



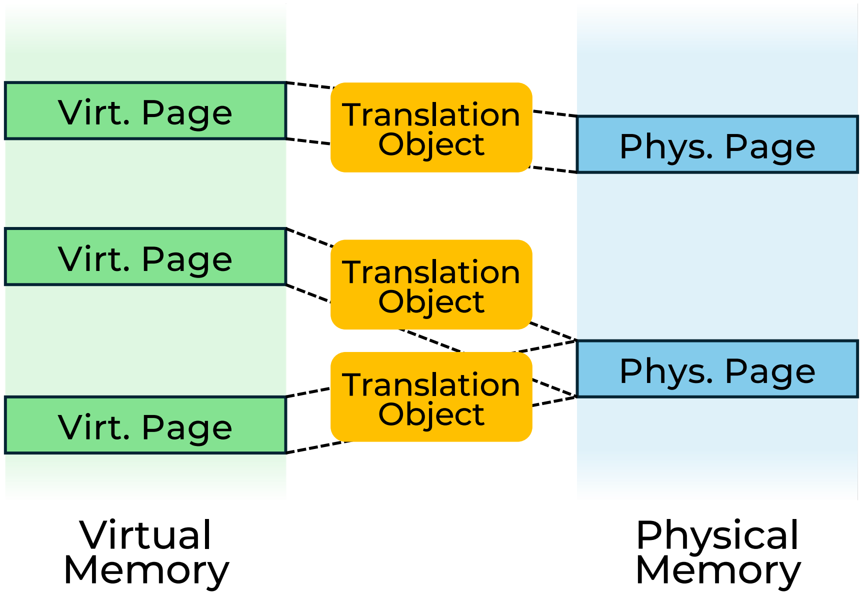
Linux coupled memory management and arch-specific code

EMT Overview



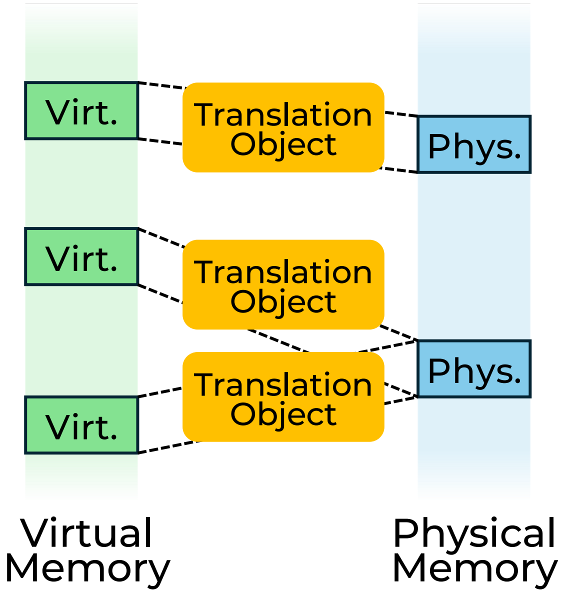
EMT decoupled memory management and arch-specific code

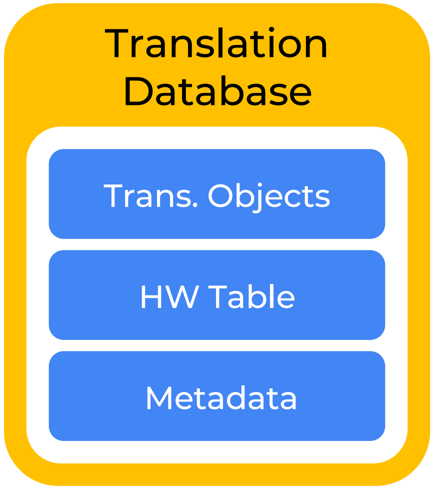
EMT models functionality, not structure



Translation Object
Models a *page mapping*

EMT models functionality, not structure

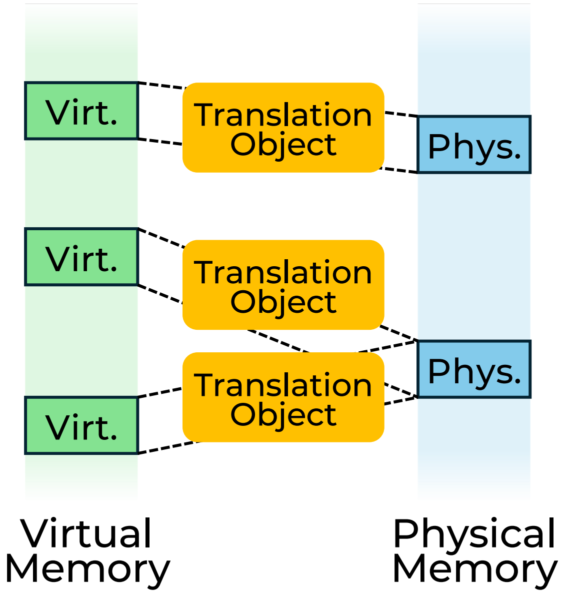




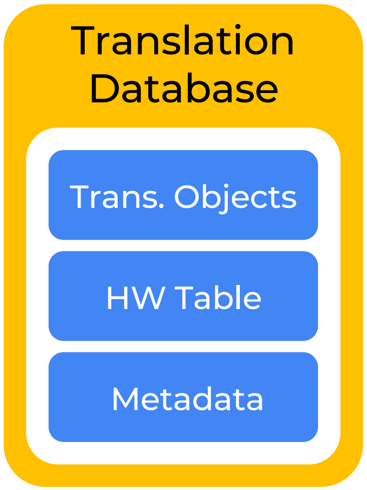
Translation Object
Models a *page mapping*

Translation Database
Models an *address space*

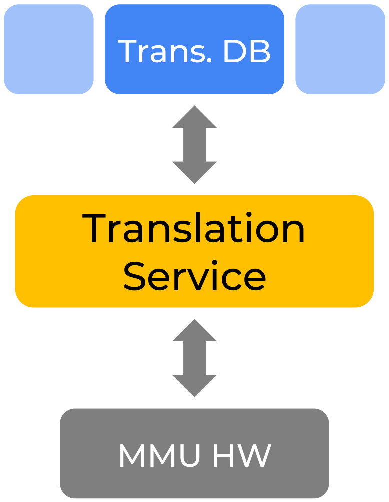
EMT models functionality, not structure



Translation Object
Models a *page mapping*

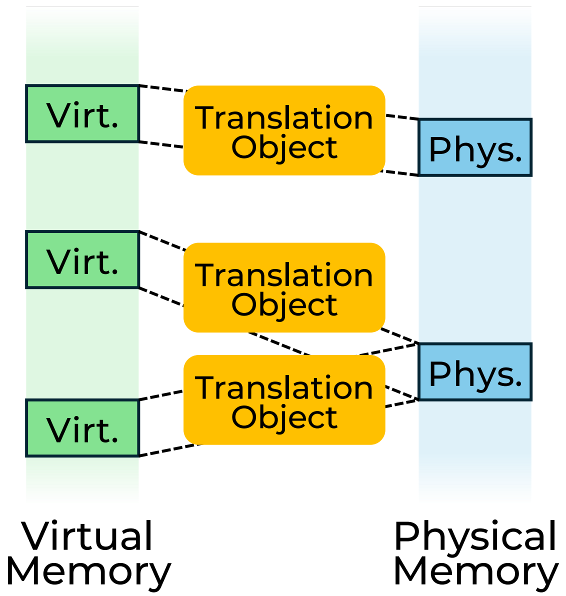


Translation Database
Models an *address space*

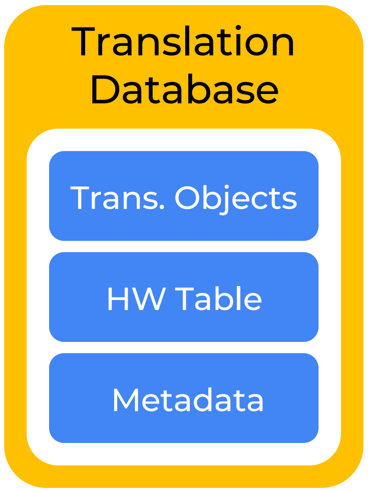


Translation Service
Models the *MMU*

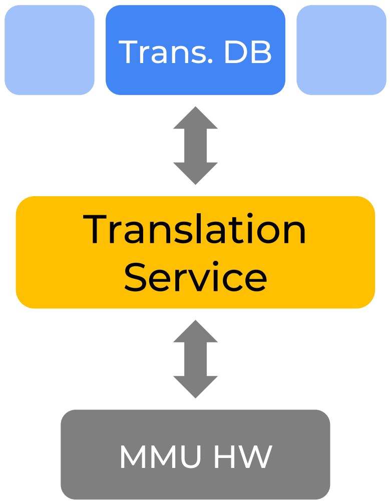
EMT models functionality, not structure



Translation Object
Models a *page mapping*

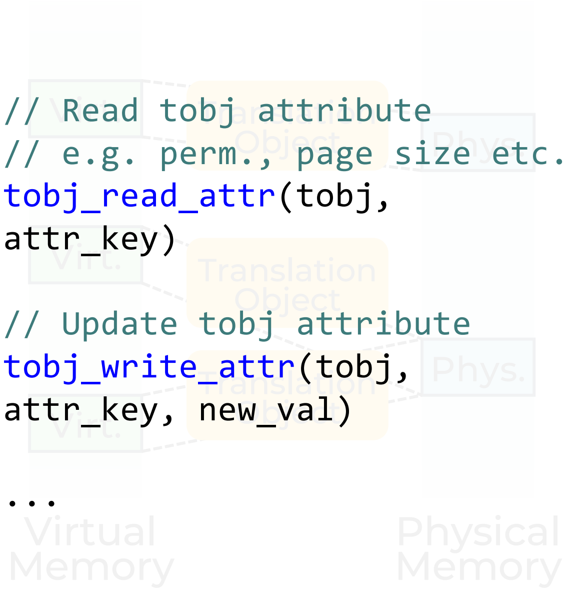


Translation Database
Models an *address space*



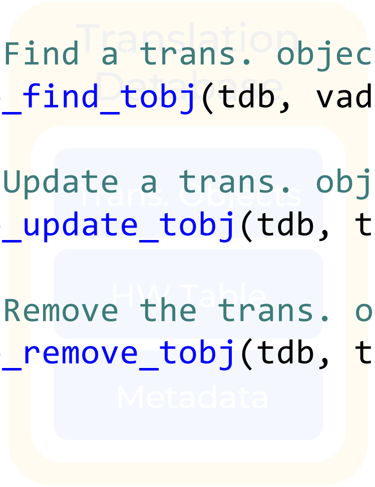
Translation Service
Models the *MMU*

EMT Basic Functions



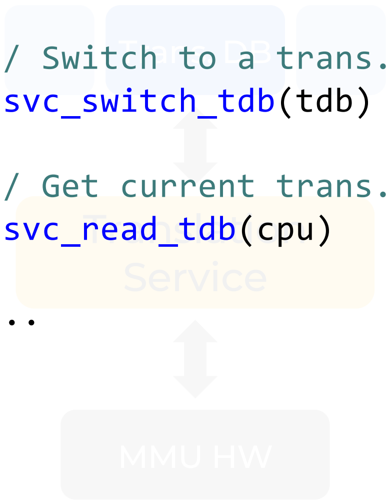
```
// Read tobj attribute
// e.g. perm., page size etc.
tobj_read_attr(tobj,
attr_key)

// Update tobj attribute
tobj_write_attr(tobj,
attr_key, new_val)

...
```

Translation Object
Models a *page mapping*



```
// Find a trans. object
tdb_find_tobj(tdb, vaddr)

// Update a trans. object
tdb_update_tobj(tdb, tobj)

// Remove the trans. object
tdb_remove_tobj(tdb, tobj)

...
```

Translation Database
Models an *address space*

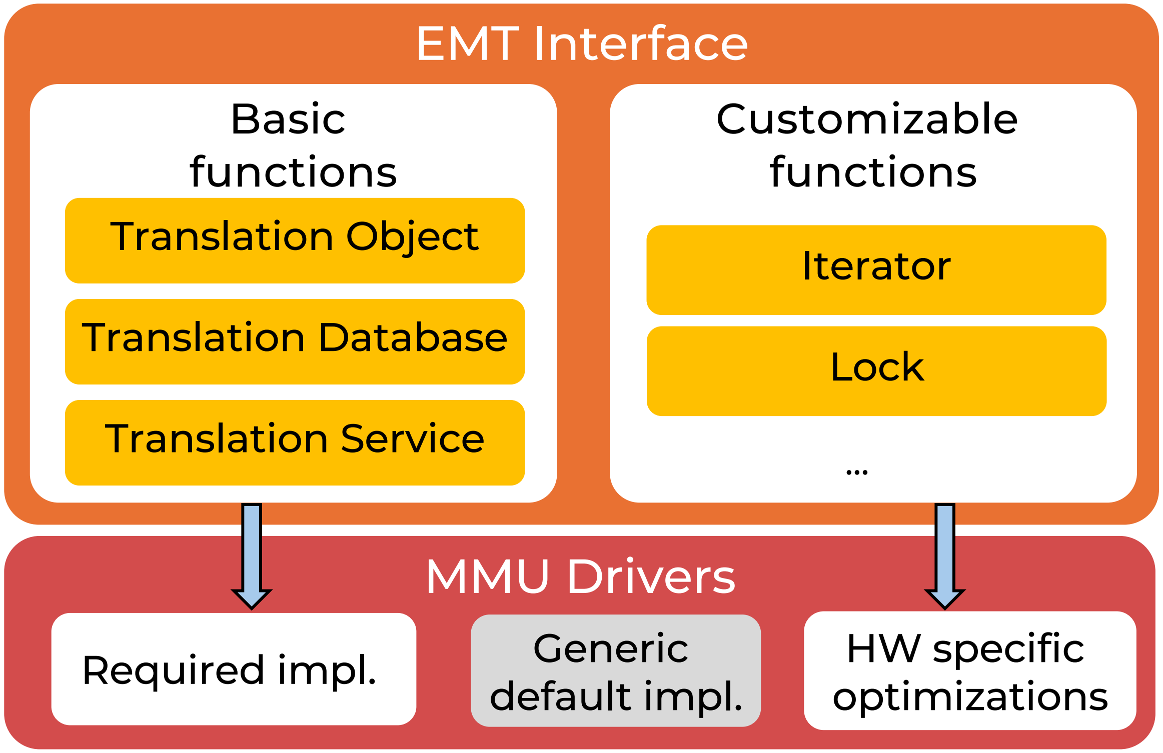


```
// Switch to a trans. db
tsvc_switch_tdb(tdb)

// Get current trans. db
tsvc_read_tdb(cpu)

...
```

Translation Service
Models the *MMU*

EMT Customizable Functions

HW neutral



Exploit HW features ***and*** guarantee HW neutrality

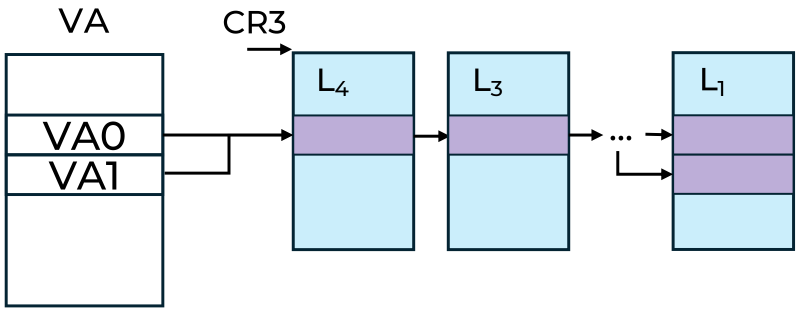
EMT enables HW-specific optimizations

Customizable functions: iterator

Iterate over a range of virtual address

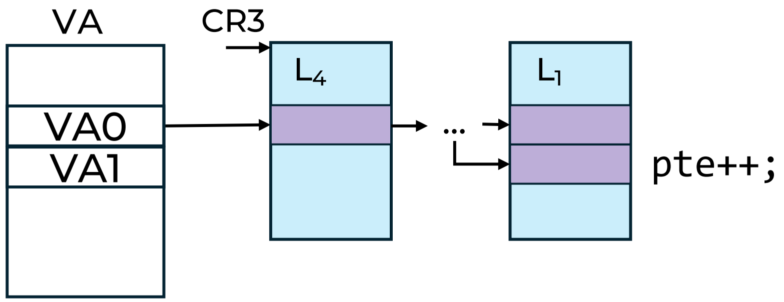
`tobj_iter_next` gets the next trans. object

Default implementation
HW neutral but less performant



Full page table walk for every VA

Radix MMU driver
Customized to exploit locality



EMT enables HW-specific optimizations

Customizable functions: iterator

Iterate over a range of virtual address

`tobj_iter_next` gets the next trans. object

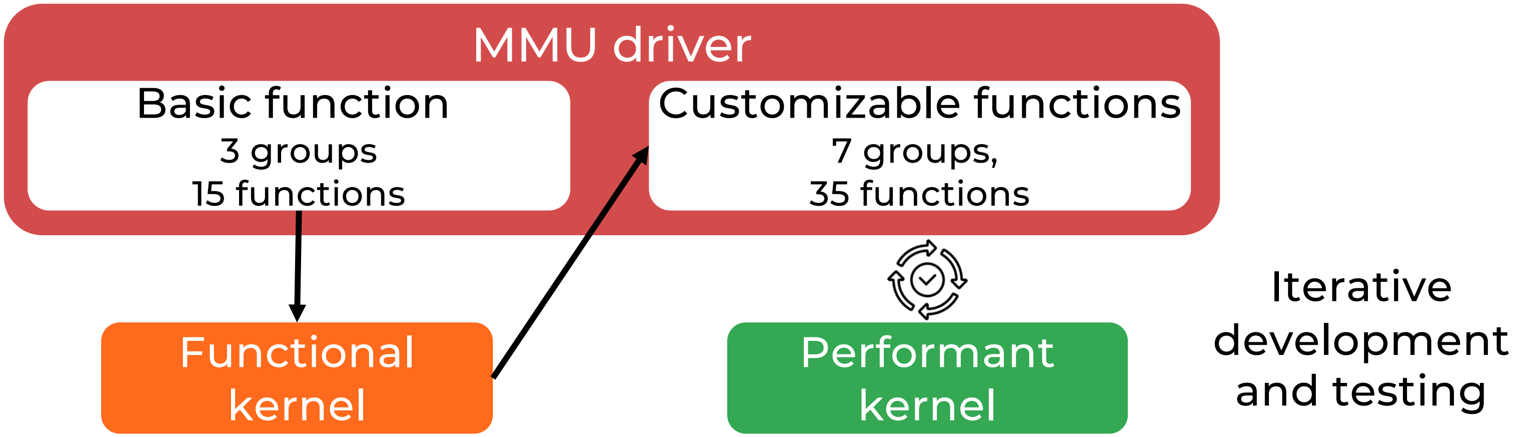
Default implementation
HW neutral but less performant

```
tdb_find_tobj(iter->tdb, iter->va,
    tobj); /* full page walk on Radix */
tobj_read_attr(tobj, TOBJ_ATTR_SIZE,
   &size);
iter->va += size
...
```

Radix MMU driver
Customized to exploit locality

```
... /* update tobj */
if ((iter->va + PAGE_SIZE) &
   (~PMD_MASK)) {
   iter->va += PAGE_SIZE;
   iter->pte++;
   return 0;
} /* handle other cases */
```

EMT simplifies OS support for different MMUs



MMU driver

Basic function
3 groups
15 functions

Customizable functions
7 groups,
35 functions

Functional kernel

Performant kernel

Iterative development and testing

EMT supports tree- and hash-based translations (e.g., Radix and ECPT)

Flattened page table support implemented with < 700 LOC

- No changes to Linux memory management routines
- Reuse part of the x86 MMU driver

EMT has negligible performance overhead

EMT-Linux on the Radix MMU driver vs. vanilla Linux

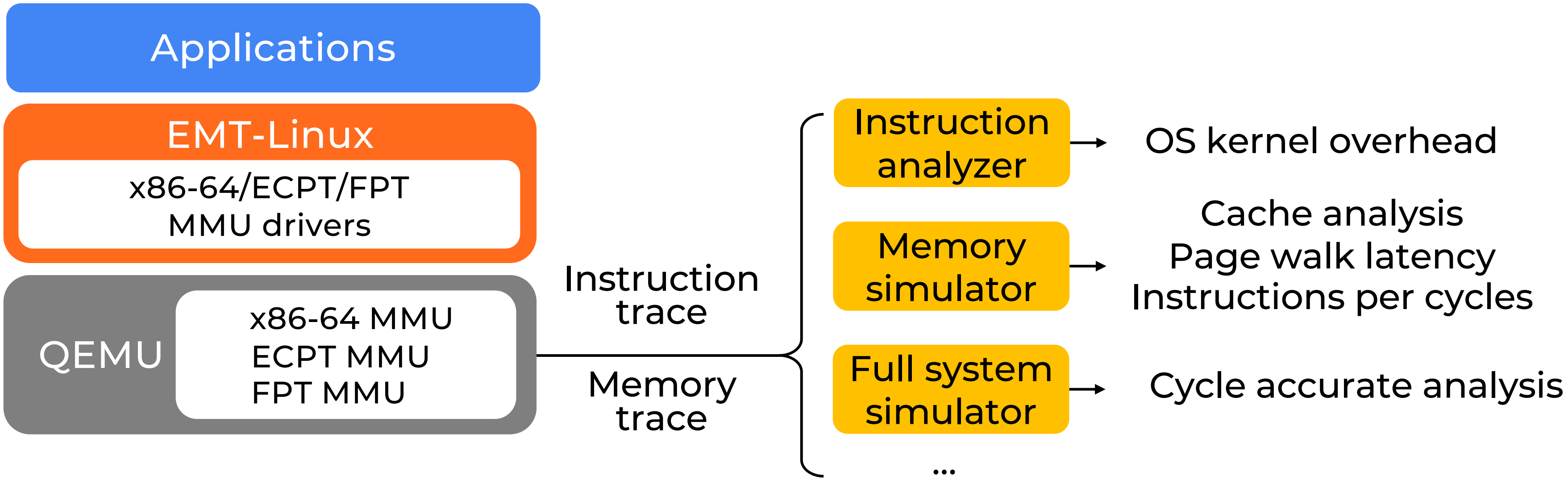
Benchmarks

EMT is carefully engineered to minimize performance overhead

Minimize call stacks depth and keep a similar cache efficiency

EMT enables all HW-specific optimizations for radix

An open platform for virtual memory research



EMT enables end-to-end system evaluations in the absence of hardware

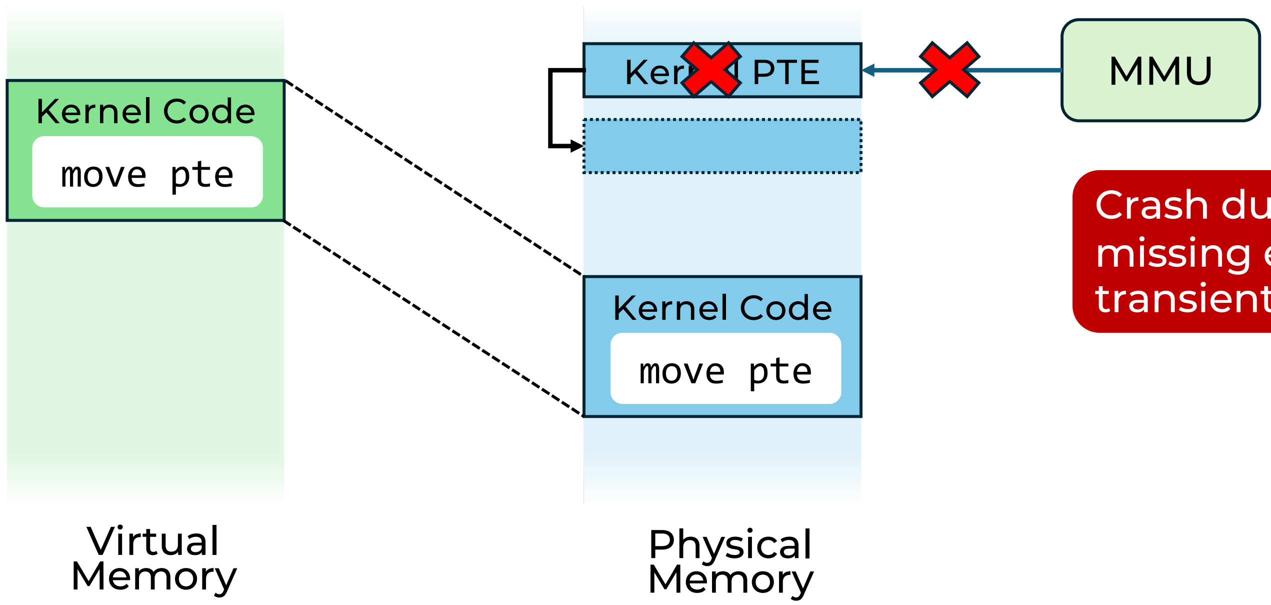
EMT supports rich performance analysis

EMT brings insights from the OS perspective

Hash page table: self-reference paradox

Approach 1: invalidation before copy

Kernel PTEs may move due to hash collisions

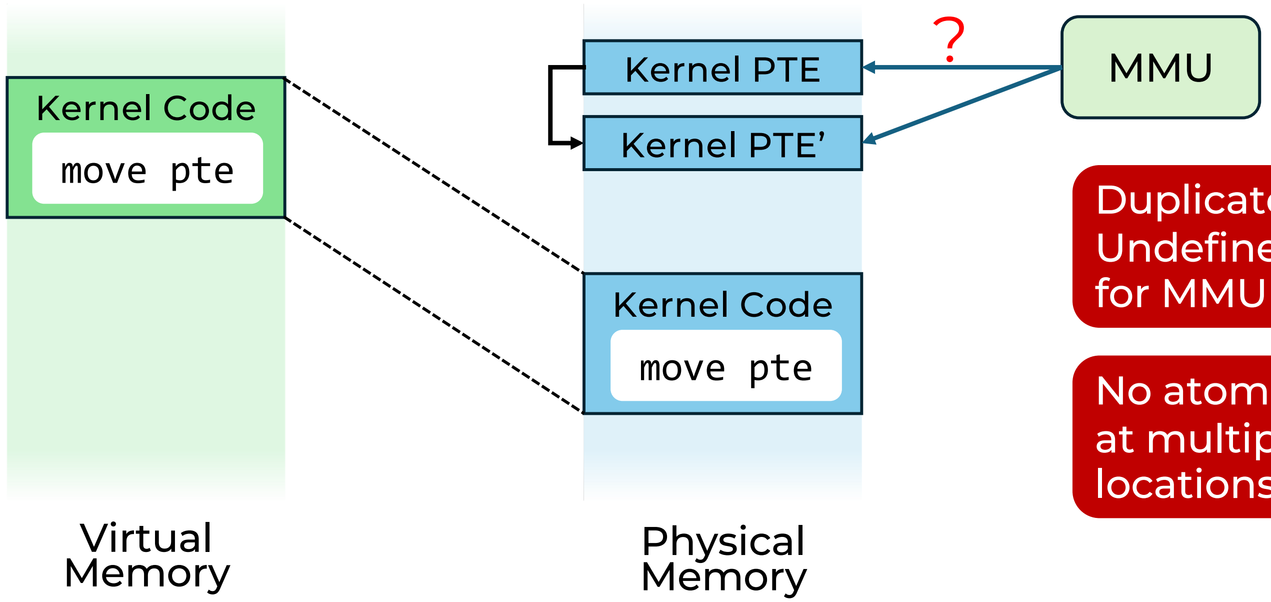


Crash due to missing entries in transient window

EMT brings insights from the OS perspective

Hash page table: self-reference paradox

Approach 2: copy before invalidation



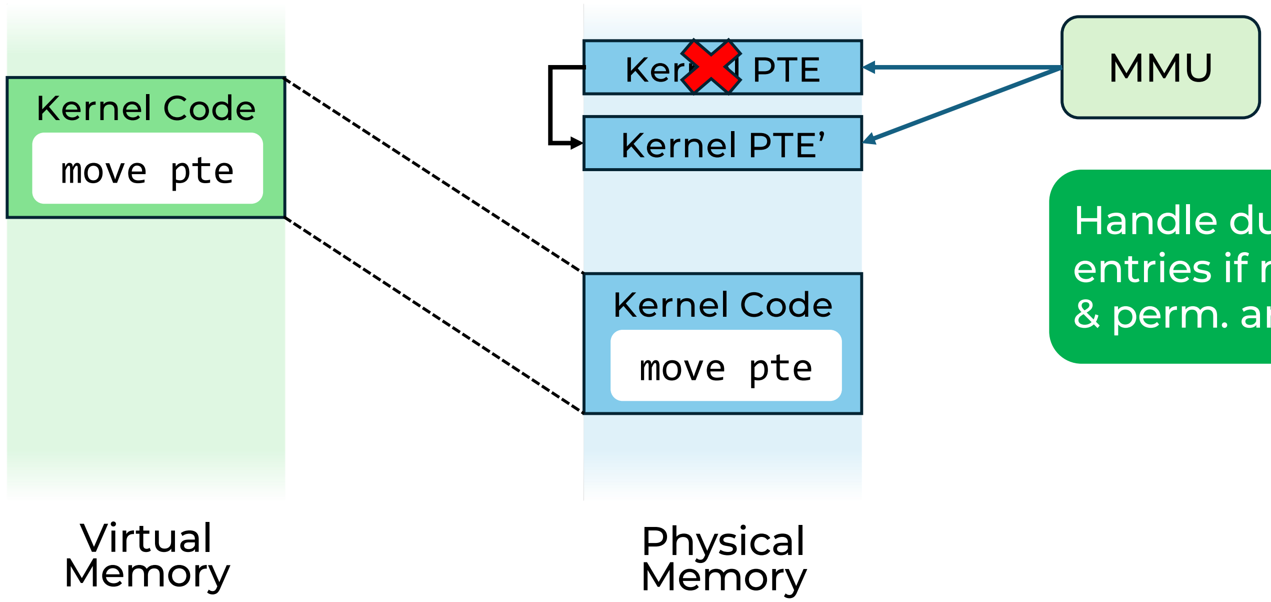
Duplicated entries
Undefined behavior for MMU

No atomic updates at multiple locations

EMT brings insights from the OS perspective

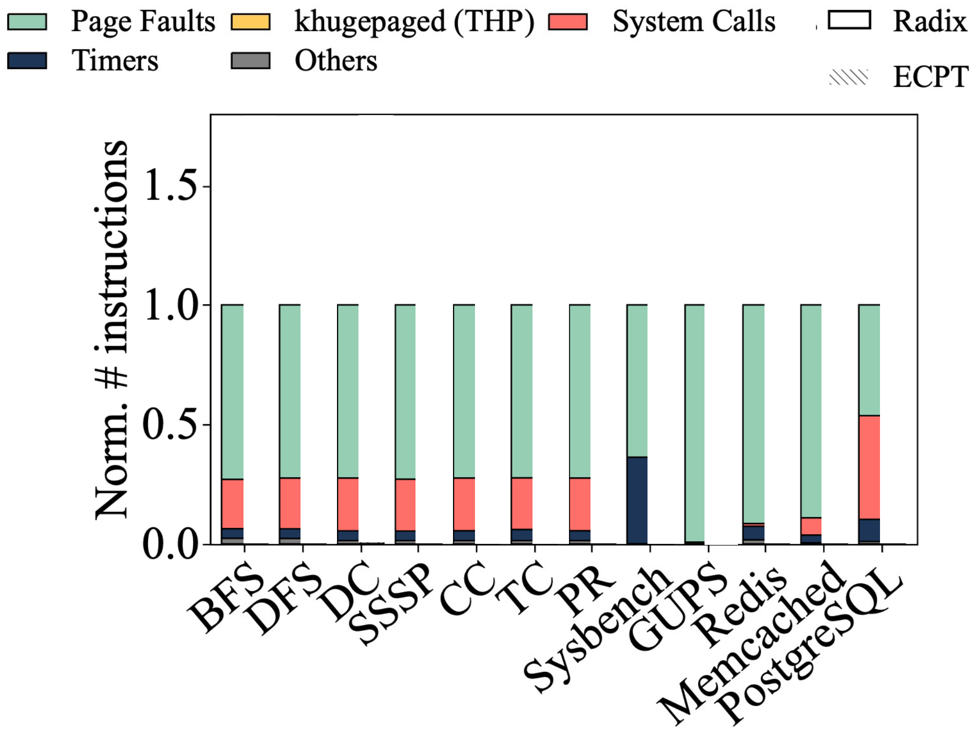
Hash page table: self-reference paradox

Solution: copying before invalidation + extend MMU logic



Handle duplicated entries if mapping & perm. are same

EMT helps analyze MMU design tradeoffs



ECPT is faster than x86 Radix on hardware metrics

ECPT incurs 1.74x page fault handling overhead over Radix

Conclusion

ARTIFACT

ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE

ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL

ARTIFACT EVALUATED usenix ASSOCIATION REPRODUCED

OS support is essential for memory translation designs

- Understanding OS implications is very beneficial
- Experimenting with modern Oses is strongly encouraged
- OS extensibility is crucial to foster diverse memory translation research

EMT: an OS framework for new memory translation architectures

- Hardware neutral design with no assumption on page table structures
- Extensible interface that enables hardware-specific optimizations
- Accurate profiling with near-zero (<0.2%) performance impacts

An open platform for memory translation research

- Research ready for full system prototyping, development, and evaluation
- Open source available at https://github.com/xlab-uiuc/emt