Elastic Translations:
Fast virtual memory with multiple translation sizes

Stratos Psomadakis; Chloe Alverti; Vasileios Karakostas; Christos Katsakioris
Dimitrios Siakavaras; Konstantinos Nikas; Georgios Goumas; Nectarios Koziris

National Technical University of Athens
CSLab

AERO
Funded by the European Union

Artifacts Evaluated acm Reusable V1.1
Artifacts Available acm V1.1

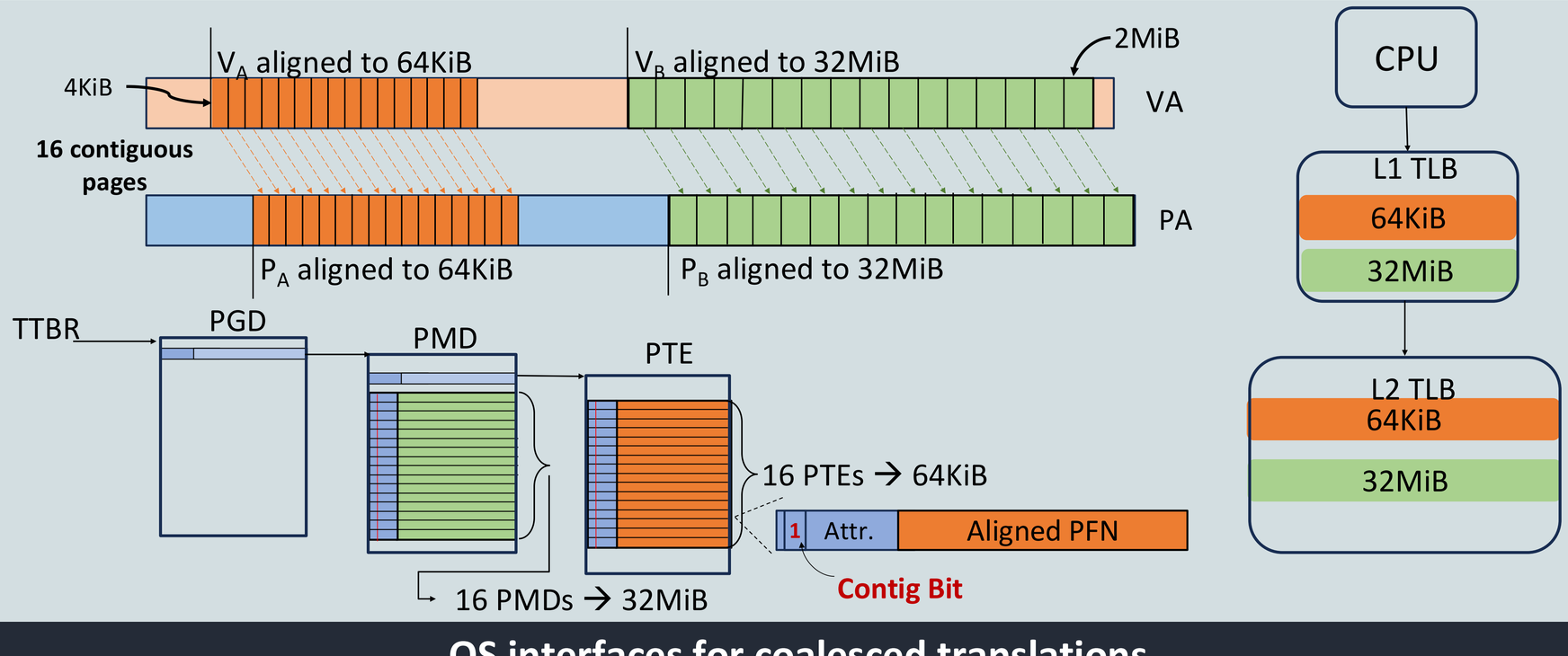
AT Wall: TLBs under pressure

Large pages extend TLB reach
2MiB → *overfit / underfit*
1GiB → *too large*

ARMv8: 64KiB and ***32MiB*** coalesced translations but ***OS support is limited***

Our proposal: ***Elastic Translations***
→***Transparent*** *OS support for coalesced translations*
→***CoalaPaging*** to generate practical contiguity
→***Leshy*** for informed translation size selection via lightweight HW-assisted sampling

ARMv8-A OS-assisted TLB coalescing



OS interfaces for coalesced translations

Interface	64KiB	32MiB	Virtualized Execution	Transparent	Size Selection	Allocation Policy	Asynchronous Promotions
Linux HugeTLB	✓	✓	✗	✗	User-defined	Pre-allocate	✗
Linux (m)THP	✓	✗	✗	✓	2MiB→ 64KiB→ 4KiB	First-touch	✗
Elastic Translations	✓	✓	✓	✓	Guided	Opportunistic	✓

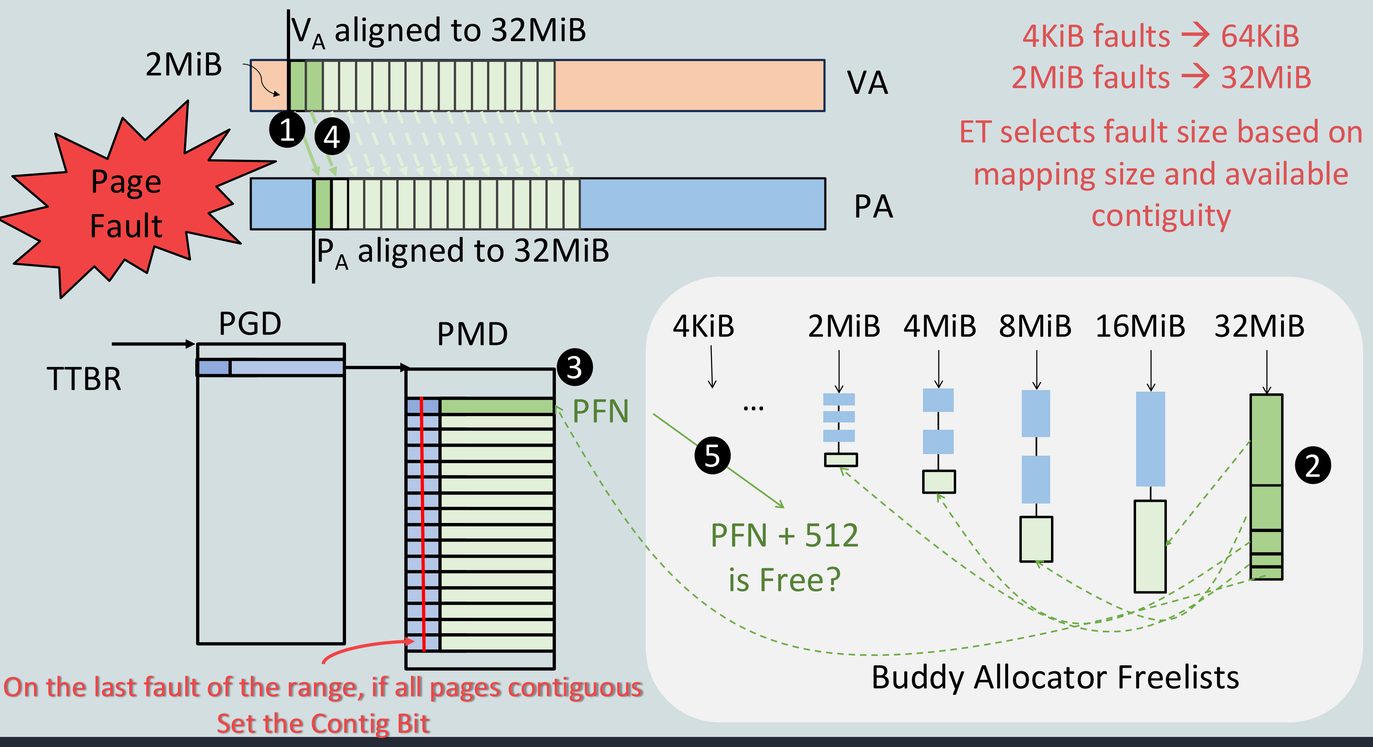
Elastic Translations (ET)

Opportunistic
Opportunistically create ***64KiB*** **and** ***32MiB*** contiguity with ***CoalaPaging*** and ***CoalaKhugepaged***

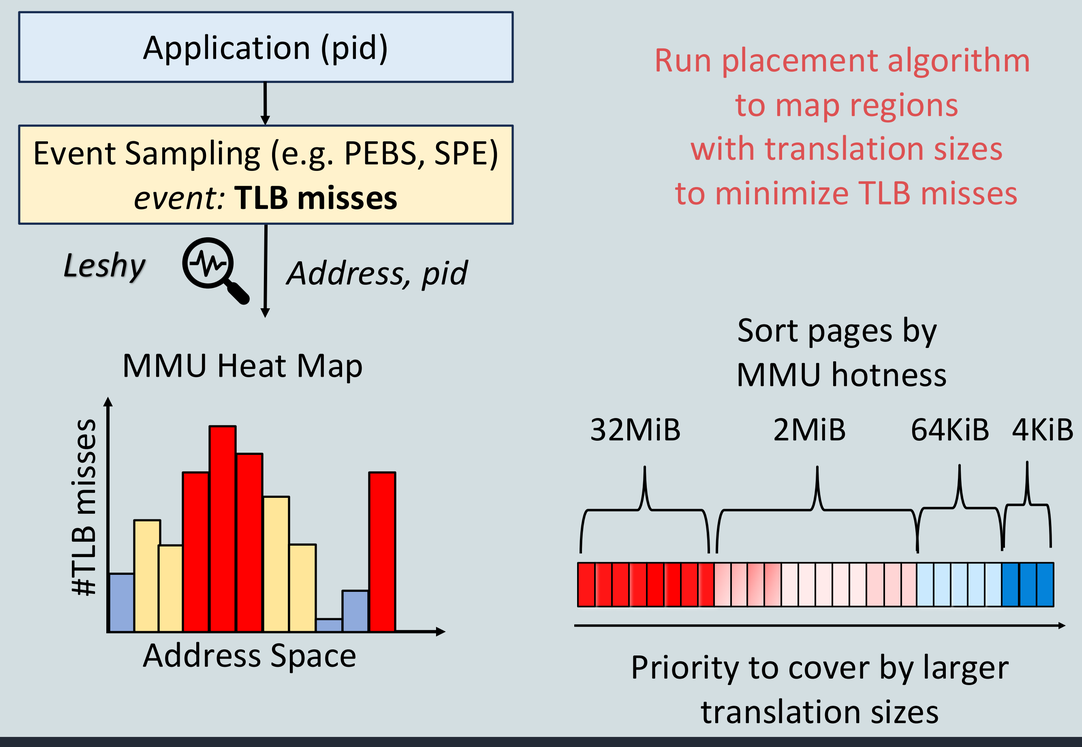
Transparent
Transparently manage the ***contig bit*** at fault time or during asynchronous promotions

Guided
Informed translation size selection with the ***Leshy*** HW-assisted profiler

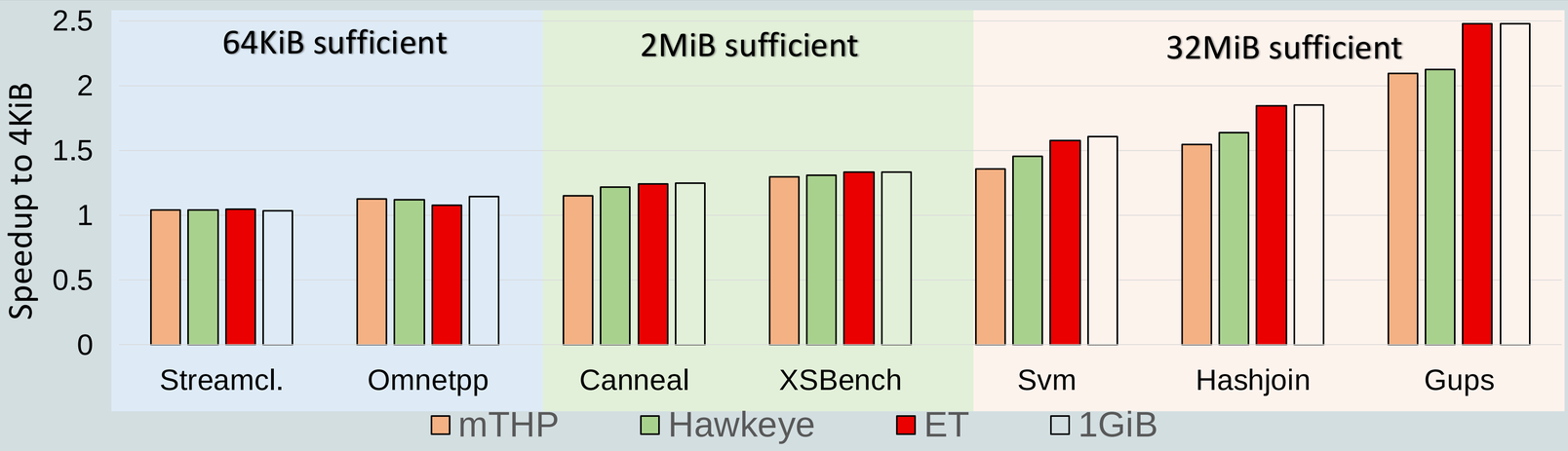
ET uses ***CoalaPaging*** (Coalescing-Aware Paging) to generate practical and aligned contiguity on demand (across faults) without reserving blocks of memory or requiring additional metadata. It harnesses this contiguity by ***transparently setting the contig bit***.



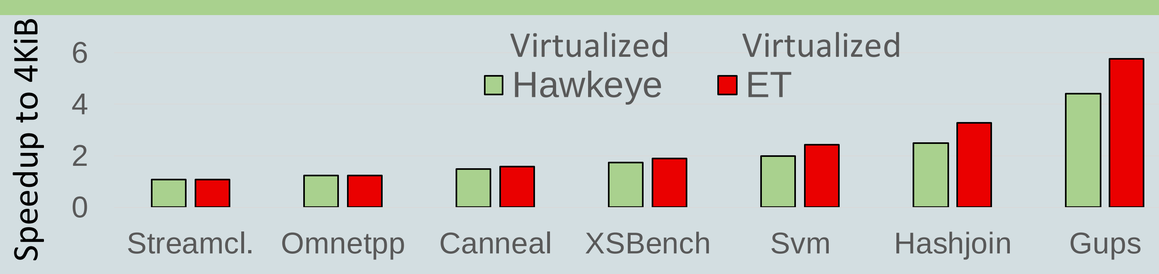
ET supports asynchronous promotions with ***CoalaKhugepaged***.
It uses the ***Leshy*** profiler for informed translation size selection (4KiB / 64KiB / 2MiB / 32MiB) based on real MMU activity



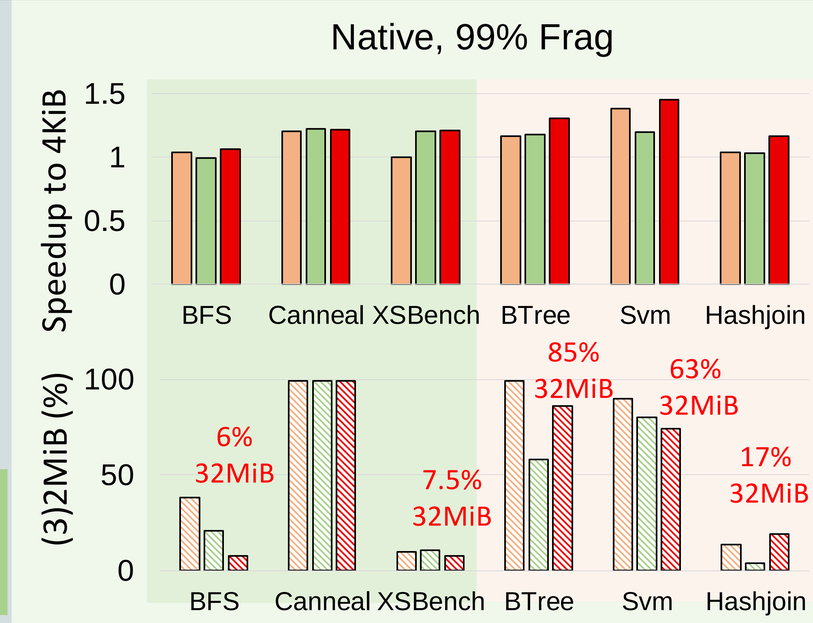
ET Real System Evaluation – Implemented in Linux v5.18 – Evaluated on an ARMv8 Ampere Altra Server



One size does not fit all! ET-generated 64KiB and 32MiB translations provide significant performance gains, slotting between 4KiB/2MiB/1GiB pages and addressing limitations of 2MiB/1GiB large page model



Nested Page Walks make TLB misses costlier, pronouncing the benefits of 32MiB translations and boosting performance up to 30% over Hawkeye



Leshy guidance allows ET to optimally use 4KiB/2MiB/32MiB translations under fragmentation and boosts performance by up to 22% while reducing 2MiB usage