Hybrid TLB Coalescing: Improving TLB Translation Coverage under Diverse Fragmented Memory Allocations

Chang Hyun Park, *Taekyung Heo, Jungi Jeong, and Jaehyuk Huh*

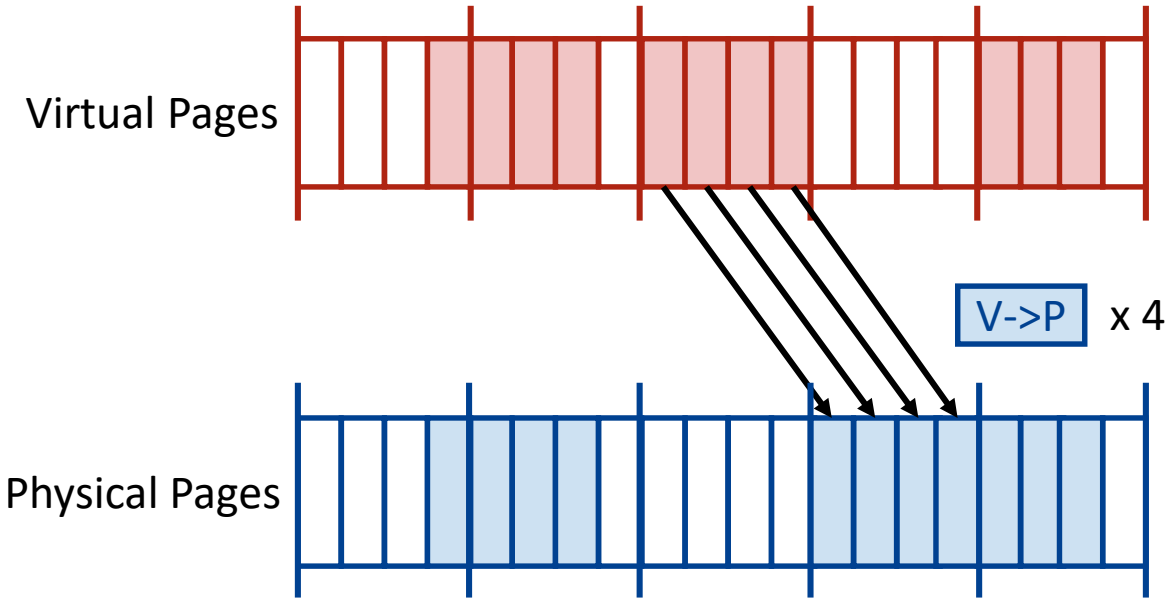
KAIST School of Computing

Introduction

- Virtual memory provides rich features
  - Requires an address translation
- Workloads have grown in size pressuring TLB
- Contiguous memory allocations to the rescue!

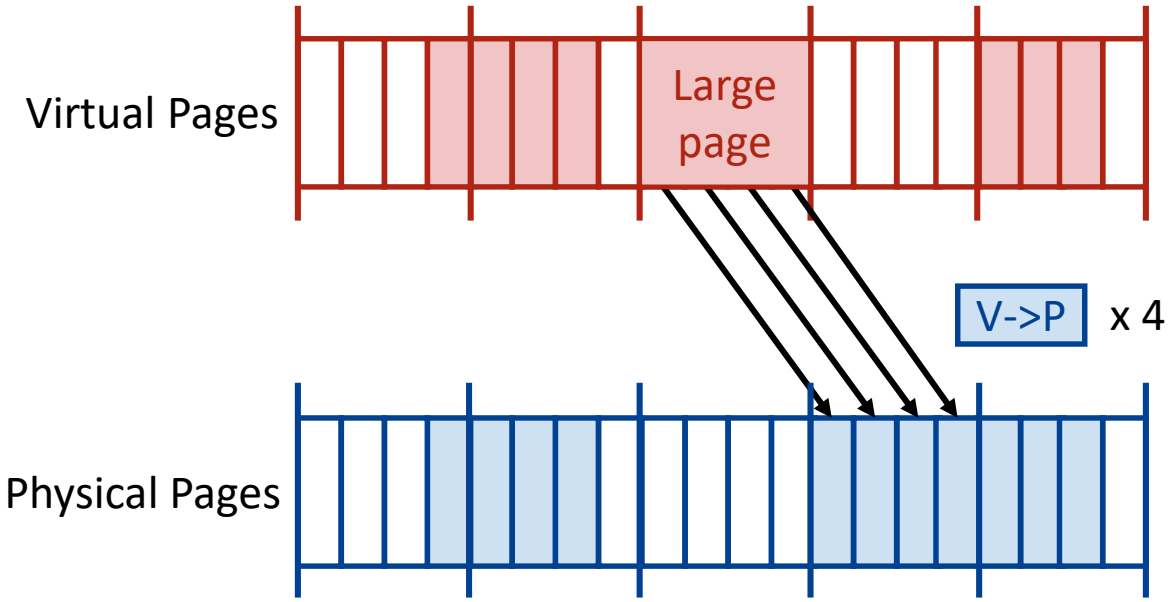
Past Proposals: Large pages

- Large pages represent larger mappings (2MB)
  - Strict alignment required
  - Exact size match required



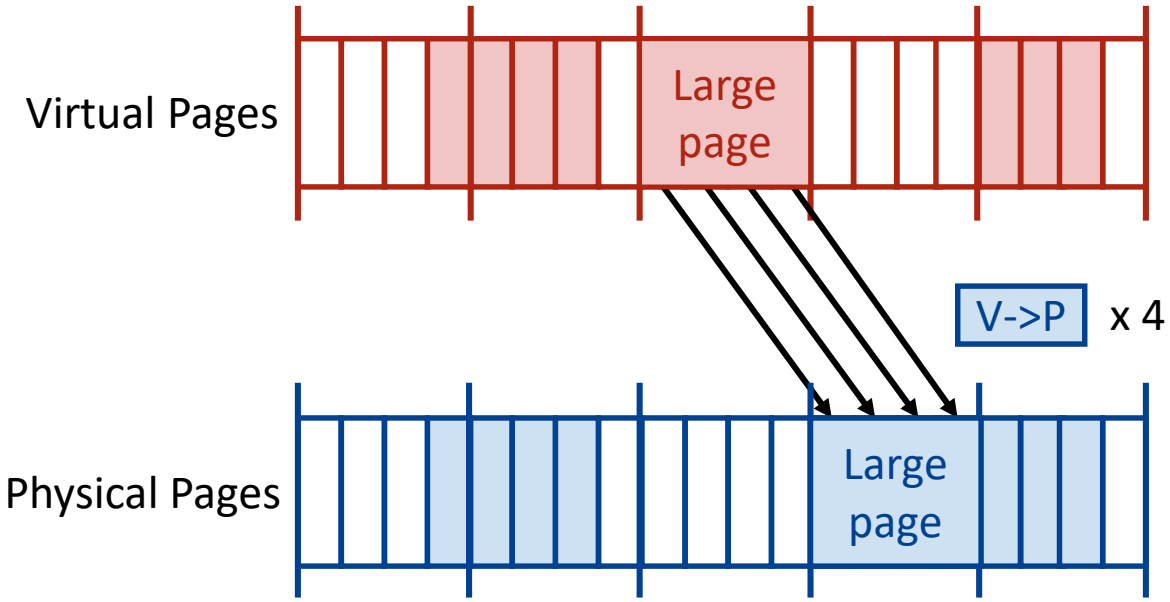
Past Proposals: Large pages

- Large pages represent larger mappings (2MB)
  - Strict alignment required
  - Exact size match required



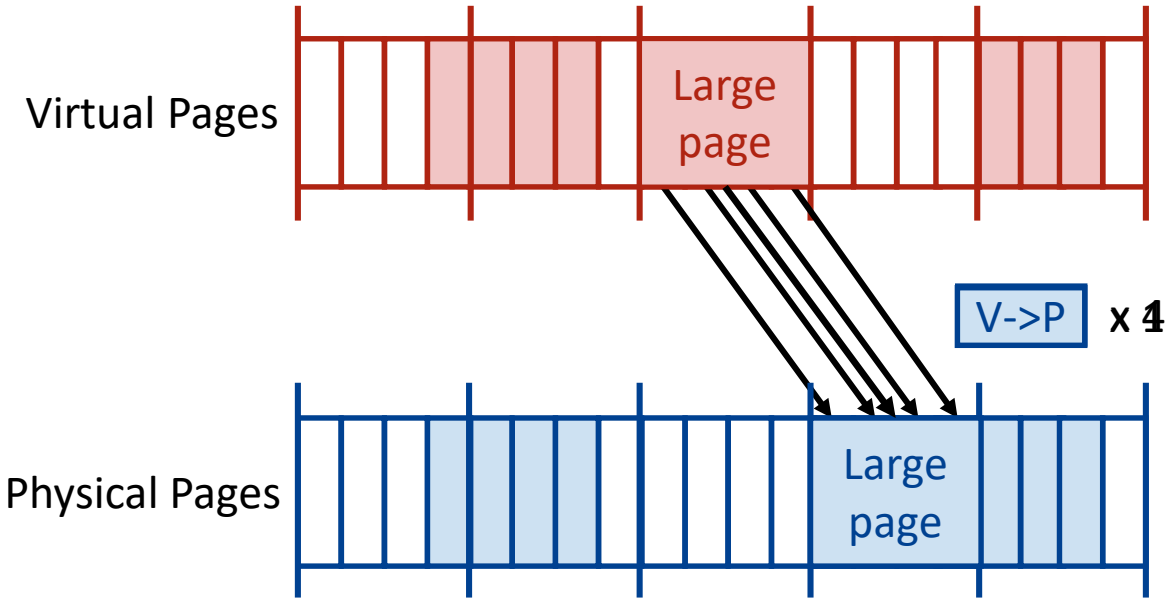
Past Proposals: Large pages

- Large pages represent larger mappings (2MB)
  - Strict alignment required
  - Exact size match required



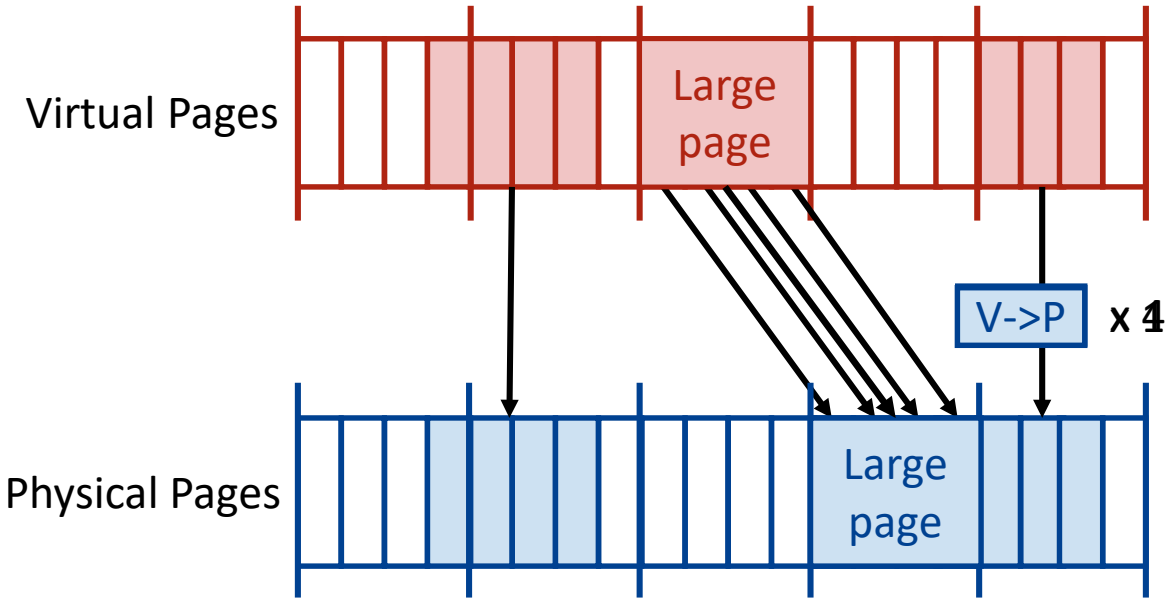
Past Proposals: Large pages

- Large pages represent larger mappings (2MB)
  - Strict alignment required
  - Exact size match required



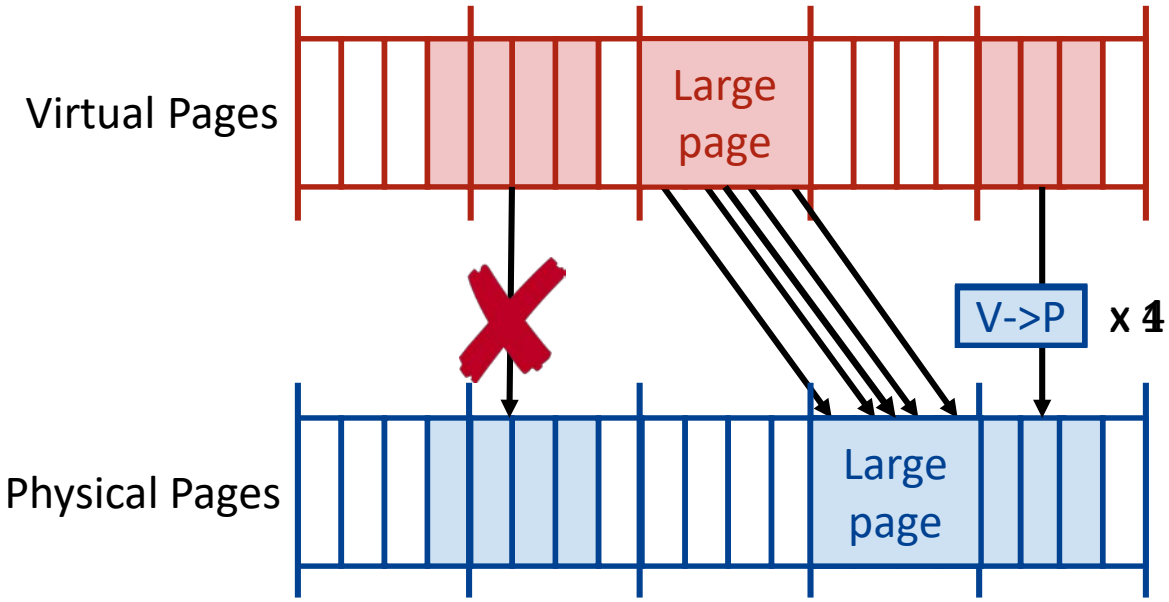
Past Proposals: Large pages

- Large pages represent larger mappings (2MB)
  - Strict alignment required
  - Exact size match required



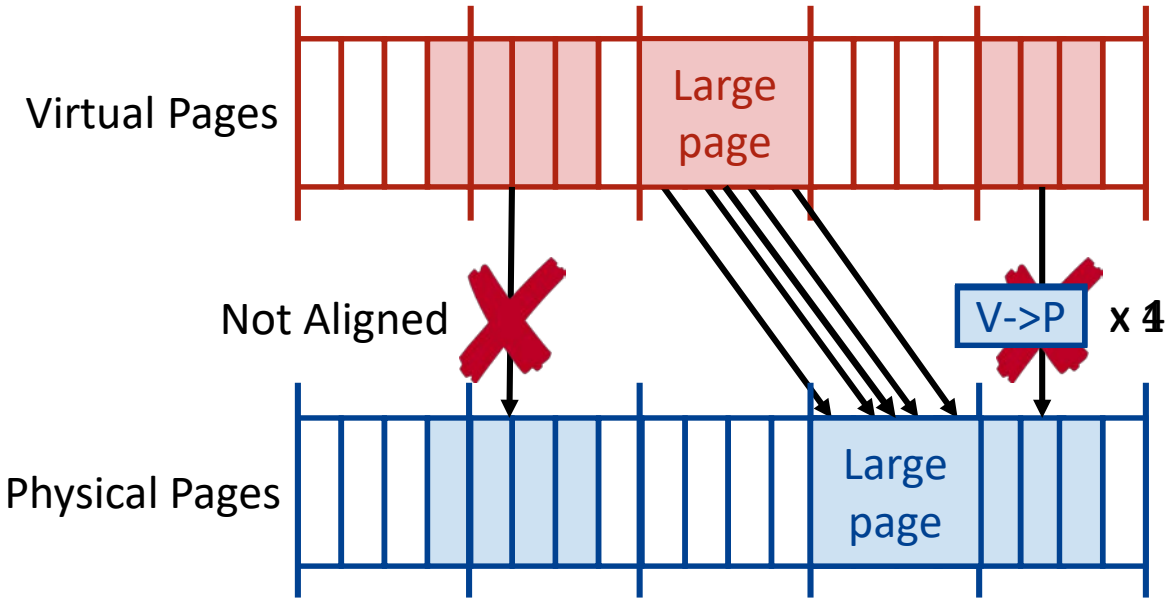
Past Proposals: Large pages

- Large pages represent larger mappings (2MB)
  - Strict alignment required
  - Exact size match required



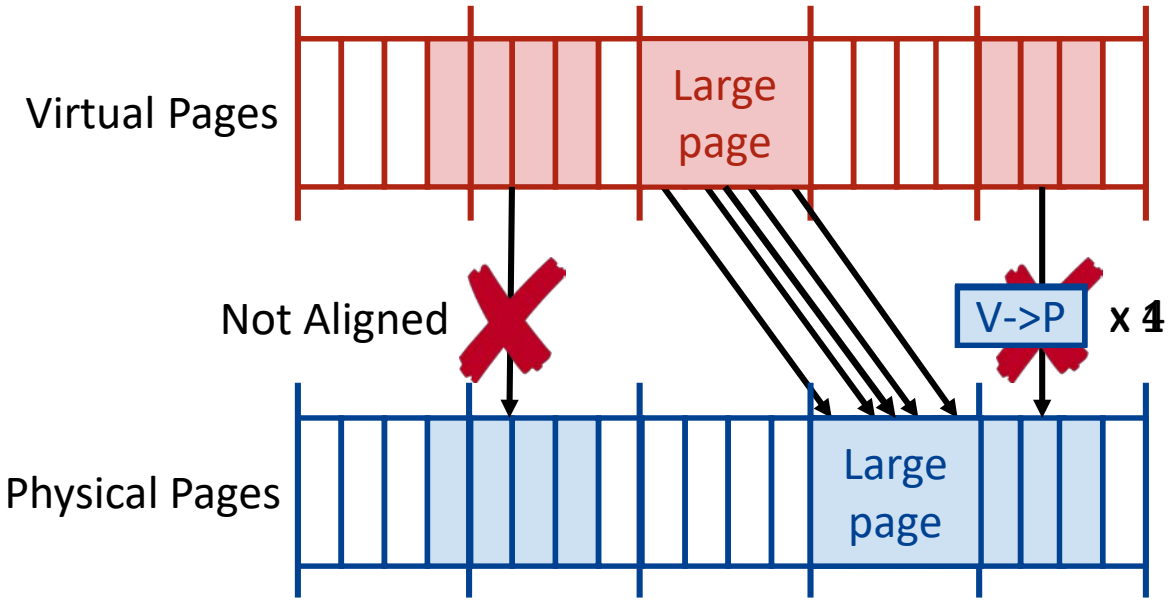
Past Proposals: Large pages

- Large pages represent larger mappings (2MB)
  - Strict alignment required
  - Exact size match required



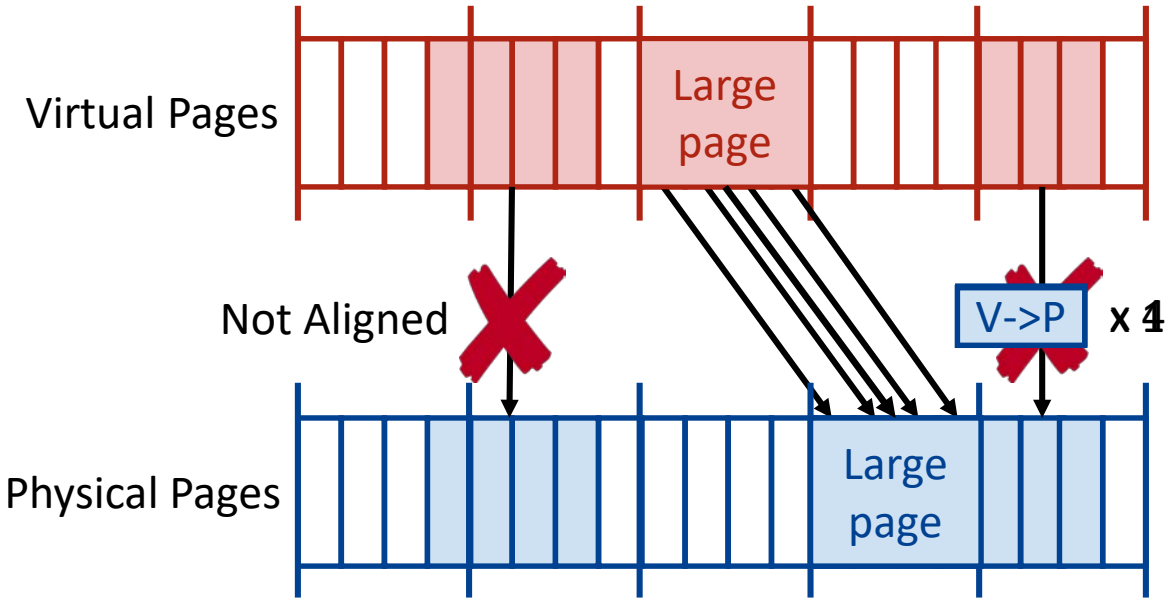
Past Proposals: Large pages

- Large pages represent larger mappings (2MB)
  - Strict alignment required
  - Exact size match required



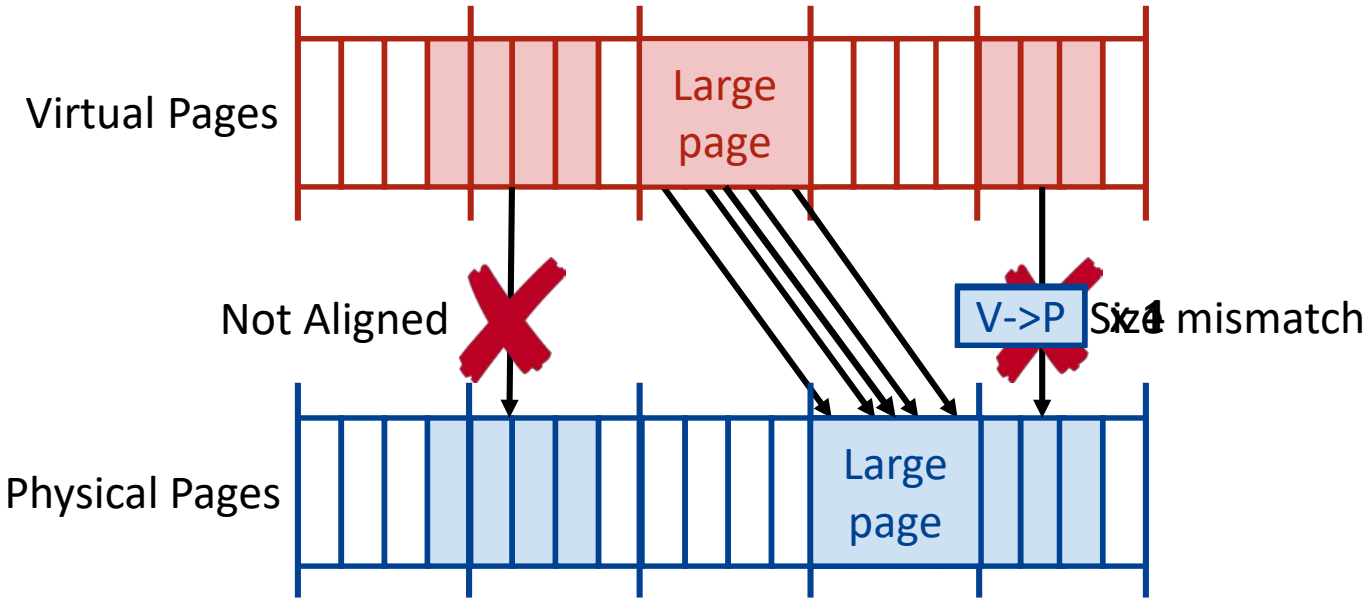
Past Proposals: Large pages

- Large pages represent larger mappings (2MB)
  - Strict alignment required
  - Exact size match required



Past Proposals: Large pages

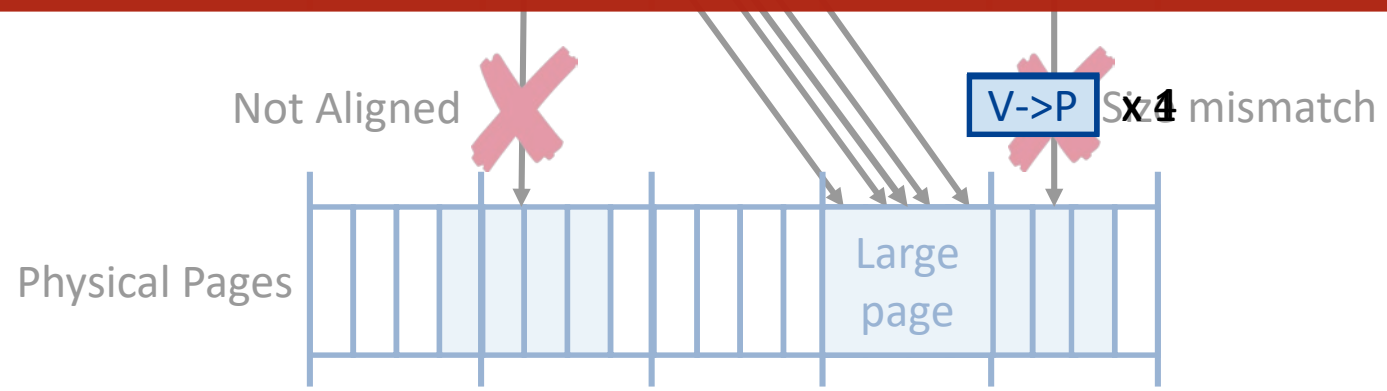
- Large pages represent larger mappings (2MB)
  - Strict alignment required
  - Exact size match required



Past Proposals: Large pages

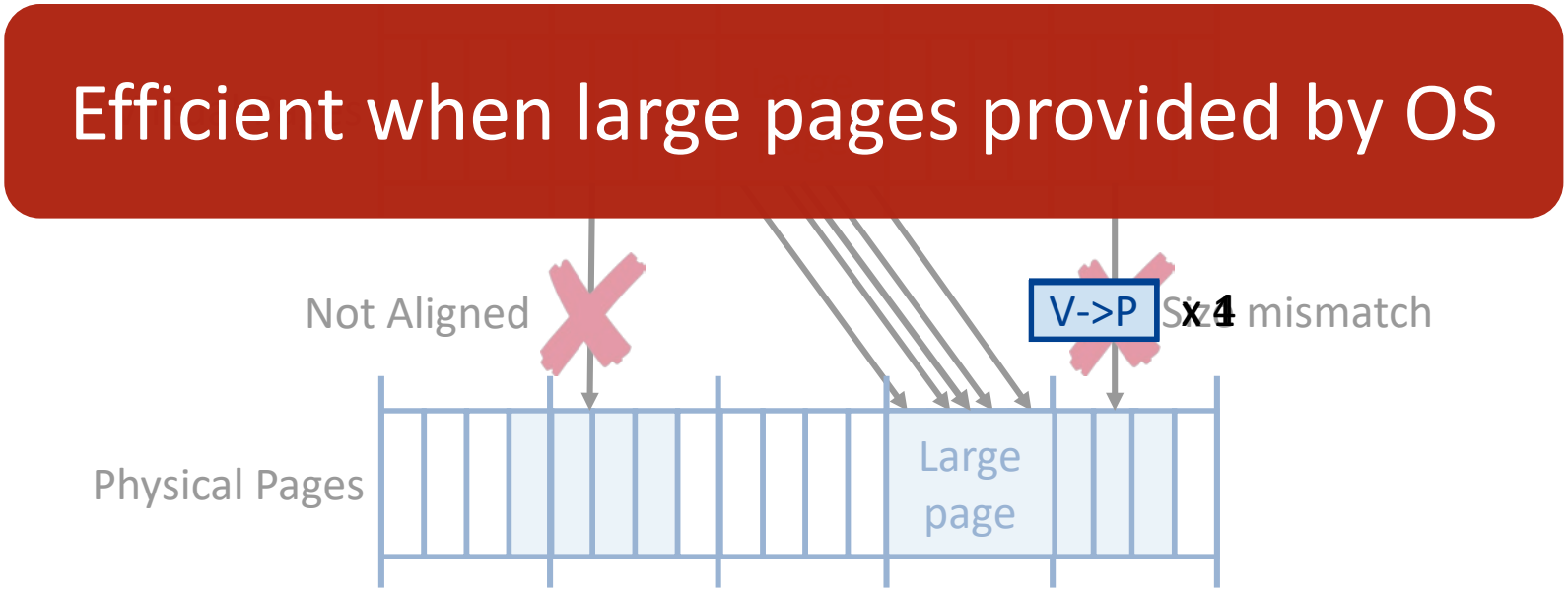
- Large pages represent larger mappings (2MB)
  - Strict alignment required
  - Exact size match required

Efficient when large pages provided by OS



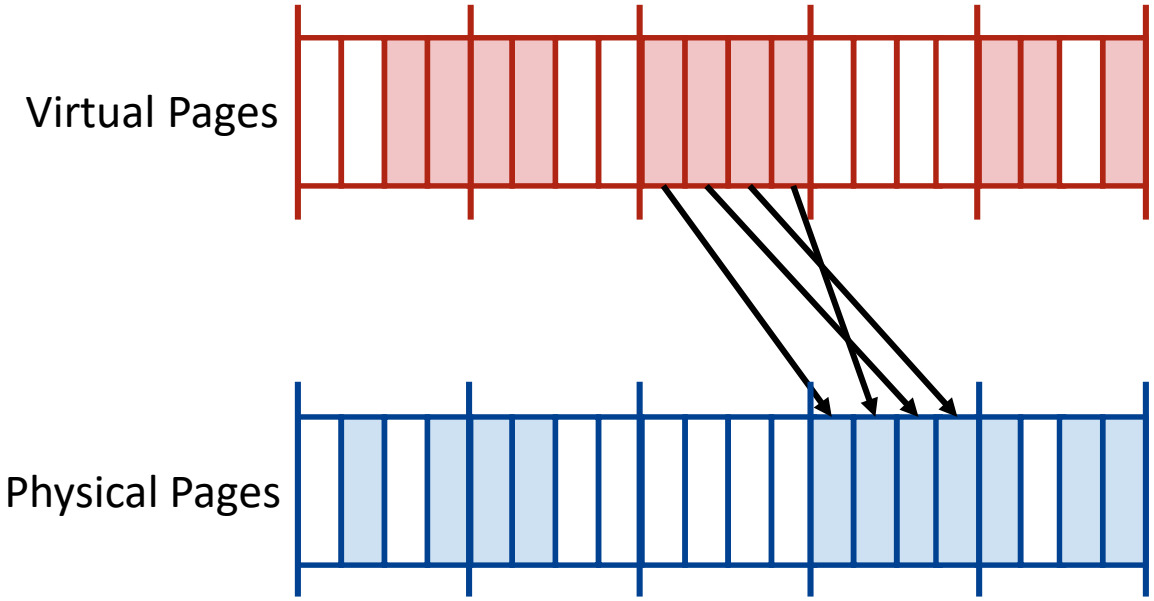
Past Proposals: Large pages

- Large pages represent larger mappings (2MB)
  - Strict alignment required
  - Exact size match required



Past Proposals: Cluster TLB

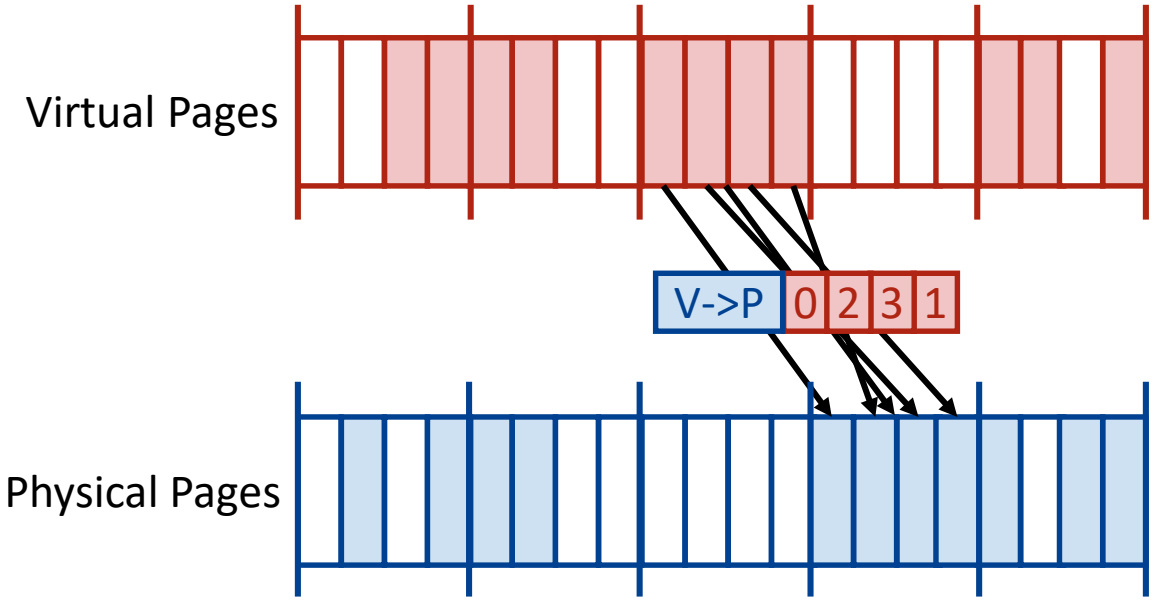
- HW oriented clustering[5]
- Cluster TLB represents flexible mapping within cluster
  - Provides flexible mapping within cluster block
  - However cluster size is fixed at design time



[5] Pham et al. HPCA ’14

Past Proposals: Cluster TLB

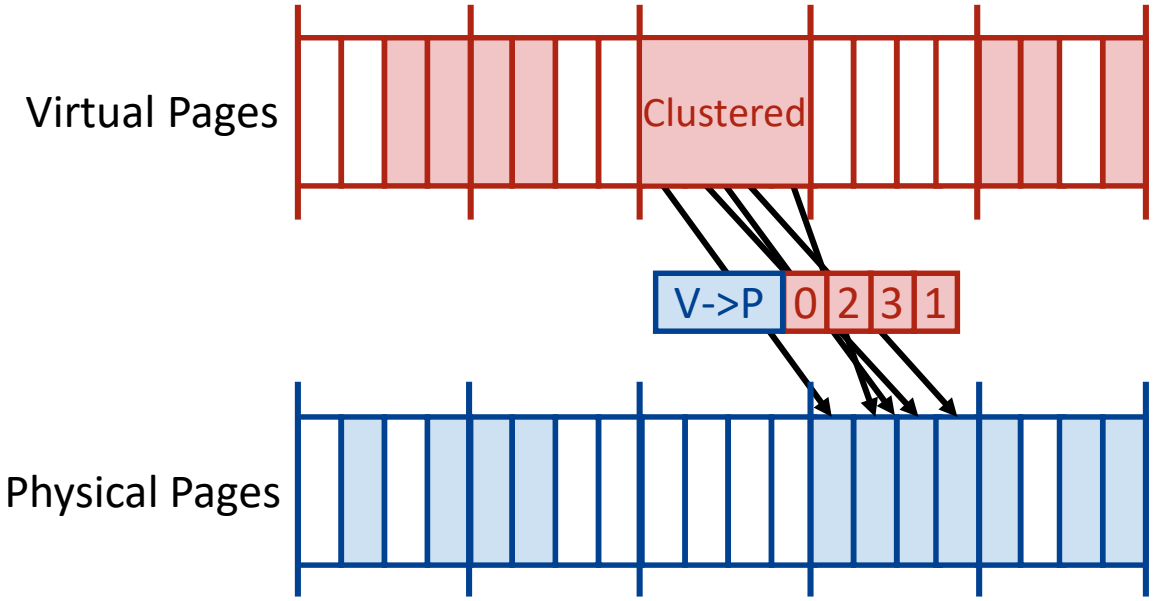
- HW oriented clustering[5]
- Cluster TLB represents flexible mapping within cluster
  - Provides flexible mapping within cluster block
  - However cluster size is fixed at design time



[5] Pham et al. HPCA '14

Past Proposals: Cluster TLB

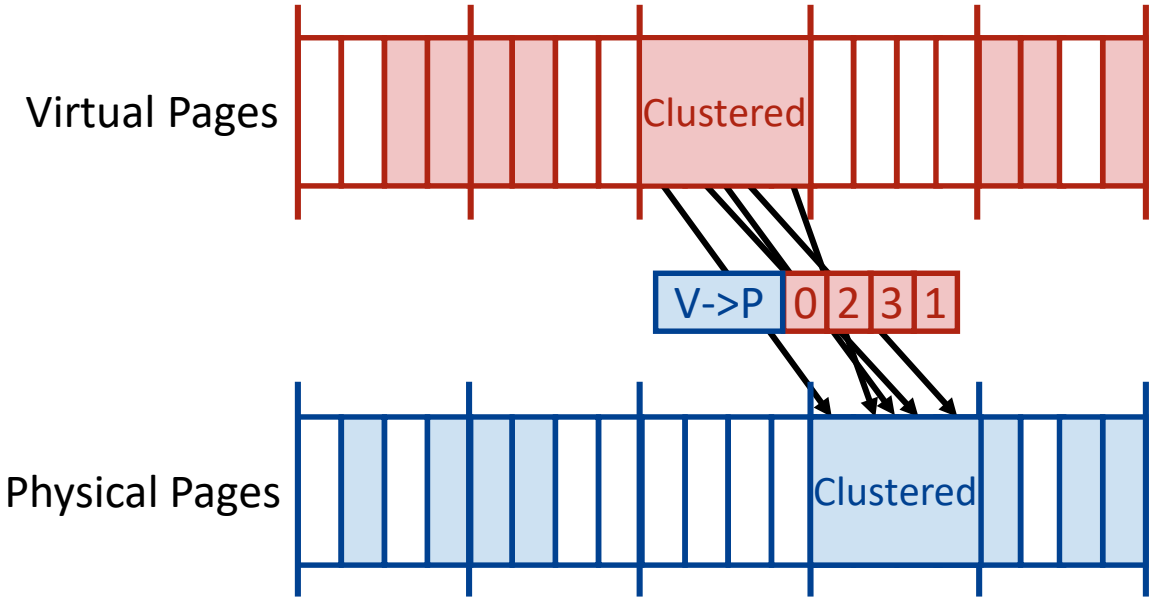
- HW oriented clustering[5]
- Cluster TLB represents flexible mapping within cluster
  - Provides flexible mapping within cluster block
  - However cluster size is fixed at design time



[5] Pham et al. HPCA '14

Past Proposals: Cluster TLB

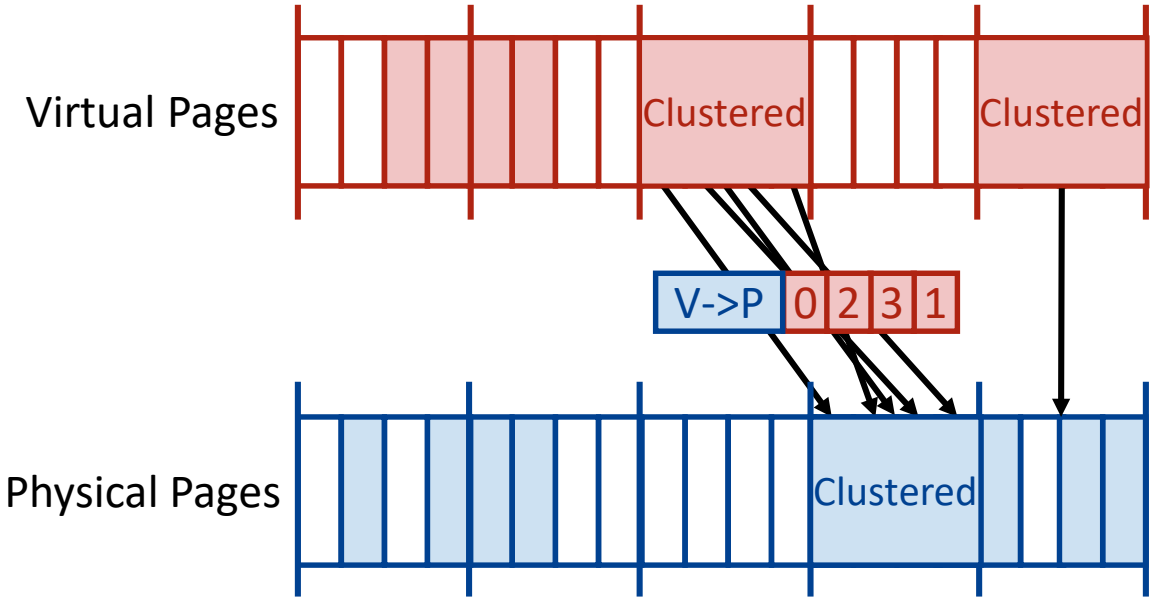
- HW oriented clustering[5]
- Cluster TLB represents flexible mapping within cluster
  - Provides flexible mapping within cluster block
  - However cluster size is fixed at design time



[5] Pham et al. HPCA '14

Past Proposals: Cluster TLB

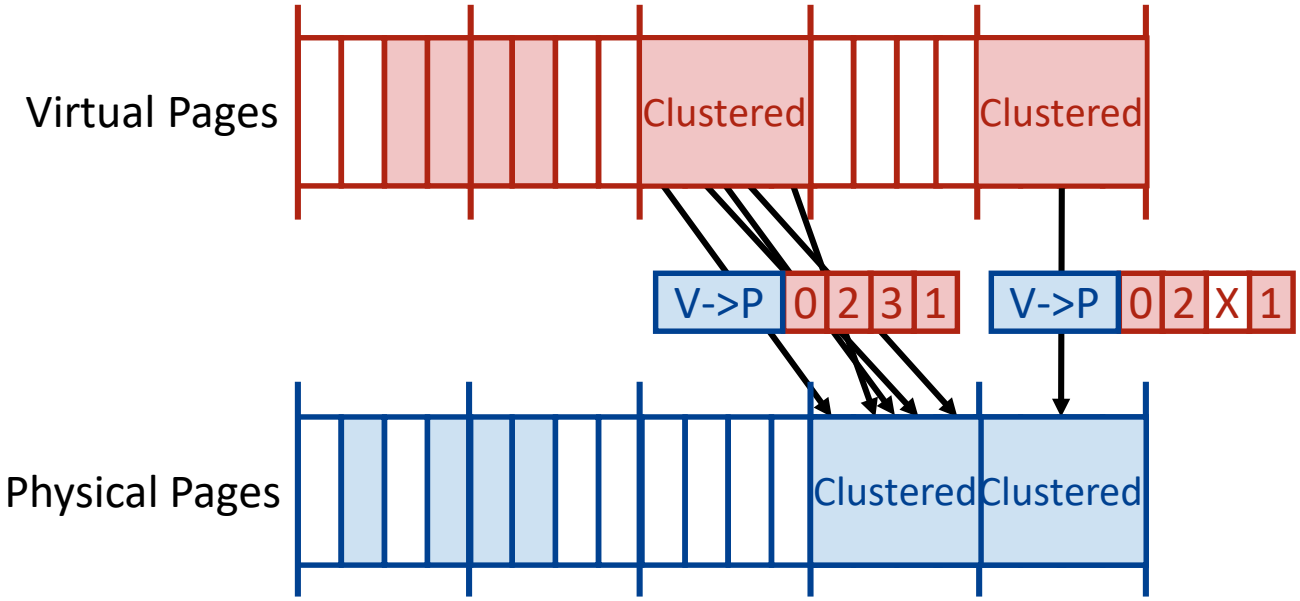
- HW oriented clustering[5]
- Cluster TLB represents flexible mapping within cluster
  - Provides flexible mapping within cluster block
  - However cluster size is fixed at design time



[5] Pham et al. HPCA '14

Past Proposals: Cluster TLB

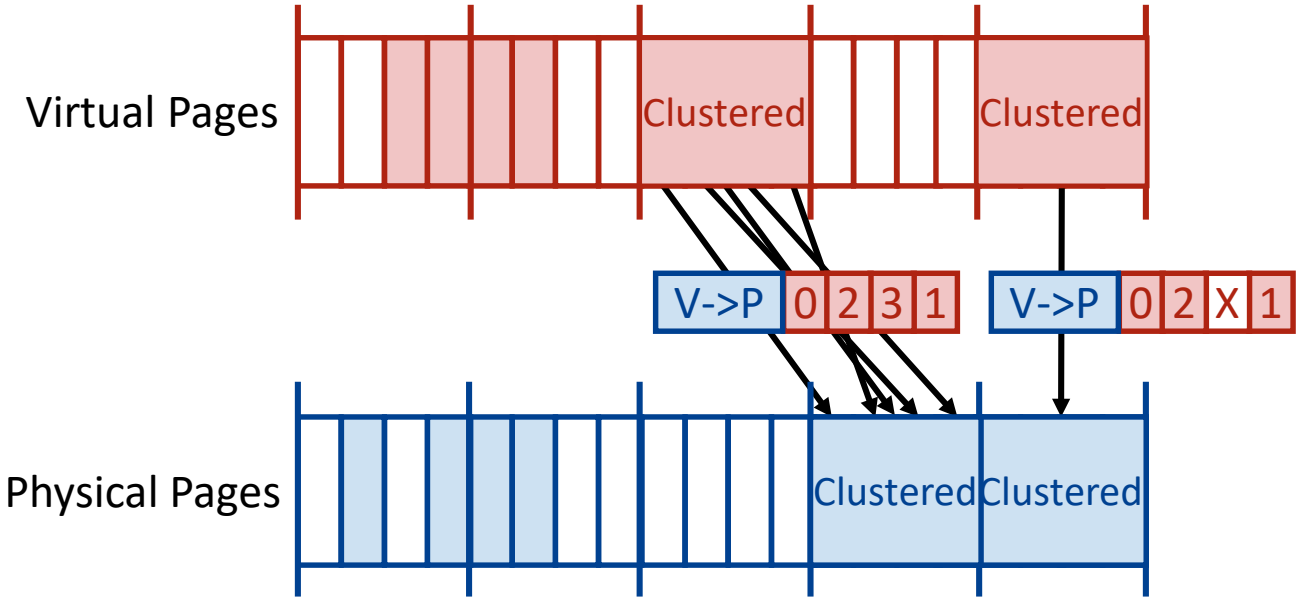
- HW oriented clustering[5]
- Cluster TLB represents flexible mapping within cluster
  - Provides flexible mapping within cluster block
  - However cluster size is fixed at design time



[5] Pham et al. HPCA '14

Past Proposals: Cluster TLB

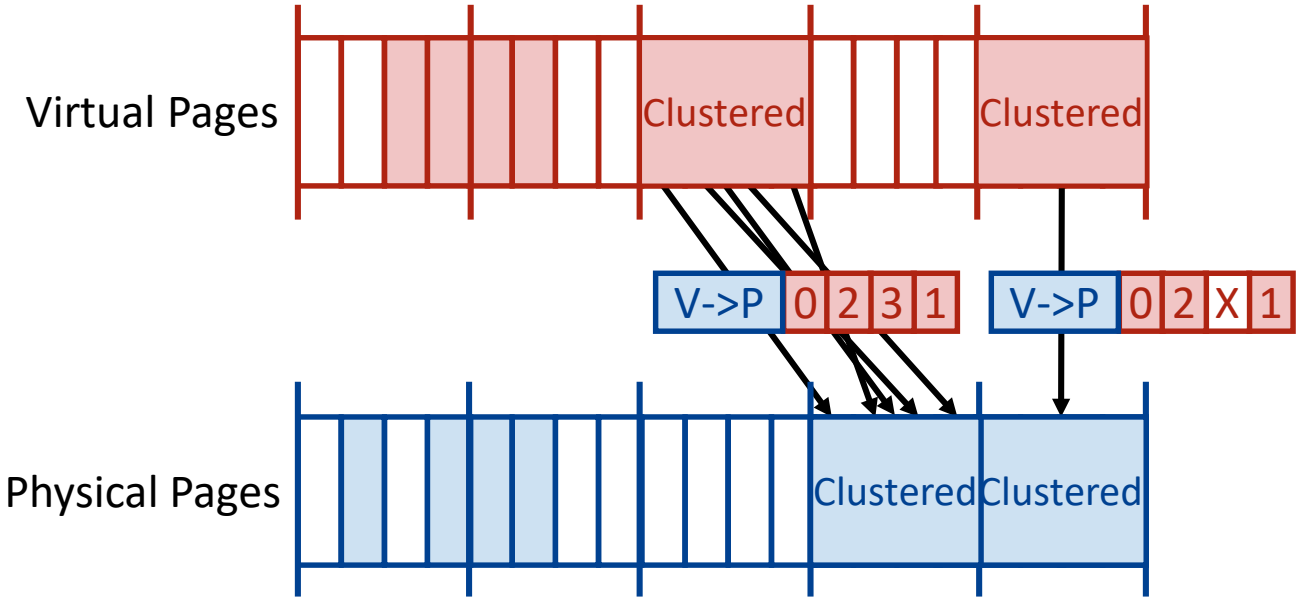
- HW oriented clustering[5]
- Cluster TLB represents flexible mapping within cluster
  - Provides flexible mapping within cluster block
  - However cluster size is fixed at design time



[5] Pham et al. HPCA '14

Past Proposals: Cluster TLB

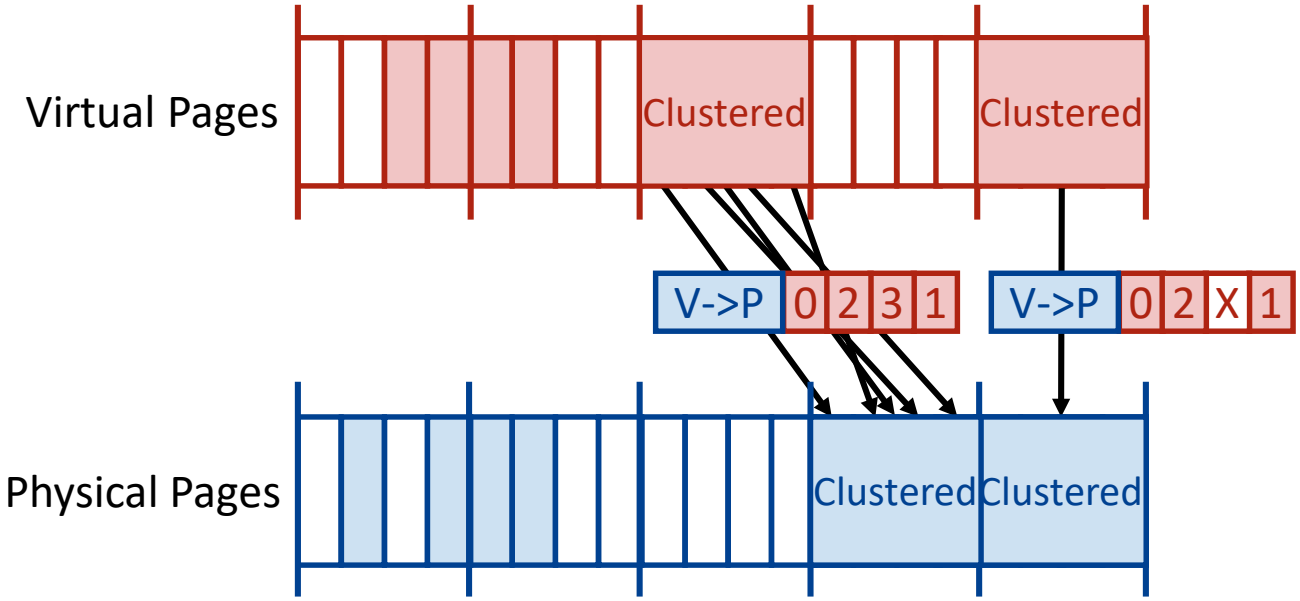
- HW oriented clustering[5]
- Cluster TLB represents flexible mapping within cluster
  - Provides flexible mapping within cluster block
  - However cluster size is fixed at design time



[5] Pham et al. HPCA '14

Past Proposals: Cluster TLB

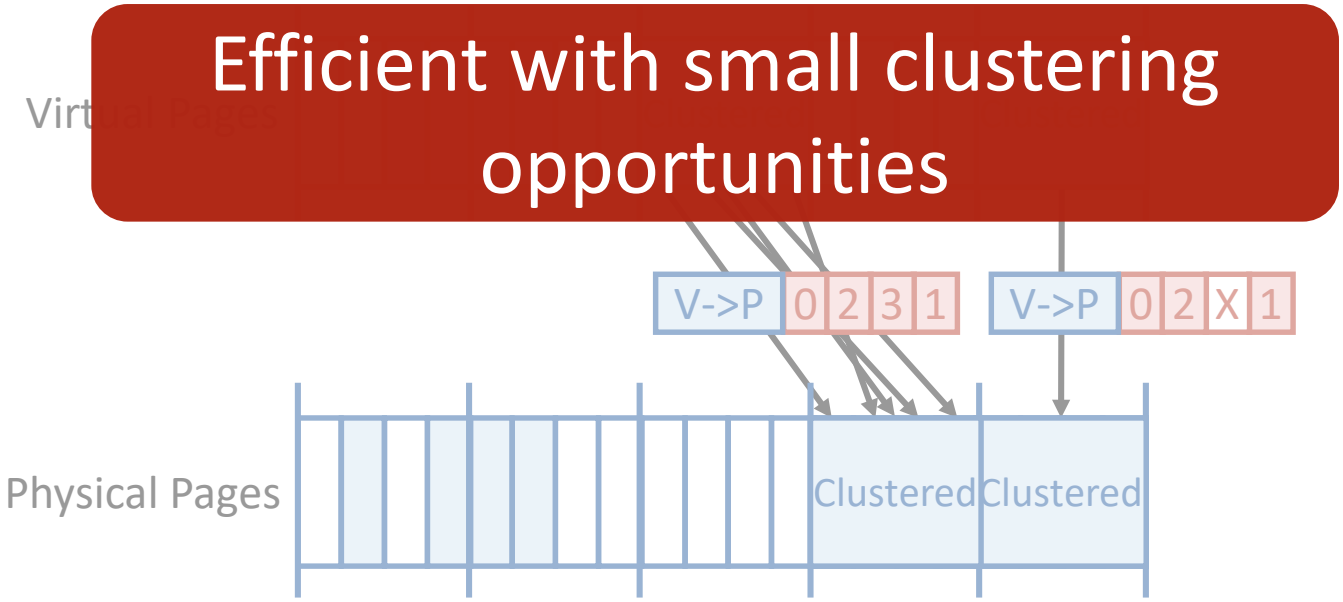
- HW oriented clustering[5]
- Cluster TLB represents flexible mapping within cluster
  - Provides flexible mapping within cluster block
  - However cluster size is fixed at design time



[5] Pham et al. HPCA '14

Past Proposals: Cluster TLB

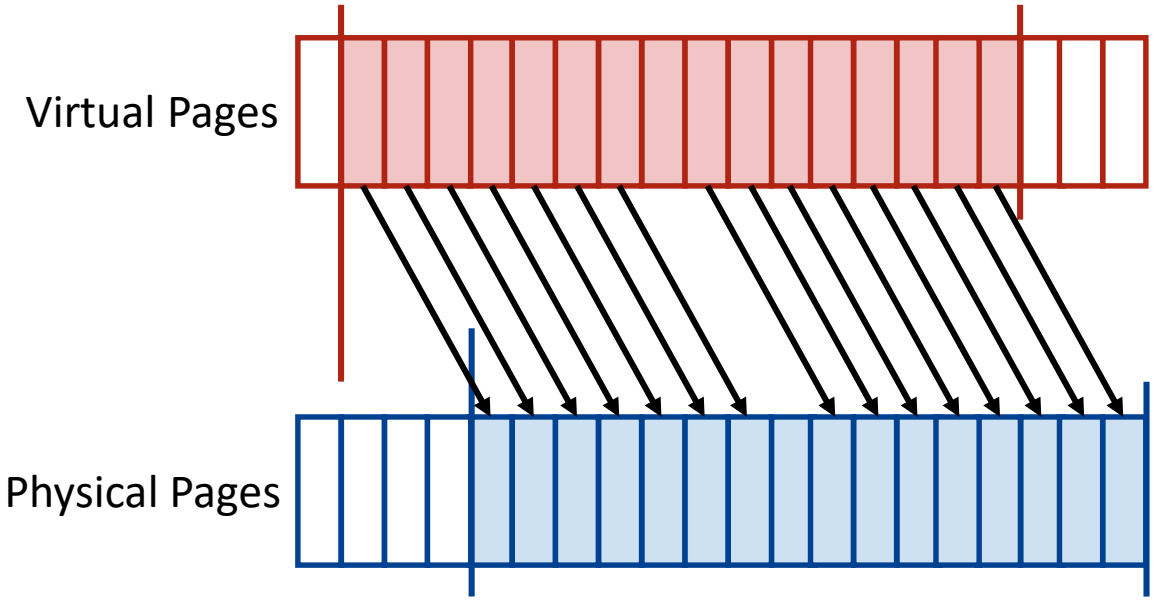
- HW oriented clustering[5]
- Cluster TLB represents flexible mapping within cluster
  - Provides flexible mapping within cluster block
  - However cluster size is fixed at design time



[5] Pham et al. HPCA '14

Past Proposals: Direct Segments

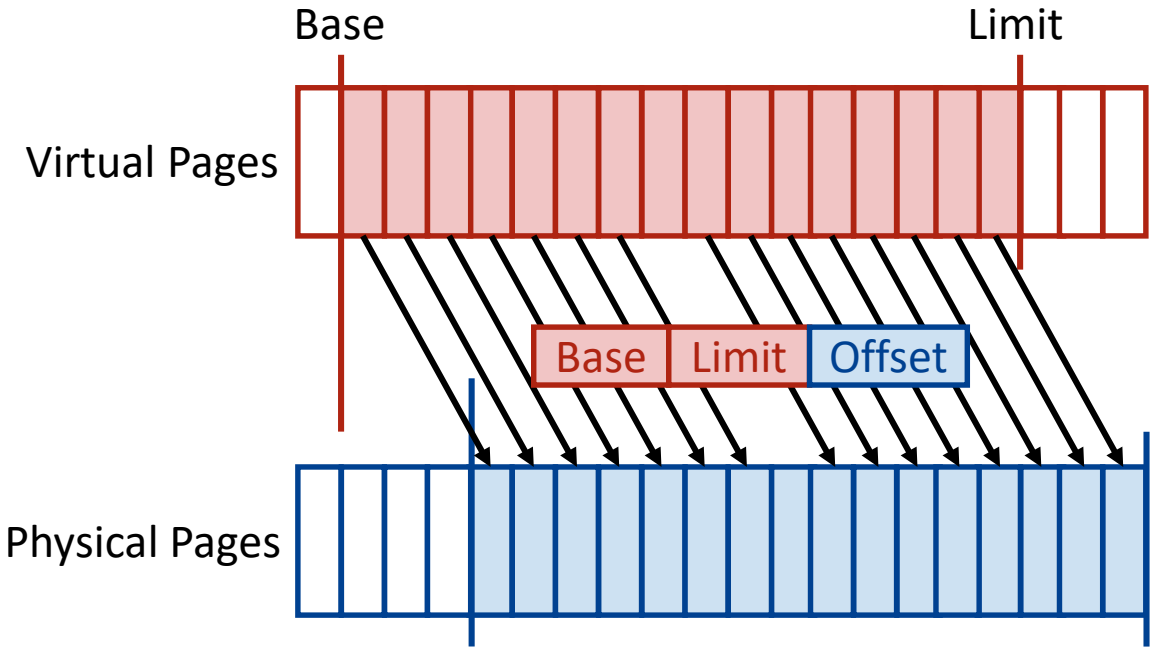
- Segment based translation[1]
  - Three values represent **contiguous** translation of any size
  - Fully assoc. lookup for multiple segments (limits size of TLB)
    - Redundant Memory Mappings (RMM) [6] -> 32 Fully-associative TLB



[1] Basu et al. ISCA '13
[6] Karakostas et al. ISCA '15

Past Proposals: Direct Segments

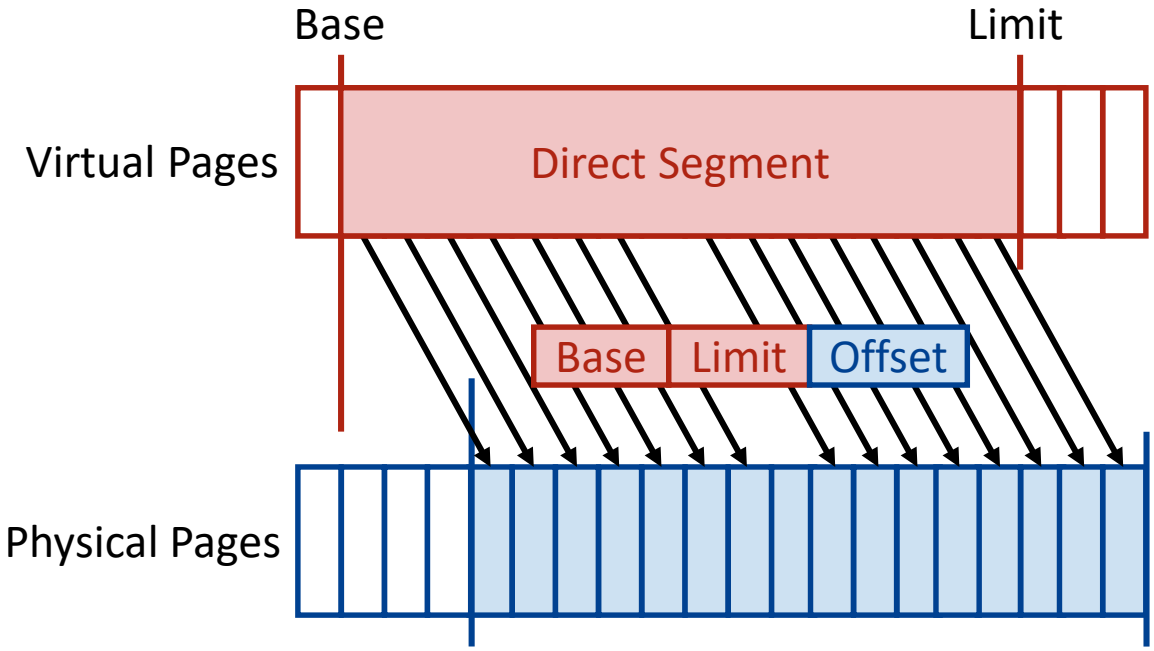
- Segment based translation[1]
  - Three values represent **contiguous** translation of any size
  - Fully assoc. lookup for multiple segments (limits size of TLB)
    - Redundant Memory Mappings (RMM) [6] -> 32 Fully-associative TLB



[1] Basu et al. ISCA '13
[6] Karakostas et al. ISCA '15

Past Proposals: Direct Segments

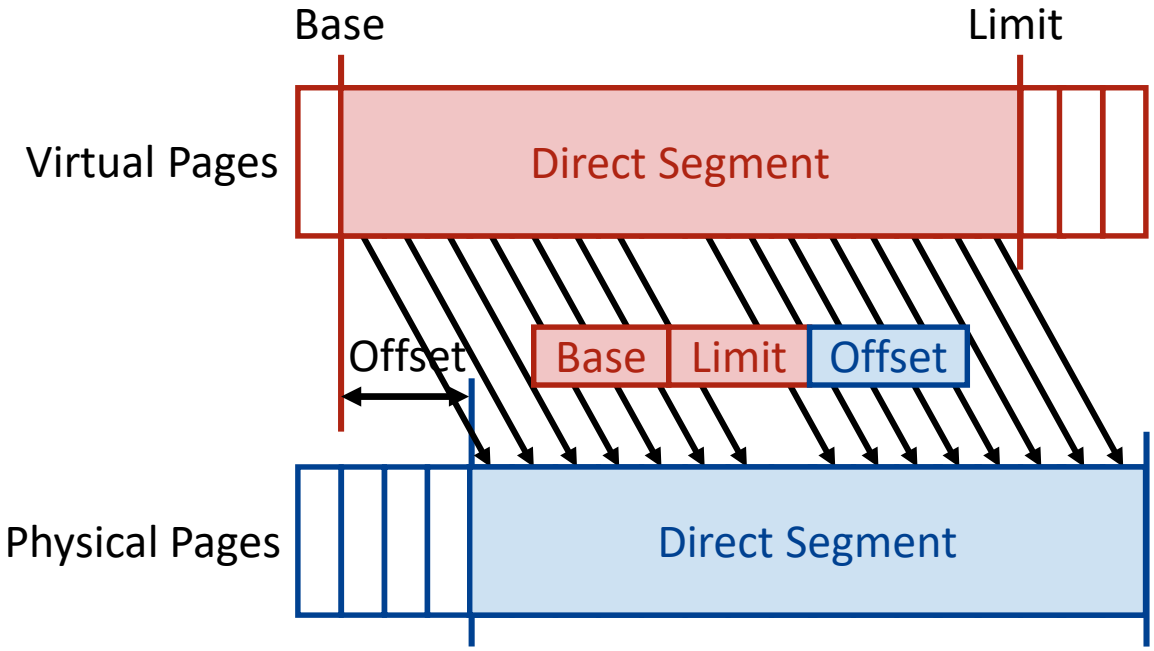
- Segment based translation[1]
  - Three values represent **contiguous** translation of any size
  - Fully assoc. lookup for multiple segments (limits size of TLB)
    - Redundant Memory Mappings (RMM) [6] -> 32 Fully-associative TLB



[1] Basu et al. ISCA '13
[6] Karakostas et al. ISCA '15

Past Proposals: Direct Segments

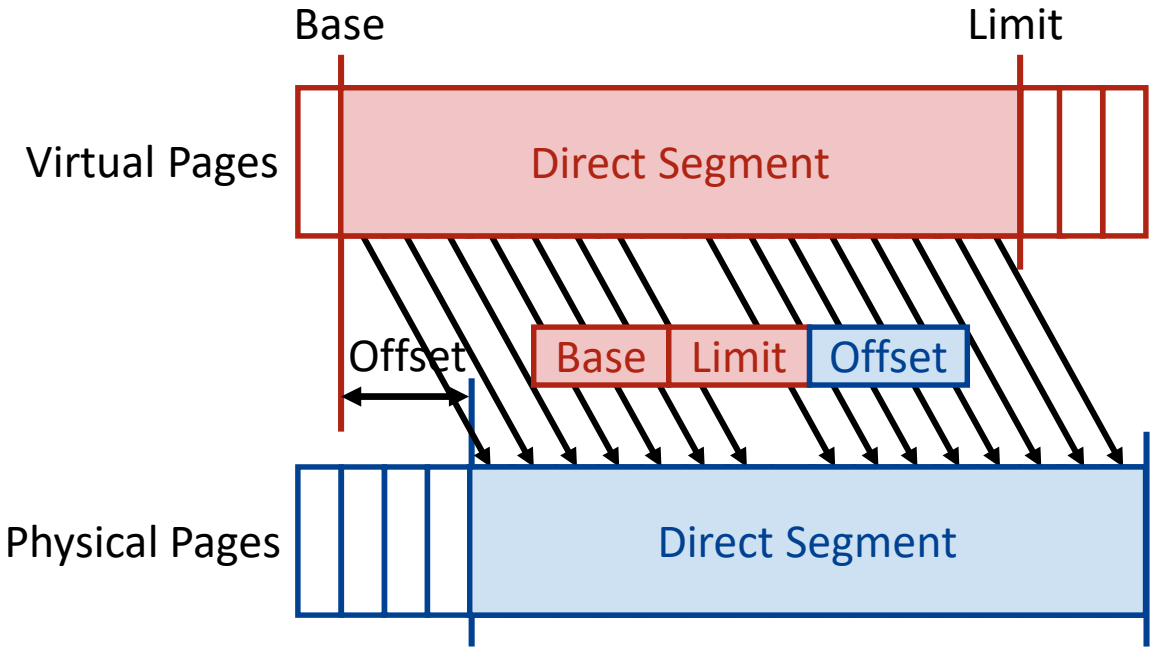
- Segment based translation[1]
  - Three values represent **contiguous** translation of any size
  - Fully assoc. lookup for multiple segments (limits size of TLB)
    - Redundant Memory Mappings (RMM) [6] -> 32 Fully-associative TLB



[1] Basu et al. ISCA '13
[6] Karakostas et al. ISCA '15

Past Proposals: Direct Segments

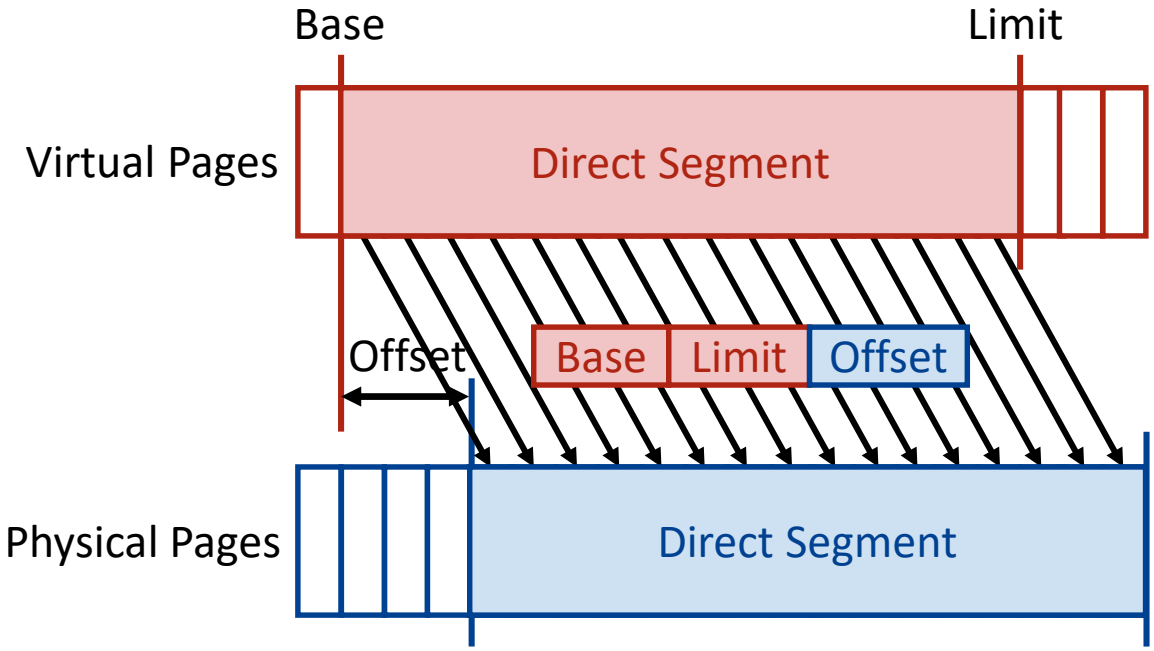
- Segment based translation[1]
  - Three values represent **contiguous** translation of any size
  - Fully assoc. lookup for multiple segments (limits size of TLB)
    - Redundant Memory Mappings (RMM) [6] -> 32 Fully-associative TLB



[1] Basu et al. ISCA '13
[6] Karakostas et al. ISCA '15

Past Proposals: Direct Segments

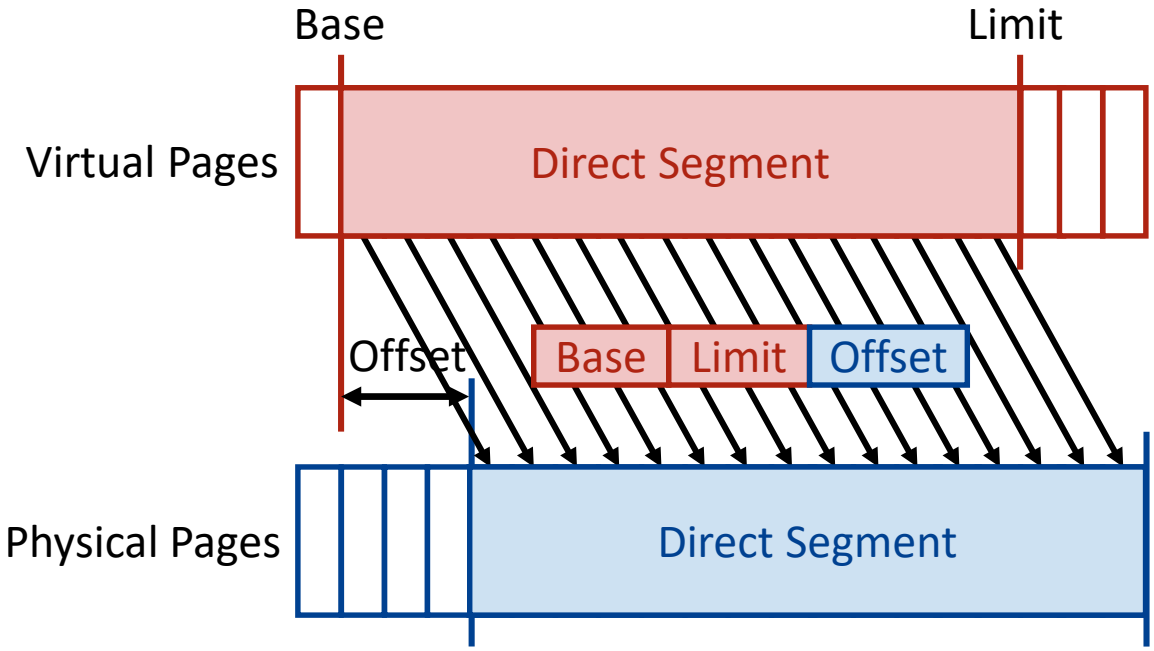
- Segment based translation[1]
  - Three values represent **contiguous** translation of any size
  - Fully assoc. lookup for multiple segments (limits size of TLB)
    - Redundant Memory Mappings (RMM) [6] -> 32 Fully-associative TLB



[1] Basu et al. ISCA '13
[6] Karakostas et al. ISCA '15

Past Proposals: Direct Segments

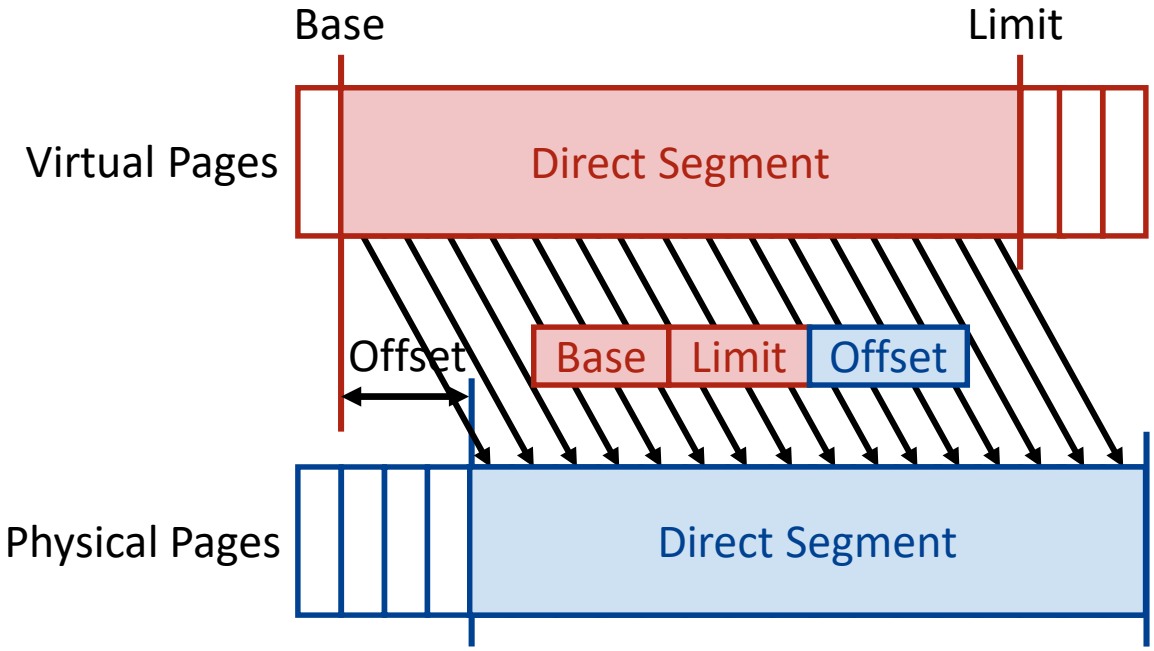
- Segment based translation[1]
  - Three values represent **contiguous** translation of any size
  - Fully assoc. lookup for multiple segments (limits size of TLB)
    - Redundant Memory Mappings (RMM) [6] -> 32 Fully-associative TLB



[1] Basu et al. ISCA '13
[6] Karakostas et al. ISCA '15

Past Proposals: Direct Segments

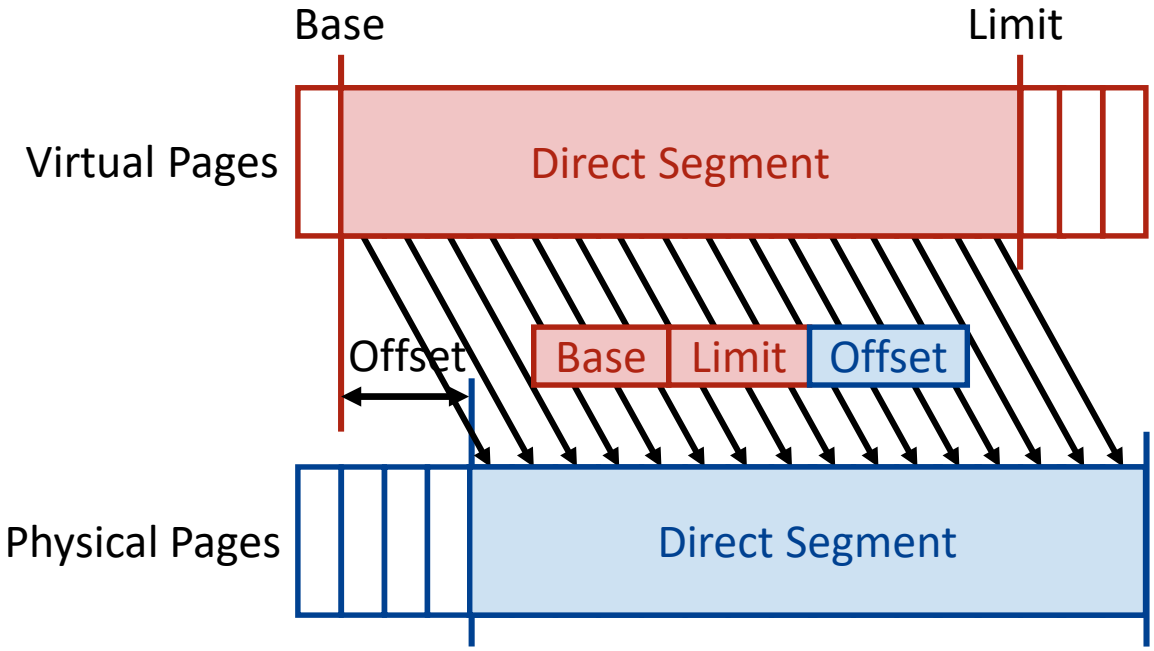
- Segment based translation[1]
  - Three values represent **contiguous** translation of any size
  - Fully assoc. lookup for multiple segments (limits size of TLB)
    - Redundant Memory Mappings (RMM) [6] -> 32 Fully-associative TLB



[1] Basu et al. ISCA '13
[6] Karakostas et al. ISCA '15

Past Proposals: Direct Segments

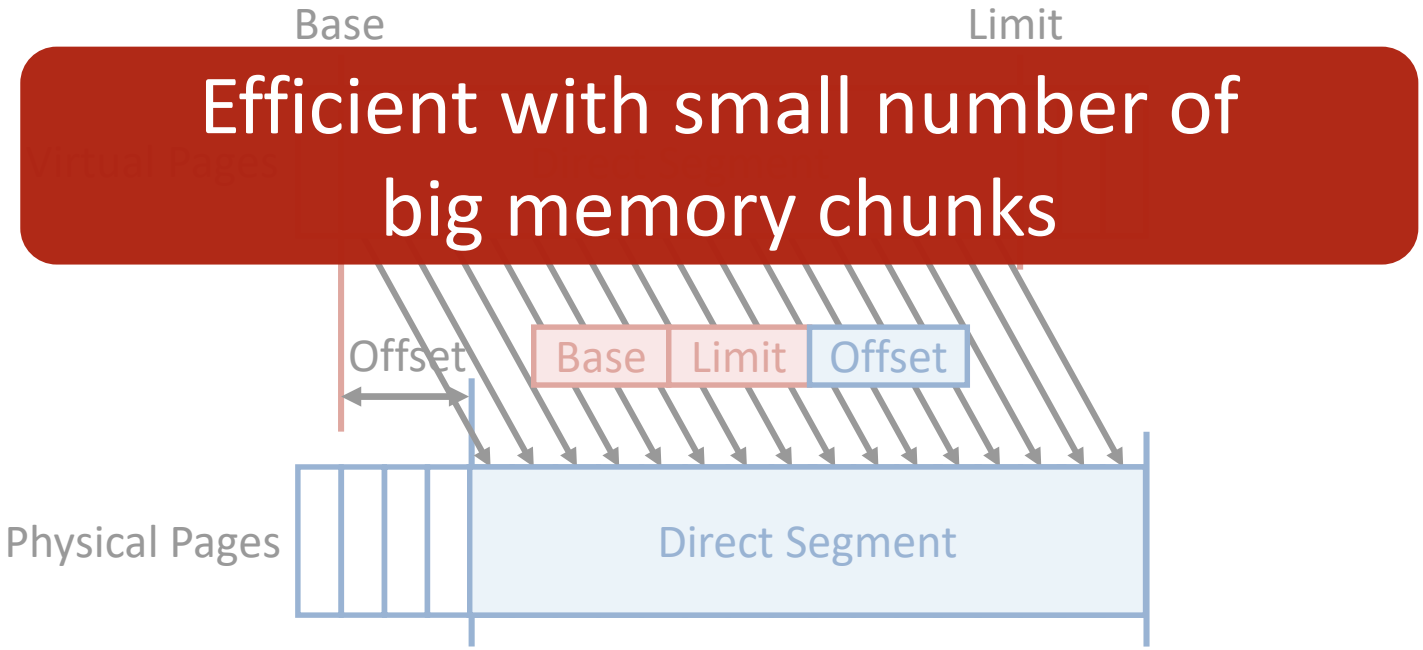
- Segment based translation[1]
  - Three values represent **contiguous** translation of any size
  - Fully assoc. lookup for multiple segments (limits size of TLB)
    - Redundant Memory Mappings (RMM) [6] -> 32 Fully-associative TLB



[1] Basu et al. ISCA '13
[6] Karakostas et al. ISCA '15

Past Proposals: Direct Segments

- Segment based translation[1]
  - Three values represent **contiguous** translation of any size
  - Fully assoc. lookup for multiple segments (limits size of TLB)
    - Redundant Memory Mappings (RMM) [6] -> 32 Fully-associative TLB



[1] Basu et al. ISCA '13
[6] Karakostas et al. ISCA '15

Past Proposals: Summary

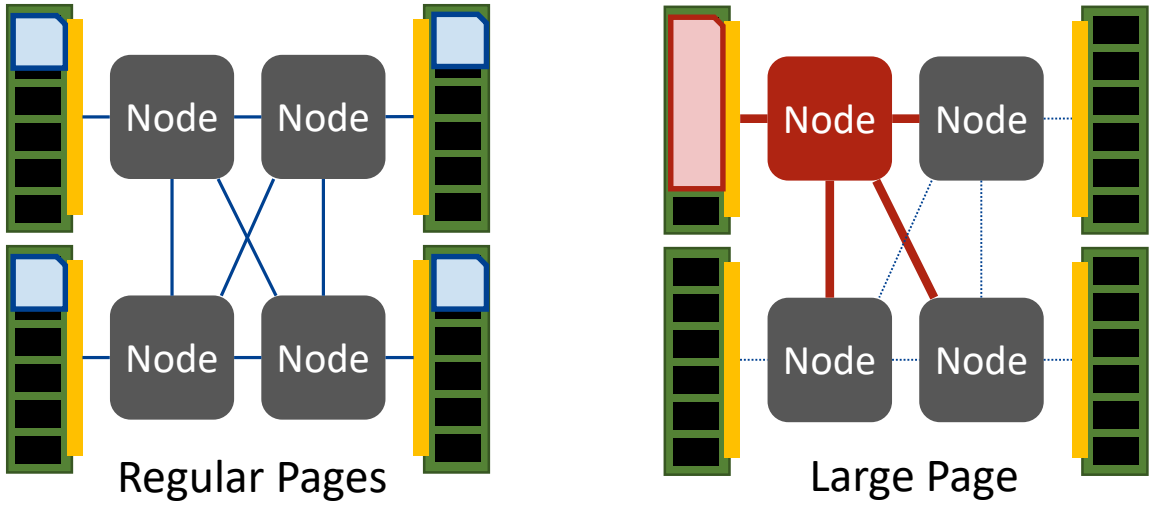
- Large pages
  - Affinity for large pages (2MB)
- Cluster TLB
  - Affinity for clustering of mapping of up to 8 pages
- Segment translations
  - Affinity for small number of large chunks (32 entry TLB)

Past Proposals: Summary

- Large pages
  - Affinity for large pages (2MB)
- Cluster TLB
  - Affinity for clustering of mapping of up to 8 pages
- Segment translations
  - Affinity for small number of large chunks (32 entry TLB)

Prior proposals efficiently support **specific** memory mapping scenarios

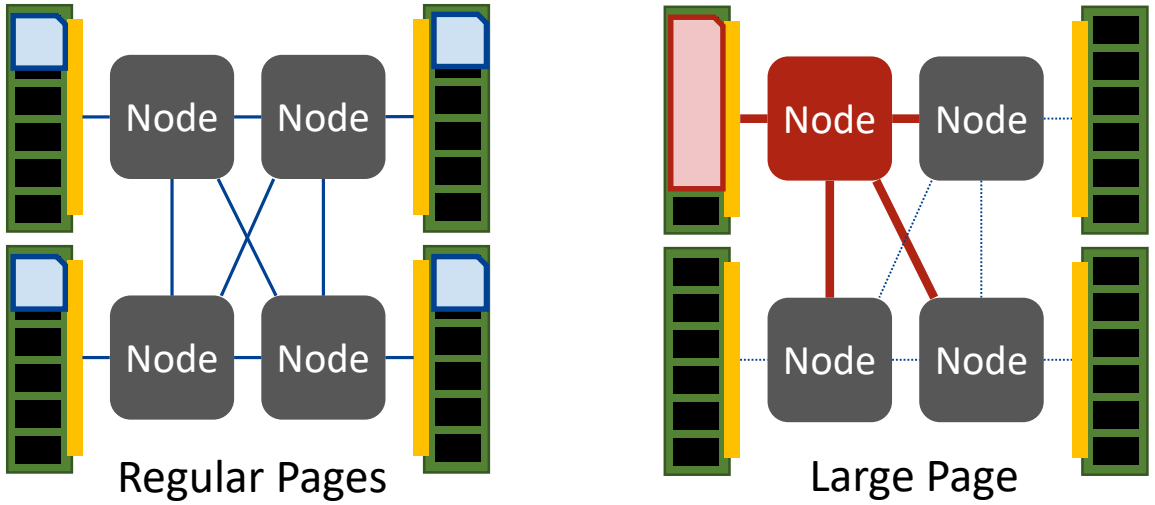
Large Contiguity vs. Memory Non-Uniformity



[2] Baptiste et al. ATC ’14
[3] Lee et al. ISCA ‘15
[4] Agarwal et al. ASPLOS ‘17

Large Contiguity vs. Memory Non-Uniformity

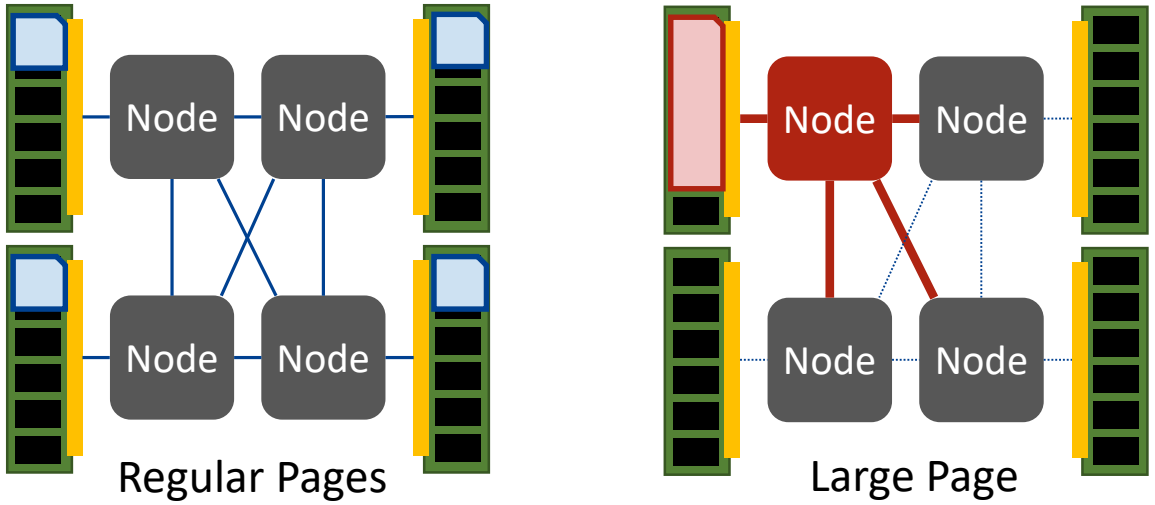
- Conflicting goals of NUMA systems and large pages[2]
  - Memory traffic balance vs. efficient address translation



[2] Baptiste et al. ATC ’14
[3] Lee et al. ISCA ‘15
[4] Agarwal et al. ASPLOS ‘17

Large Contiguity vs. Memory Non-Uniformity

- Conflicting goals of NUMA systems and large pages[2]
  - Memory traffic balance vs. efficient address translation

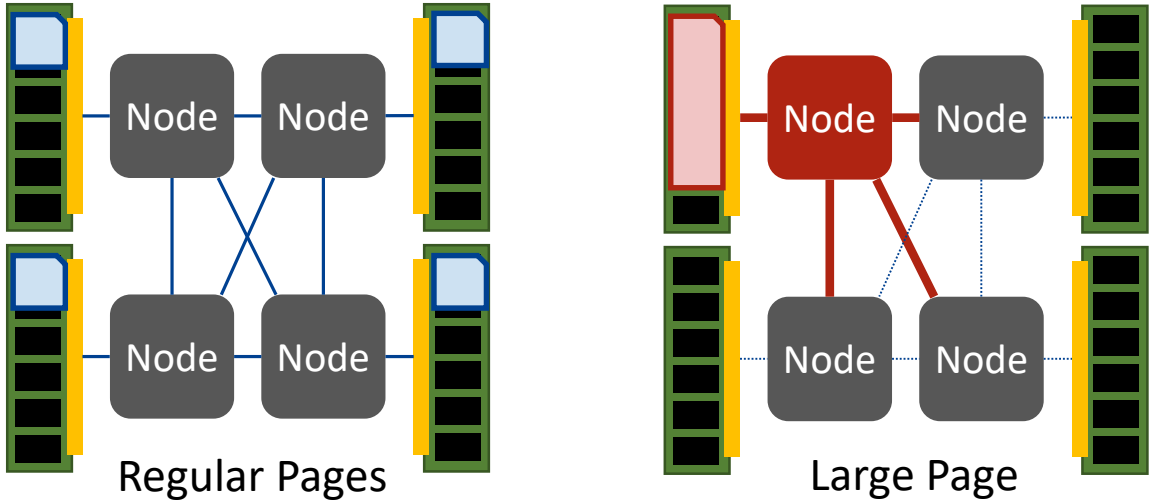


- Heterogeneous memory worsens non-uniformity [3][4]

[2] Baptiste et al. ATC ’14
[3] Lee et al. ISCA ‘15
[4] Agarwal et al. ASPLOS ‘17

Large Contiguity vs. Memory Non-Uniformity

- Conflicting goals of NUMA systems and large pages[2]
  - Memory traffic balance vs. efficient address translation

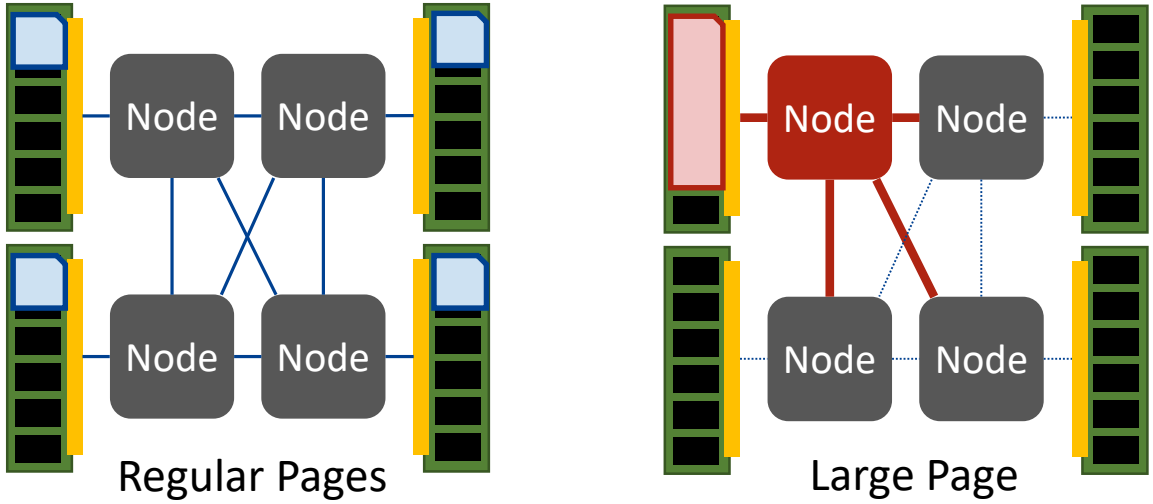


- Heterogeneous memory worsens non-uniformity [3][4]

[2] Baptiste et al. ATC '14
[3] Lee et al. ISCA '15
[4] Agarwal et al. ASPLOS '17

Large Contiguity vs. Memory Non-Uniformity

- Conflicting goals of NUMA systems and large pages[2]
  - Memory traffic balance vs. efficient address translation

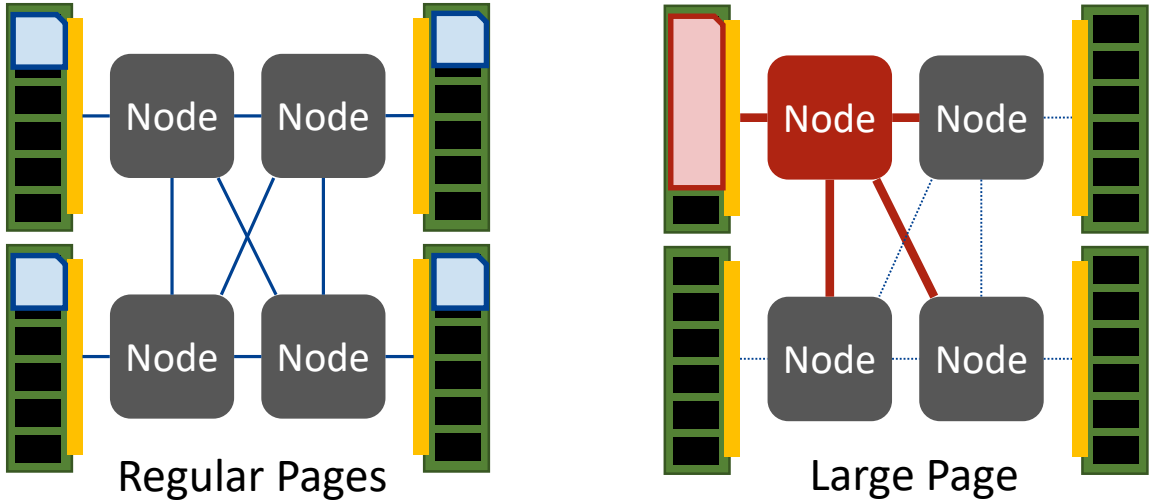


- Heterogeneous memory worsens non-uniformity [3][4]

[2] Baptiste et al. ATC '14
[3] Lee et al. ISCA '15
[4] Agarwal et al. ASPLOS '17

Large Contiguity vs. Memory Non-Uniformity

- Conflicting goals of NUMA systems and large pages[2]
  - Memory traffic balance vs. efficient address translation

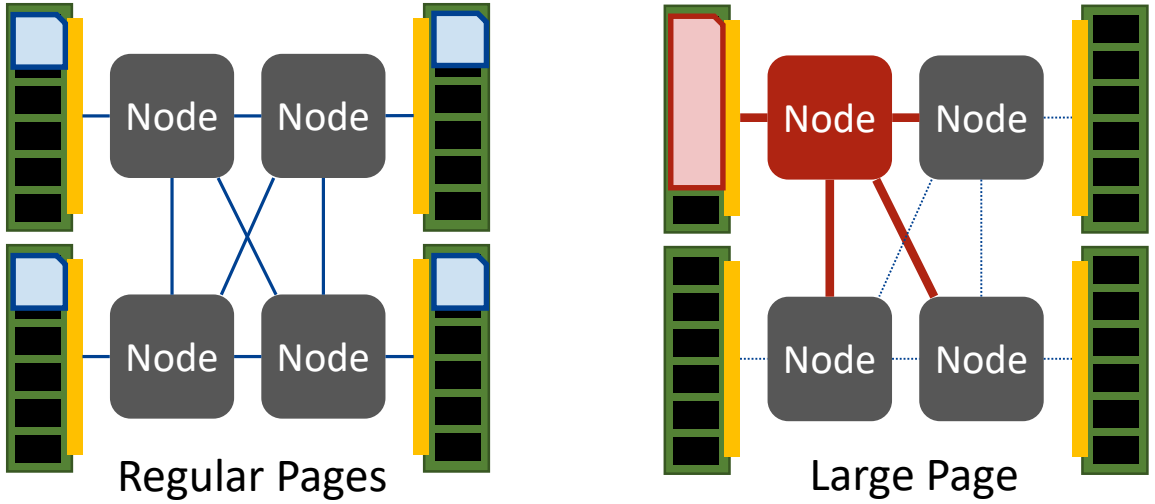


- Heterogeneous memory worsens non-uniformity [3][4]

[2] Baptiste et al. ATC '14
[3] Lee et al. ISCA '15
[4] Agarwal et al. ASPLOS '17

Large Contiguity vs. Memory Non-Uniformity

- Conflicting goals of NUMA systems and large pages[2]
  - Memory traffic balance vs. efficient address translation

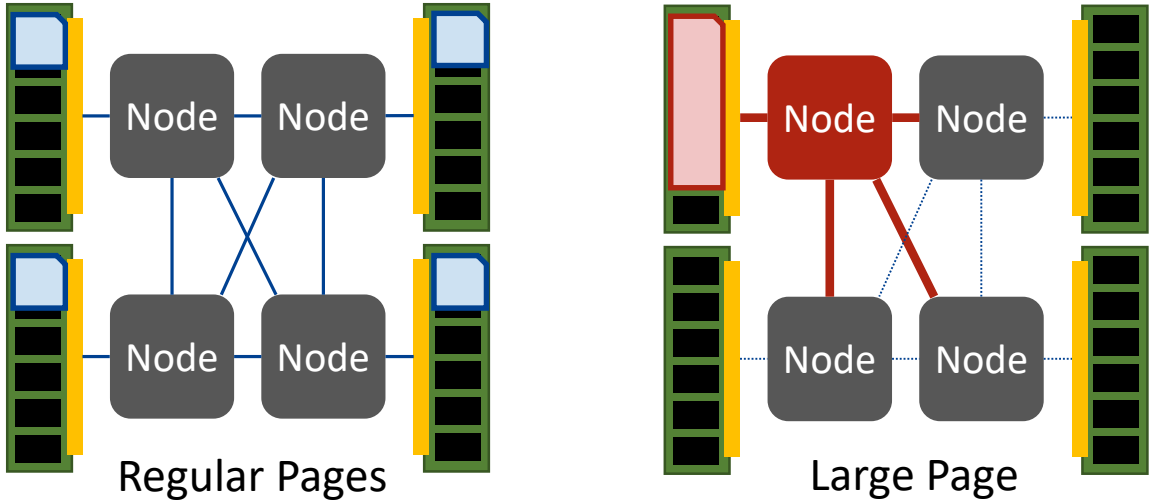


- Heterogeneous memory worsens non-uniformity [3][4]

[2] Baptiste et al. ATC '14
[3] Lee et al. ISCA '15
[4] Agarwal et al. ASPLOS '17

Large Contiguity vs. Memory Non-Uniformity

- Conflicting goals of NUMA systems and large pages[2]
  - Memory traffic balance vs. efficient address translation

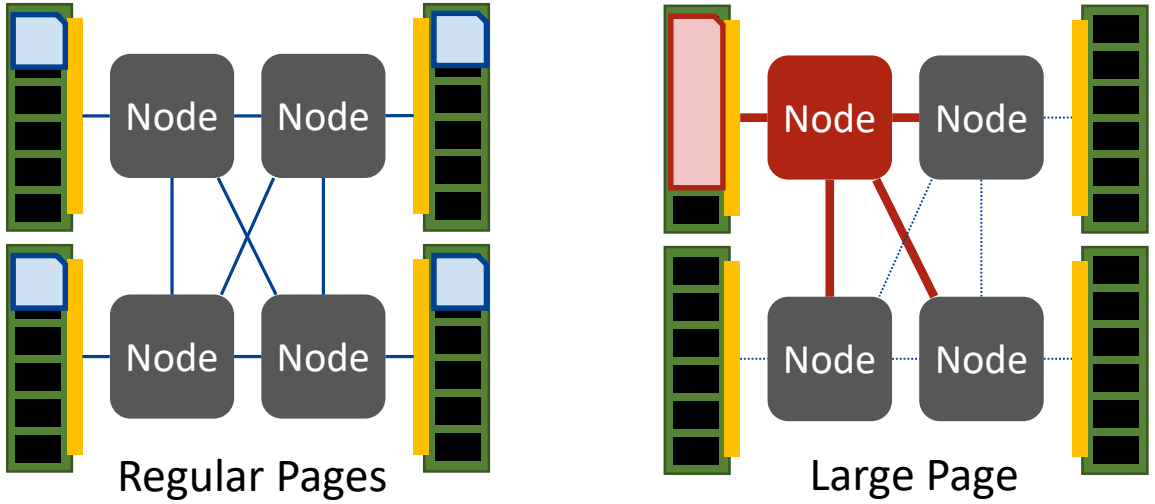


- Heterogeneous memory worsens non-uniformity [3][4]

[2] Baptiste et al. ATC '14
[3] Lee et al. ISCA '15
[4] Agarwal et al. ASPLOS '17

Large Contiguity vs. Memory Non-Uniformity

- Conflicting goals of NUMA systems and large pages[2]
  - Memory traffic balance vs. efficient address translation

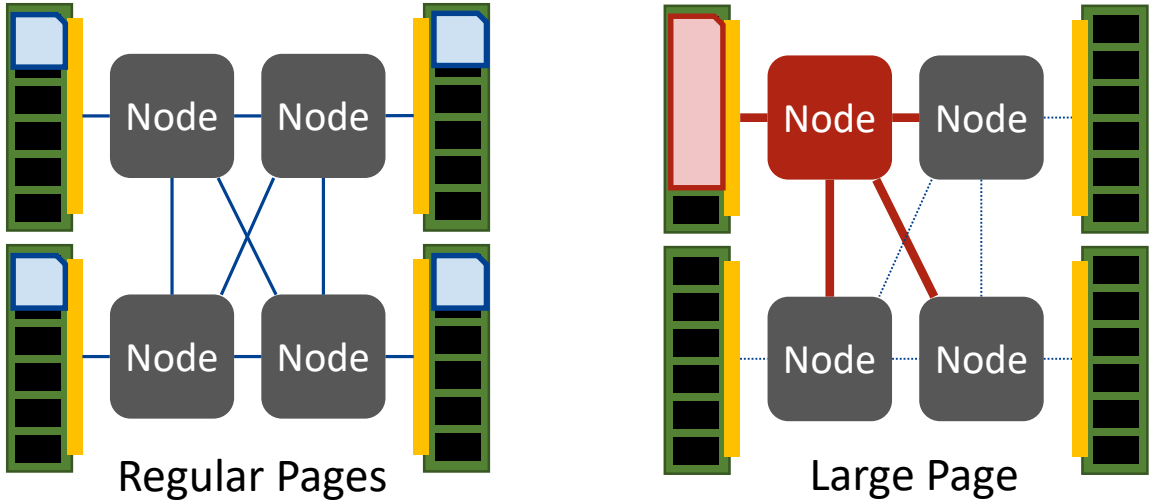


- Heterogeneous memory worsens non-uniformity [3][4]

[2] Baptiste et al. ATC ’14
[3] Lee et al. ISCA ‘15
[4] Agarwal et al. ASPLOS ‘17

Large Contiguity vs. Memory Non-Uniformity

- Conflicting goals of NUMA systems and large pages[2]
  - Memory traffic balance vs. efficient address translation

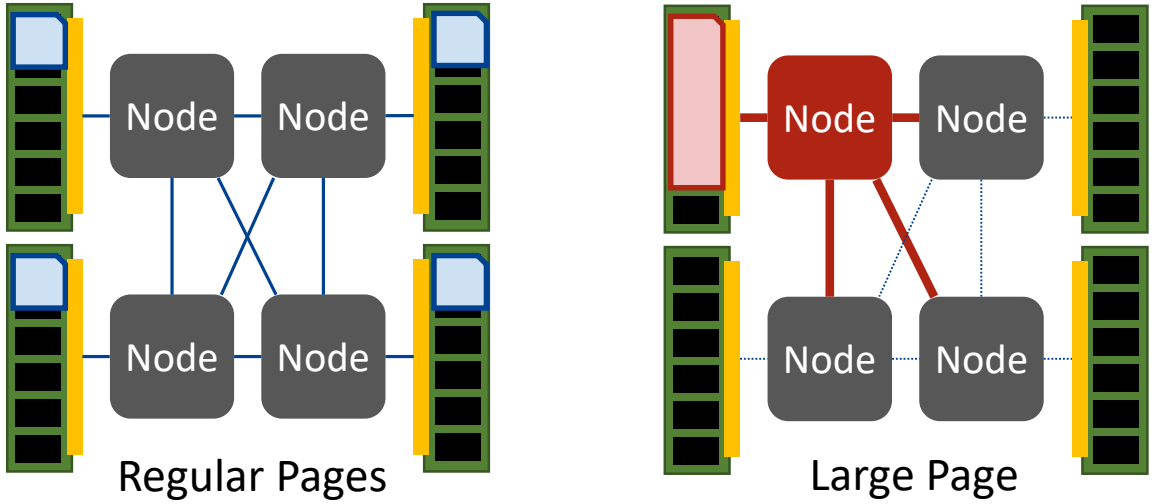


- Heterogeneous memory worsens non-uniformity [3][4]

[2] Baptiste et al. ATC ’14
[3] Lee et al. ISCA ‘15
[4] Agarwal et al. ASPLOS ‘17

Large Contiguity vs. Memory Non-Uniformity

- Conflicting goals of NUMA systems and large pages[2]
  - Memory traffic balance vs. efficient address translation

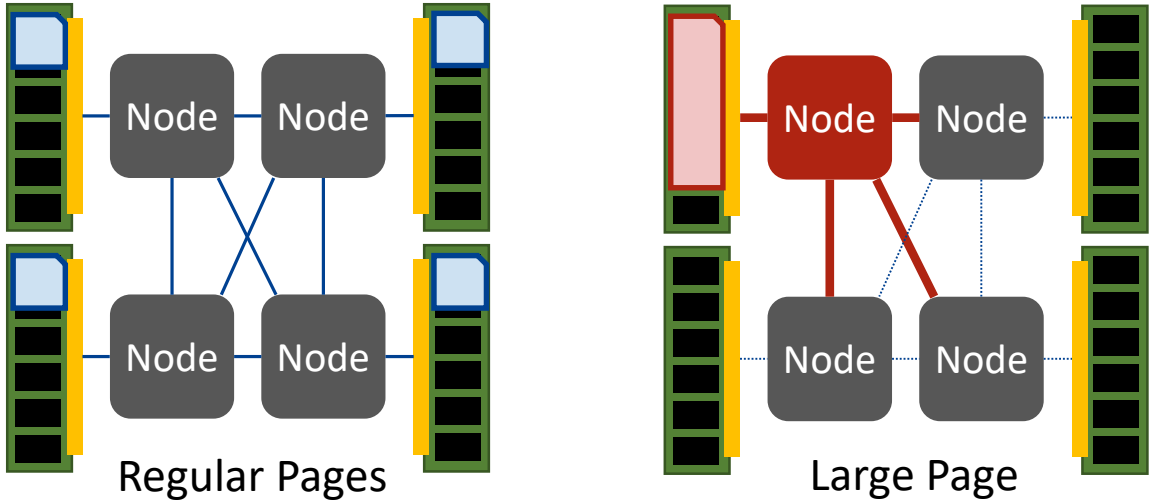


- Heterogeneous memory worsens non-uniformity [3][4]

[2] Baptiste et al. ATC ’14
[3] Lee et al. ISCA ‘15
[4] Agarwal et al. ASPLOS ‘17

Large Contiguity vs. Memory Non-Uniformity

- Conflicting goals of NUMA systems and large pages[2]
  - Memory traffic balance vs. efficient address translation

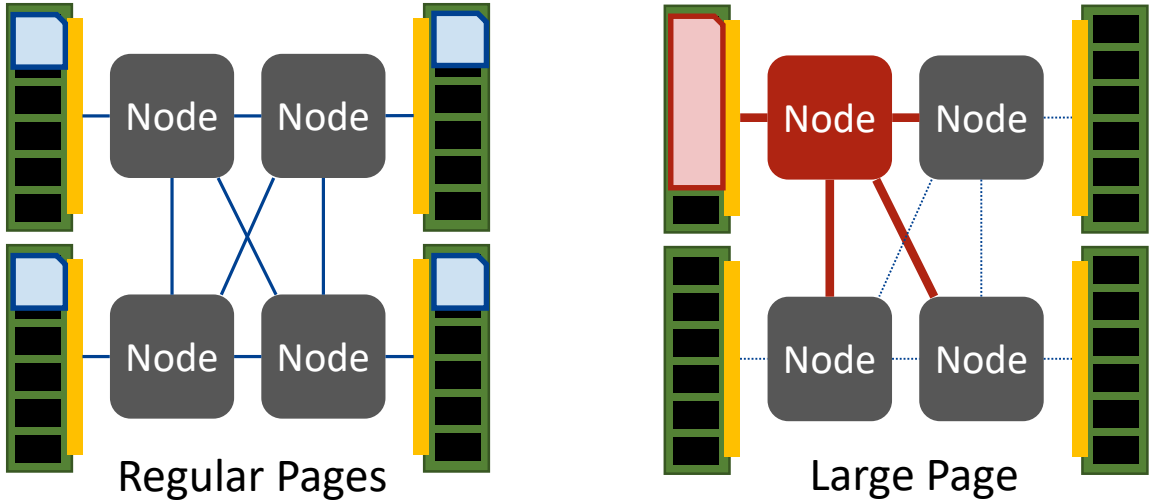


- Heterogeneous memory worsens non-uniformity [3][4]

[2] Baptiste et al. ATC ’14
[3] Lee et al. ISCA ‘15
[4] Agarwal et al. ASPLOS ‘17

Large Contiguity vs. Memory Non-Uniformity

- Conflicting goals of NUMA systems and large pages[2]
  - Memory traffic balance vs. efficient address translation

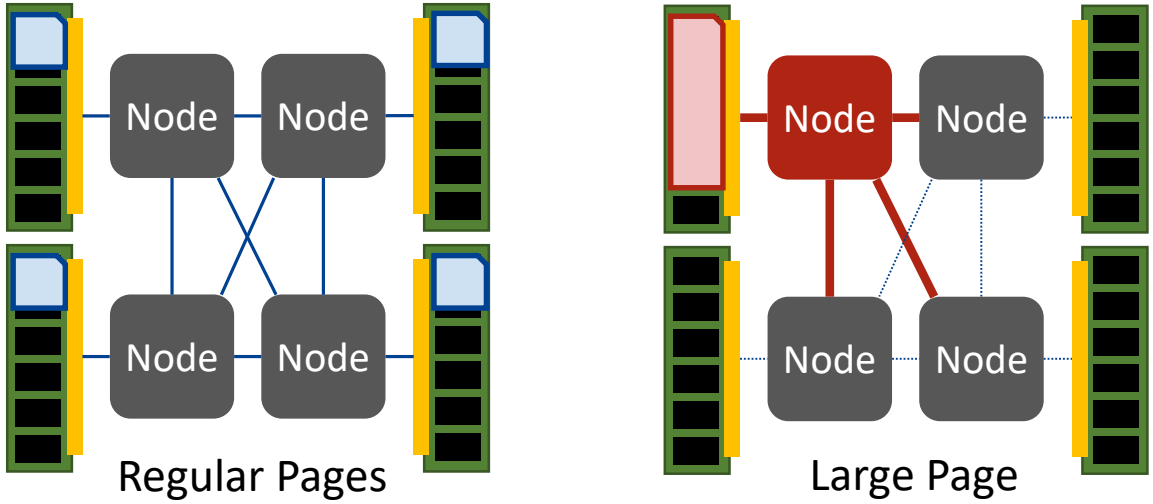


- Heterogeneous memory worsens non-uniformity [3][4]

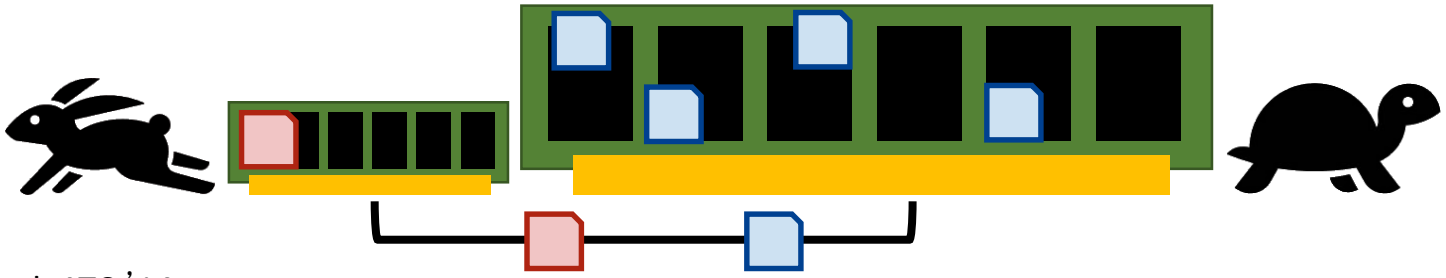
[2] Baptiste et al. ATC '14
[3] Lee et al. ISCA '15
[4] Agarwal et al. ASPLOS '17

Large Contiguity vs. Memory Non-Uniformity

- Conflicting goals of NUMA systems and large pages[2]
  - Memory traffic balance vs. efficient address translation



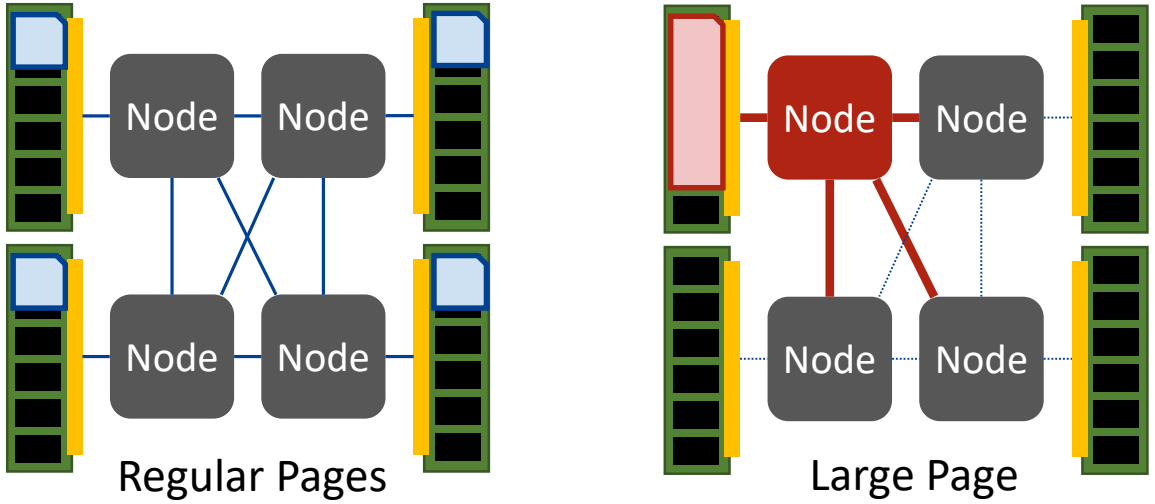
- Heterogeneous memory worsens non-uniformity [3][4]



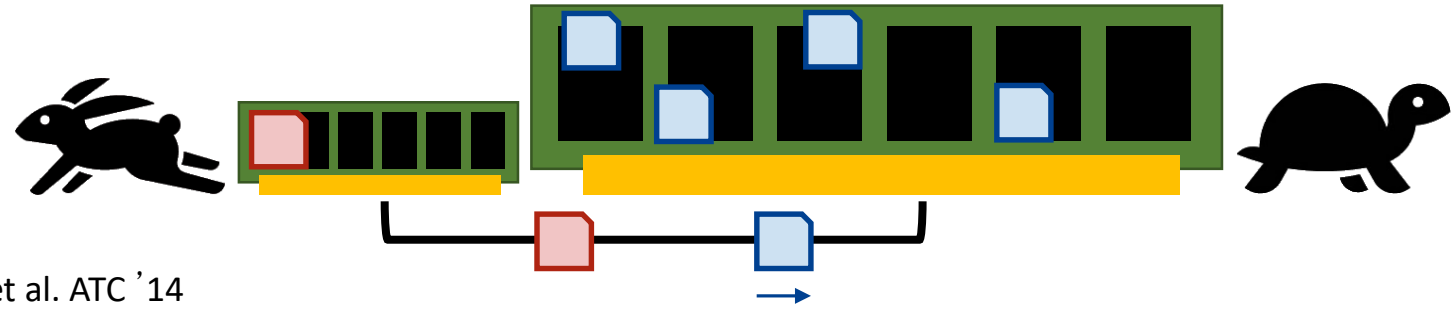
[2] Baptiste et al. ATC '14
[3] Lee et al. ISCA '15
[4] Agarwal et al. ASPLOS '17

Large Contiguity vs. Memory Non-Uniformity

- Conflicting goals of NUMA systems and large pages[2]
  - Memory traffic balance vs. efficient address translation



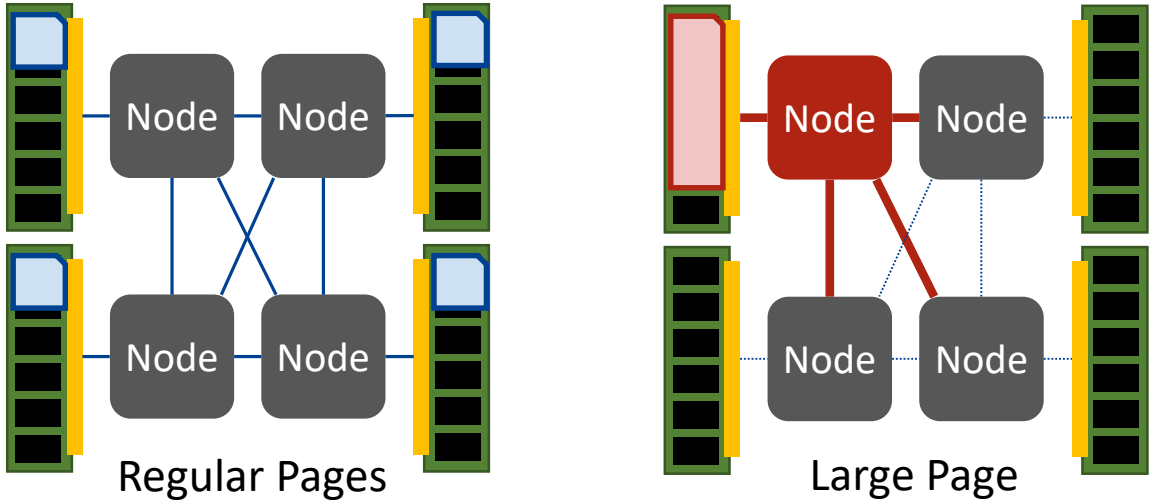
- Heterogeneous memory worsens non-uniformity [3][4]



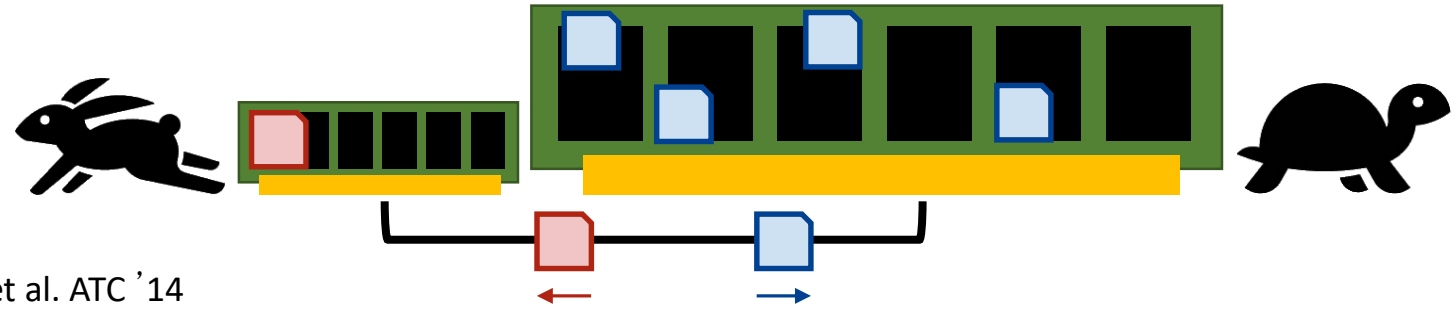
[2] Baptiste et al. ATC ’14
[3] Lee et al. ISCA ‘15
[4] Agarwal et al. ASPLOS ‘17

Large Contiguity vs. Memory Non-Uniformity

- Conflicting goals of NUMA systems and large pages[2]
  - Memory traffic balance vs. efficient address translation



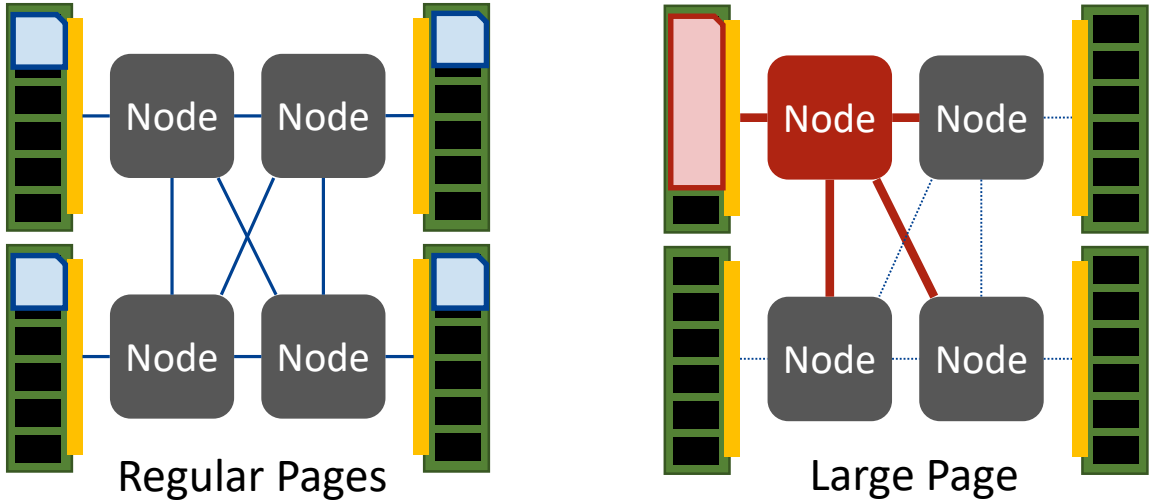
- Heterogeneous memory worsens non-uniformity [3][4]



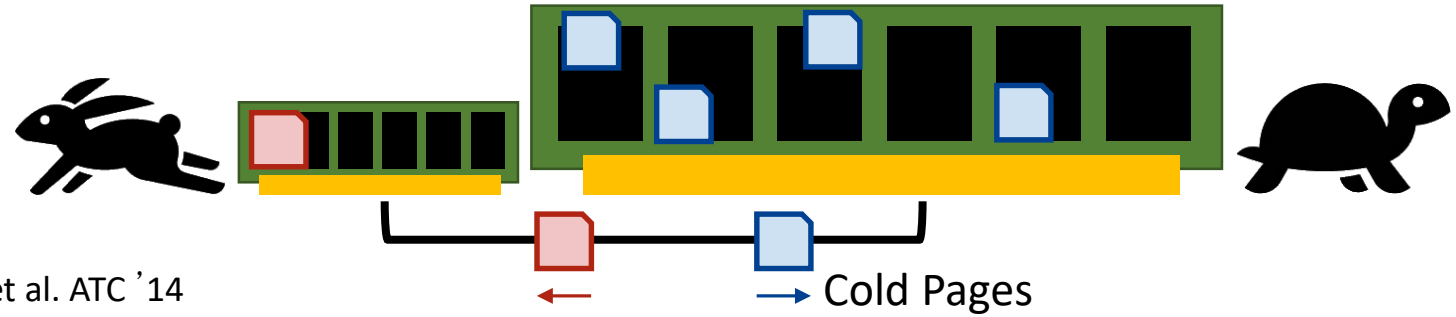
[2] Baptiste et al. ATC ’14
[3] Lee et al. ISCA ‘15
[4] Agarwal et al. ASPLOS ‘17

Large Contiguity vs. Memory Non-Uniformity

- Conflicting goals of NUMA systems and large pages[2]
  - Memory traffic balance vs. efficient address translation



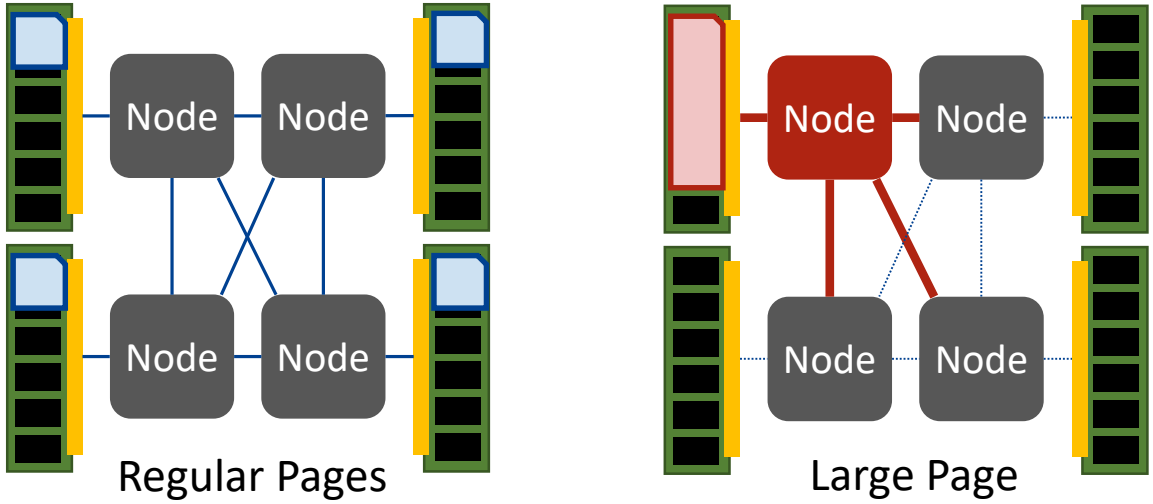
- Heterogeneous memory worsens non-uniformity [3][4]



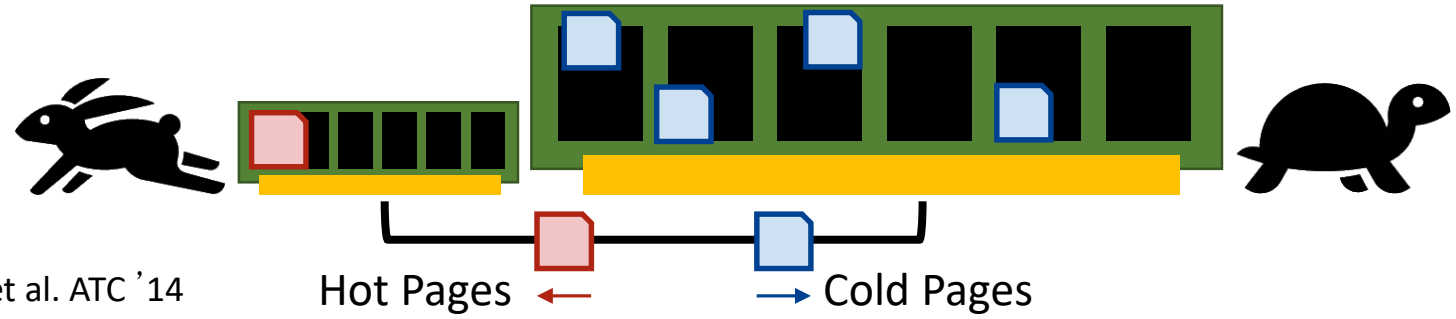
[2] Baptiste et al. ATC ’14
[3] Lee et al. ISCA ‘15
[4] Agarwal et al. ASPLOS ‘17

Large Contiguity vs. Memory Non-Uniformity

- Conflicting goals of NUMA systems and large pages[2]
  - Memory traffic balance vs. efficient address translation



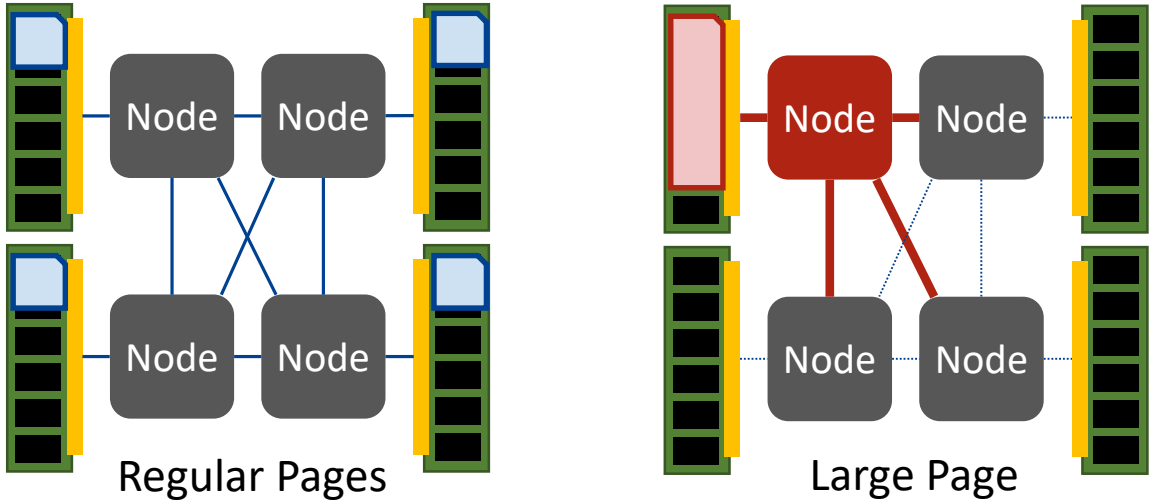
- Heterogeneous memory worsens non-uniformity [3][4]



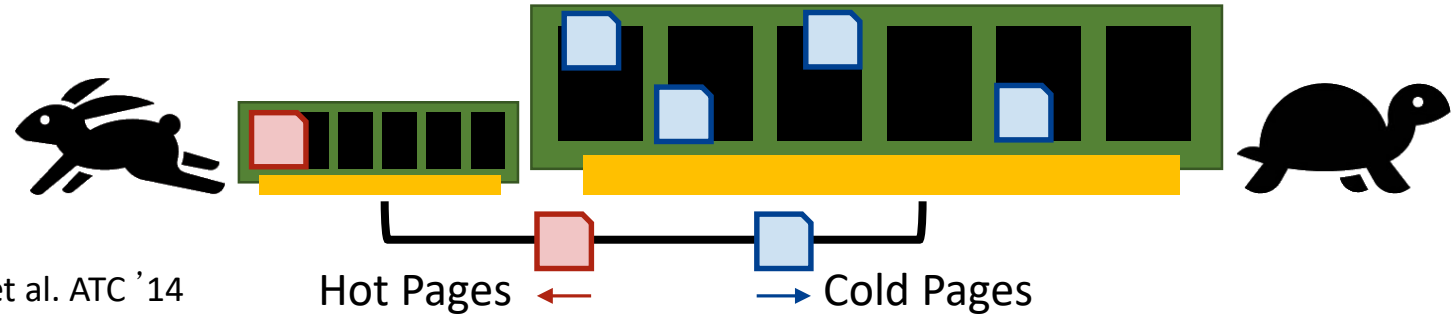
[2] Baptiste et al. ATC '14
[3] Lee et al. ISCA '15
[4] Agarwal et al. ASPLOS '17

Large Contiguity vs. Memory Non-Uniformity

- Conflicting goals of NUMA systems and large pages[2]
  - Memory traffic balance vs. efficient address translation



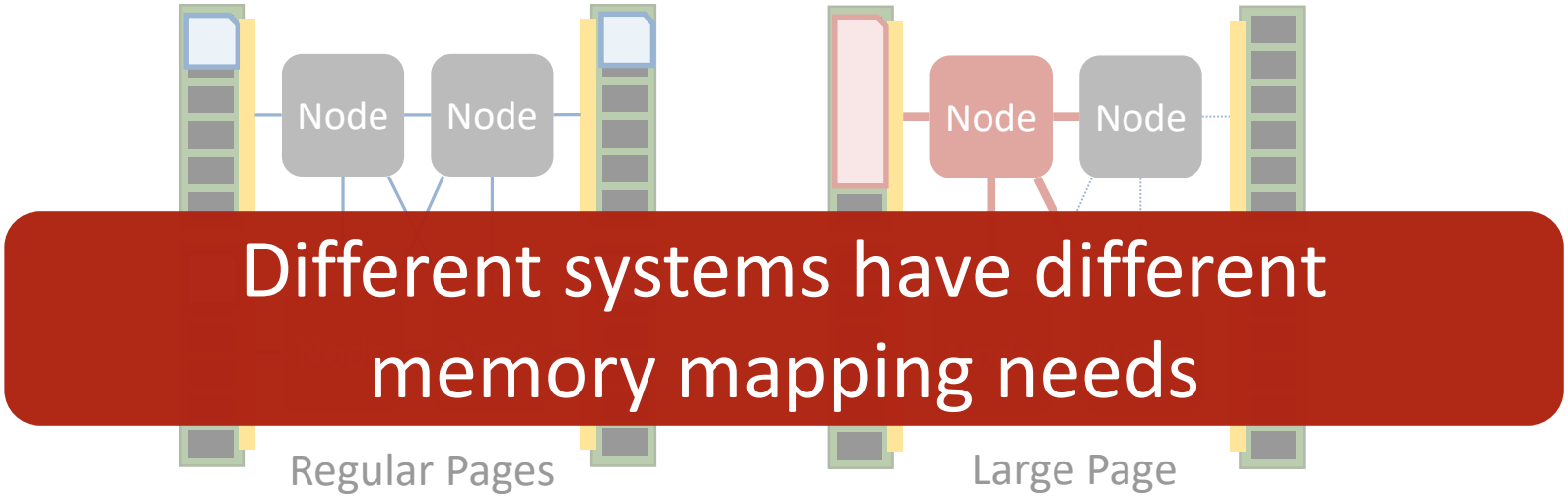
- Heterogeneous memory worsens non-uniformity [3][4]



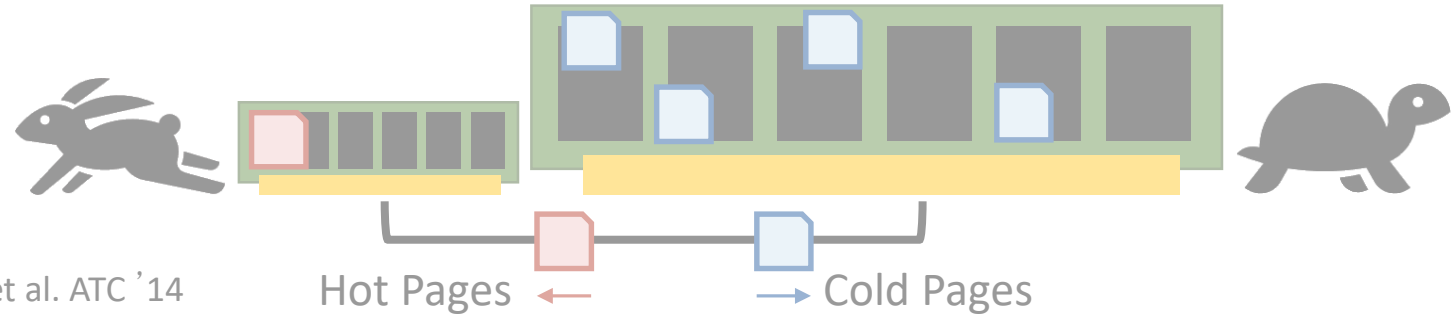
[2] Baptiste et al. ATC '14
[3] Lee et al. ISCA '15
[4] Agarwal et al. ASPLOS '17

Large Contiguity vs. Memory Non-Uniformity

- Conflicting goals of NUMA systems and large pages[2]
  - Memory traffic balance vs. efficient address translation



- Heterogeneous memory worsens non-uniformity [3][4]



[2] Baptiste et al. ATC '14
[3] Lee et al. ISCA '15
[4] Agarwal et al. ASPLOS '17

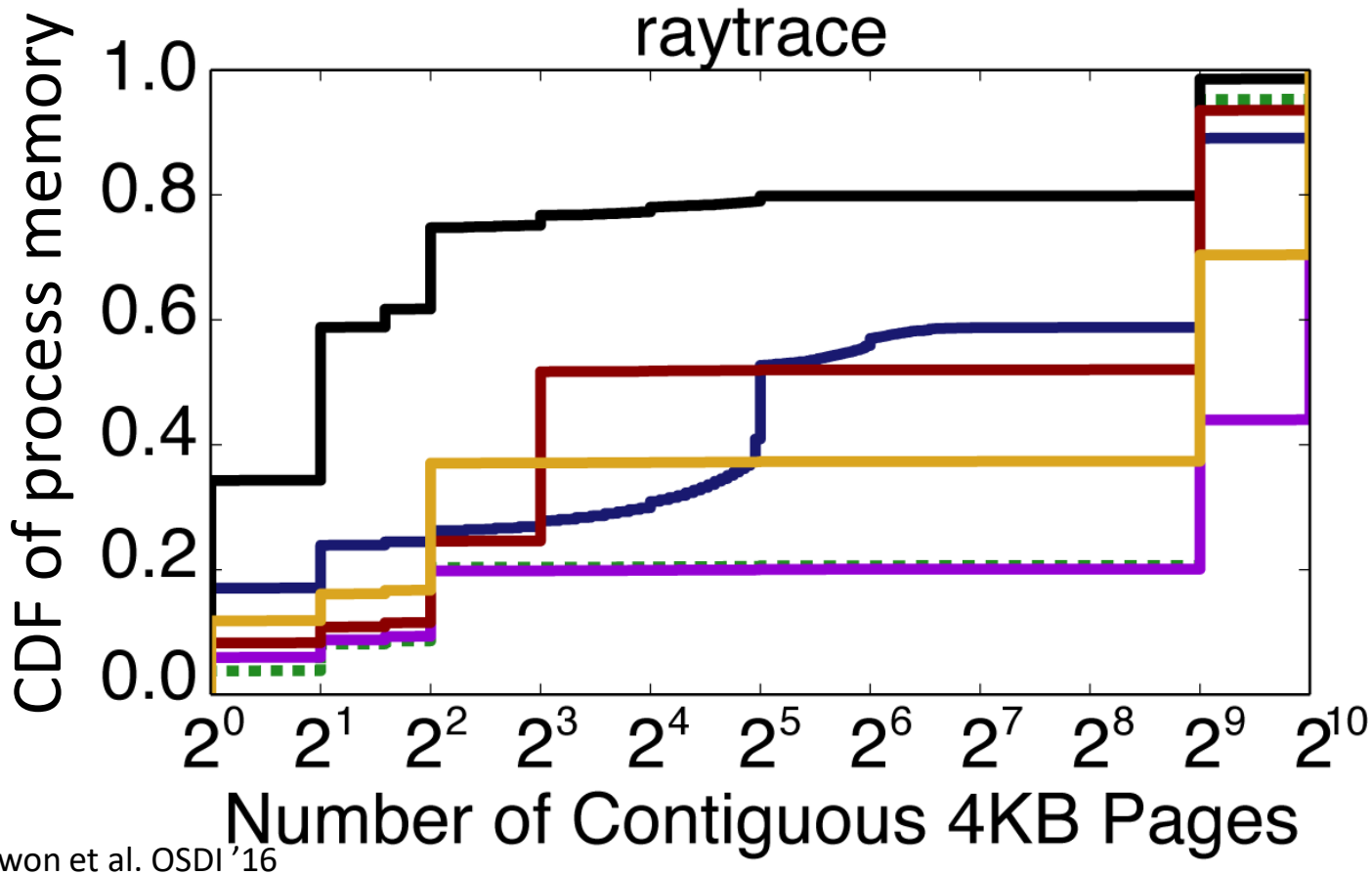
Need for an All-Rounder Solution

- Contiguity distribution varies among workloads
- Also varies within the same workload[7]

[7] Kwon et al. OSDI ’16

Need for an All-Rounder Solution

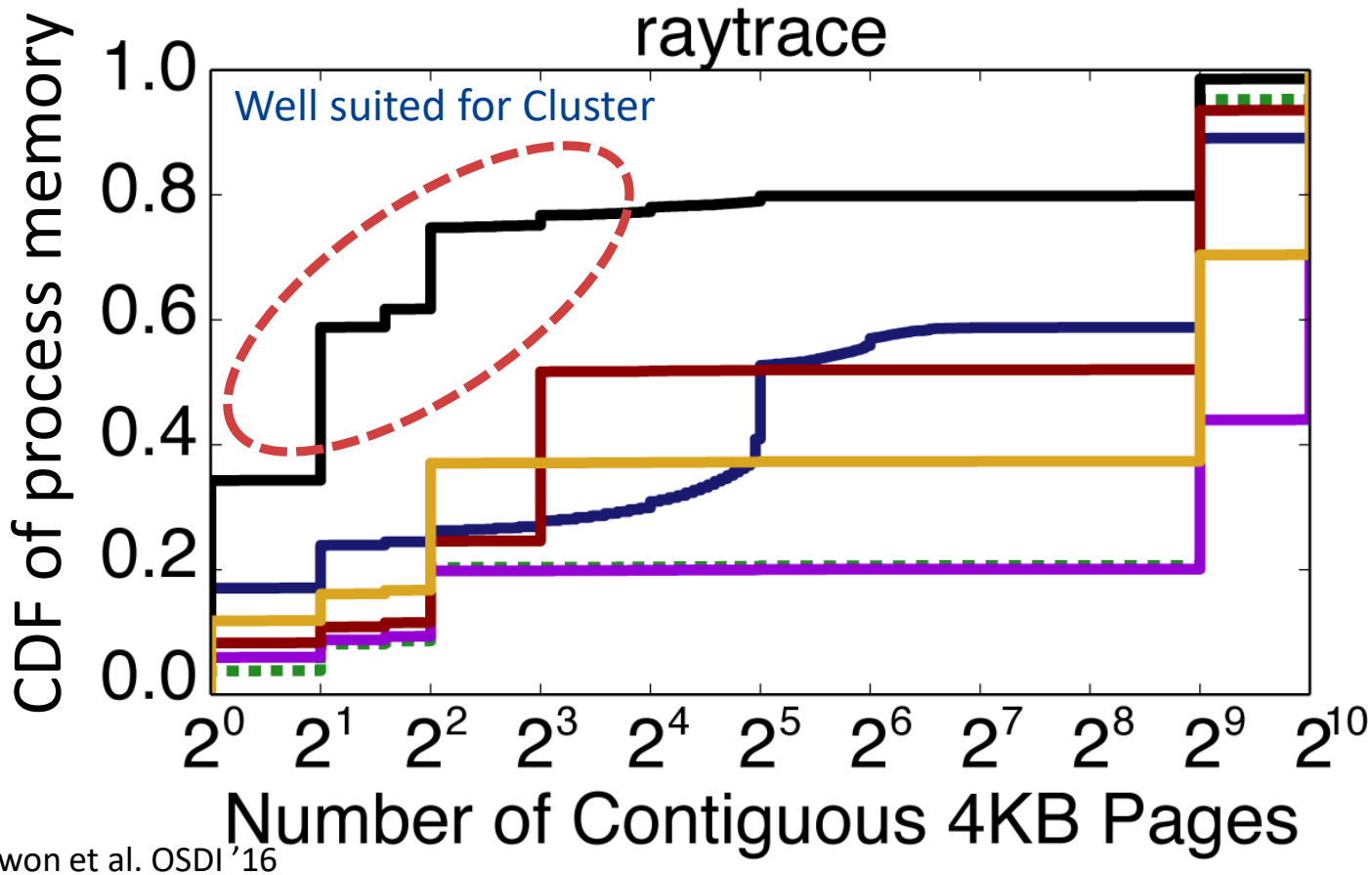
- Contiguity distribution varies among workloads
- Also varies within the same workload[7]



[7] Kwon et al. OSDI '16

Need for an All-Rounder Solution

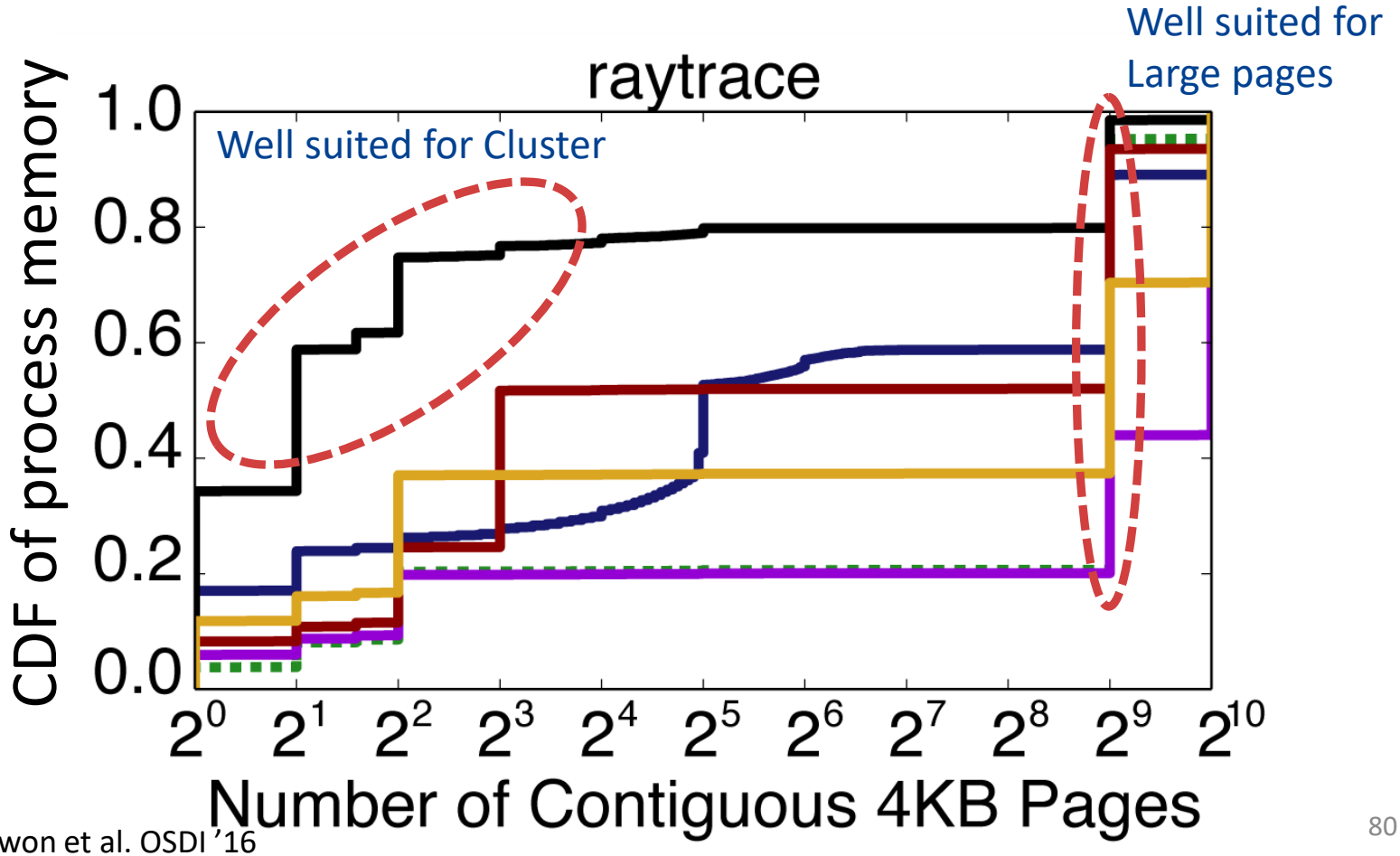
- Contiguity distribution varies among workloads
- Also varies within the same workload[7]



[7] Kwon et al. OSDI '16

Need for an All-Rounder Solution

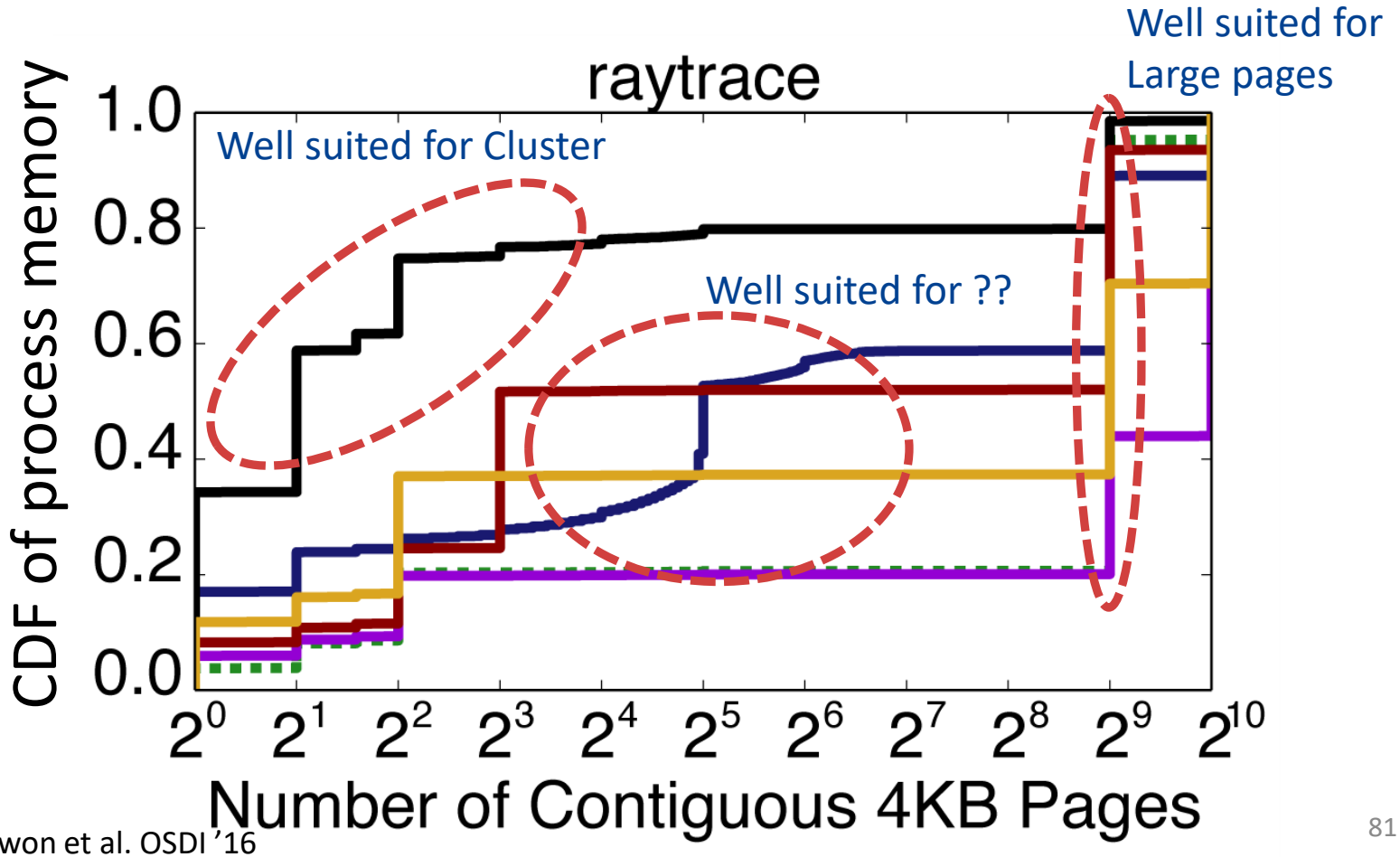
- Contiguity distribution varies among workloads
- Also varies within the same workload[7]



[7] Kwon et al. OSDI '16

Need for an All-Rounder Solution

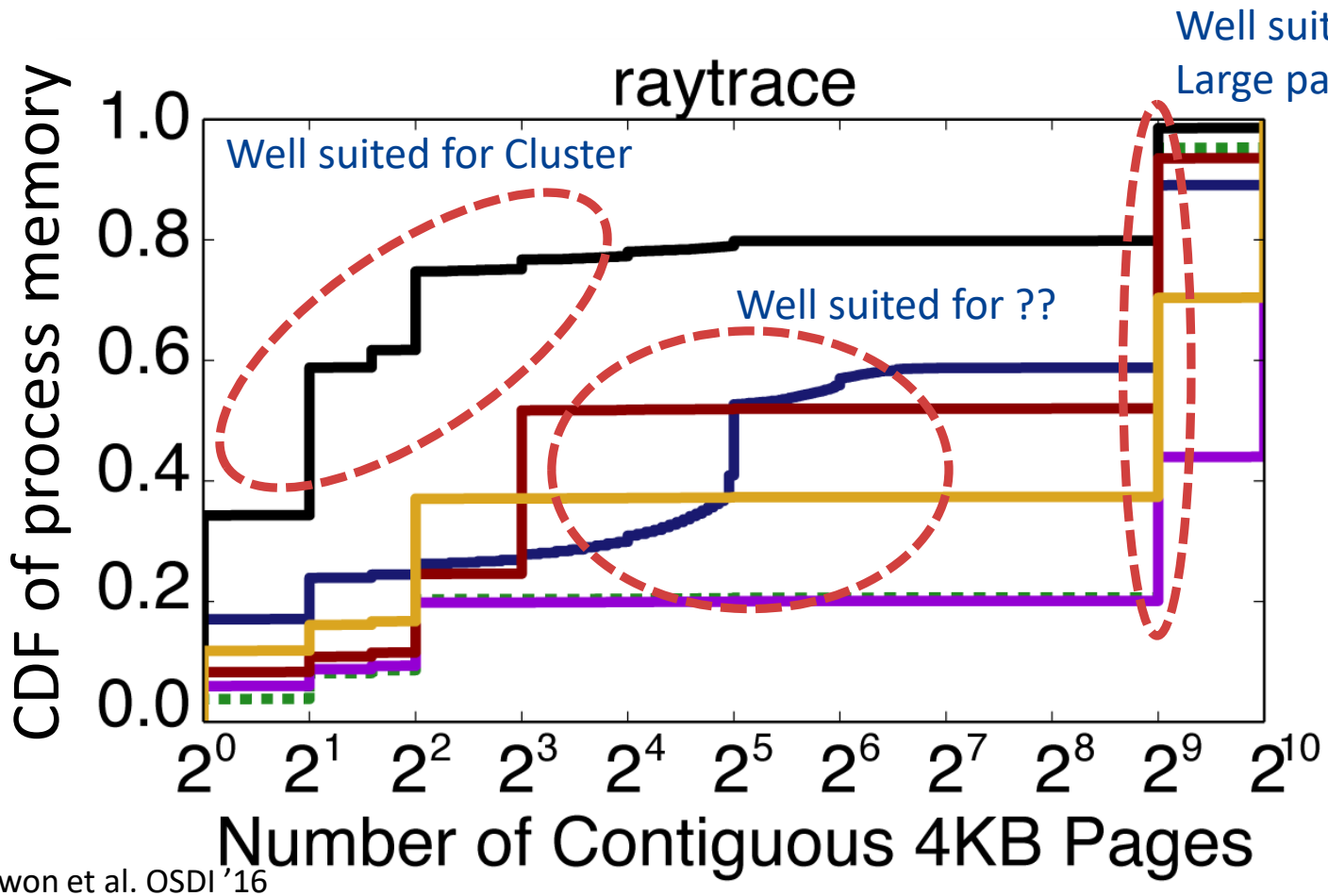
- Contiguity distribution varies among workloads
- Also varies within the same workload[7]



[7] Kwon et al. OSDI '16

Need for an All-Rounder Solution

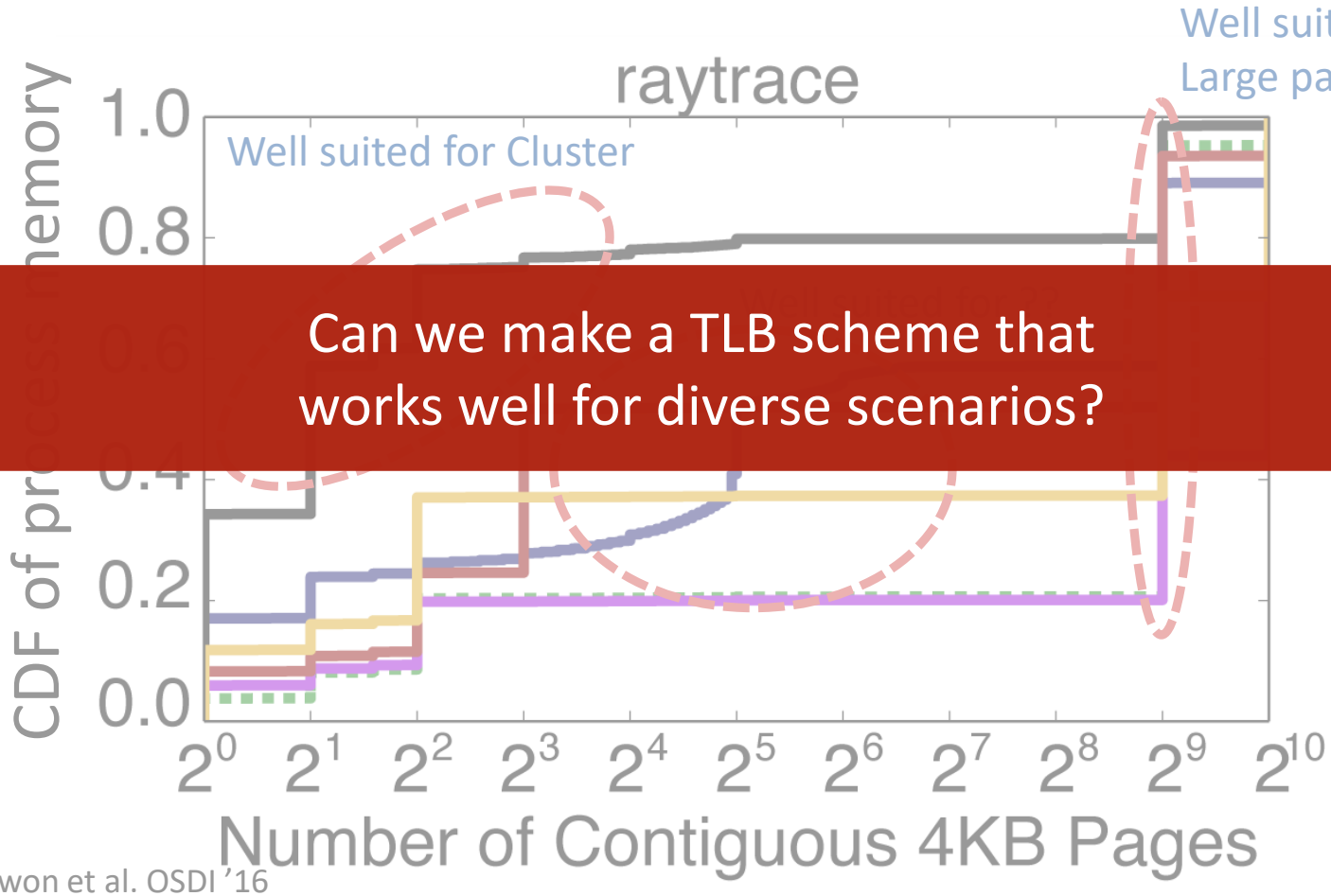
- Contiguity distribution varies among workloads
- Also varies within the same workload[7]



[7] Kwon et al. OSDI '16

Need for an All-Rounder Solution

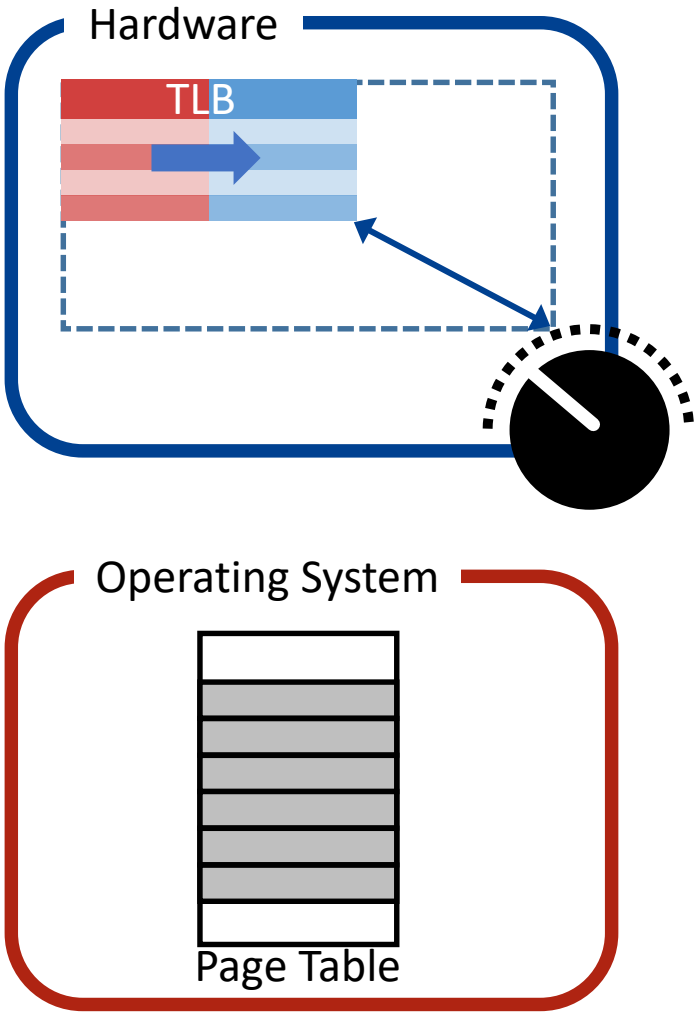
- Contiguity distribution varies among workloads
- Also varies within the same workload[7]



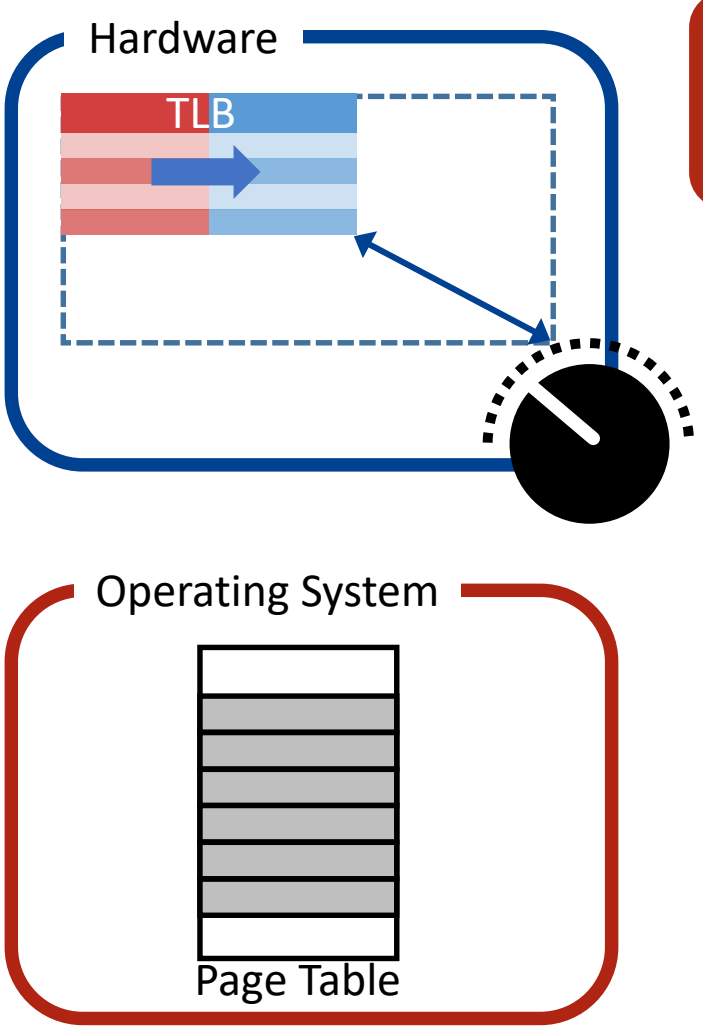
Can we make a TLB scheme that works well for diverse scenarios?

[7] Kwon et al. OSDI '16

Hybrid TLB Coalescing



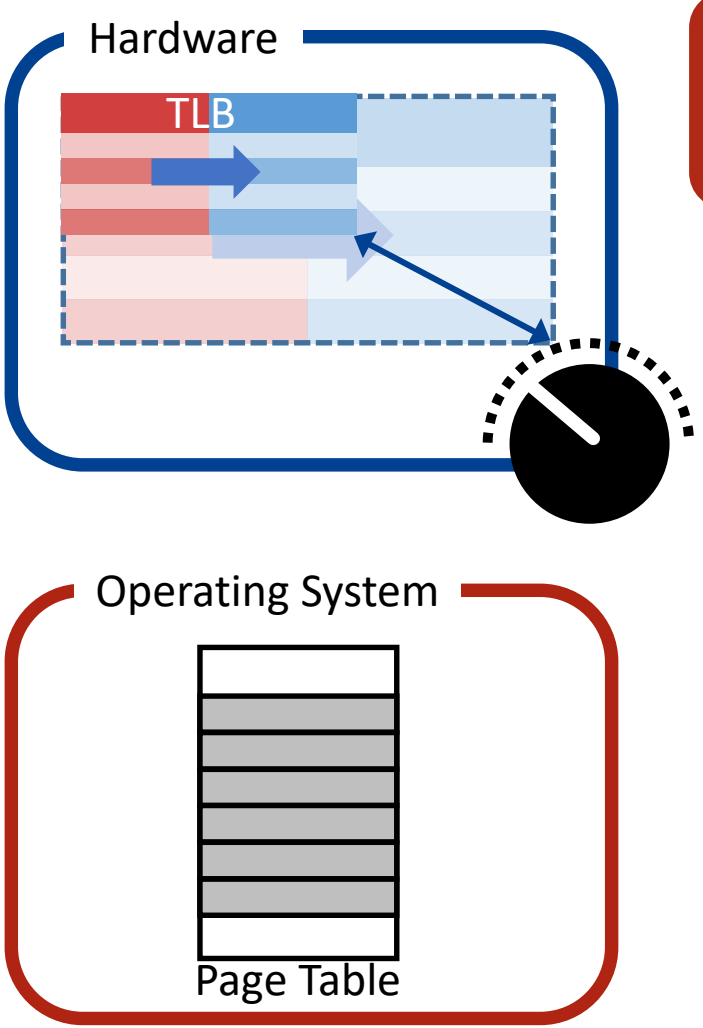
Hybrid TLB Coalescing



We propose a TLB with adjustable coverage

- **HW**-**SW** Joint Effort
- **HW** offers adjustable TLB coverage
  - Number of TLB entries fixed
  - Coverage of entry adjustable
- **OS** decides best TLB coverage
  - Adjusts TLB coverage per process
- **OS** identifies contiguous chunks
  - Marks onto process page table

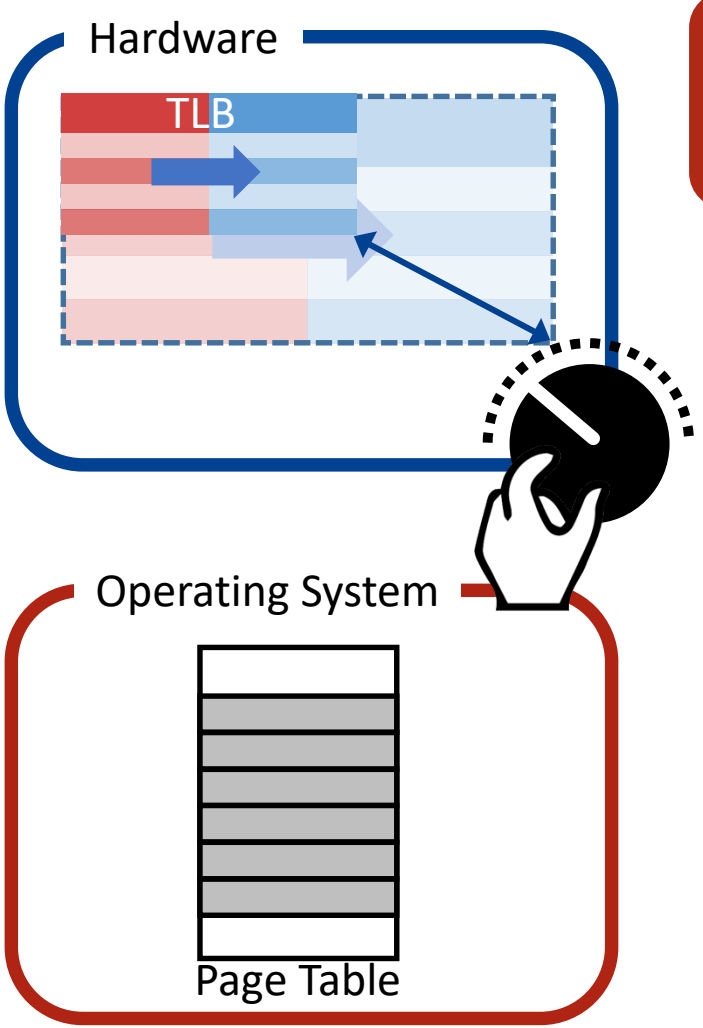
Hybrid TLB Coalescing



We propose a TLB with adjustable coverage

- **HW**-**SW** Joint Effort
- **HW** offers adjustable TLB coverage
  - Number of TLB entries fixed
  - Coverage of entry adjustable
- **OS** decides best TLB coverage
  - Adjusts TLB coverage per process
- **OS** identifies contiguous chunks
  - Marks onto process page table

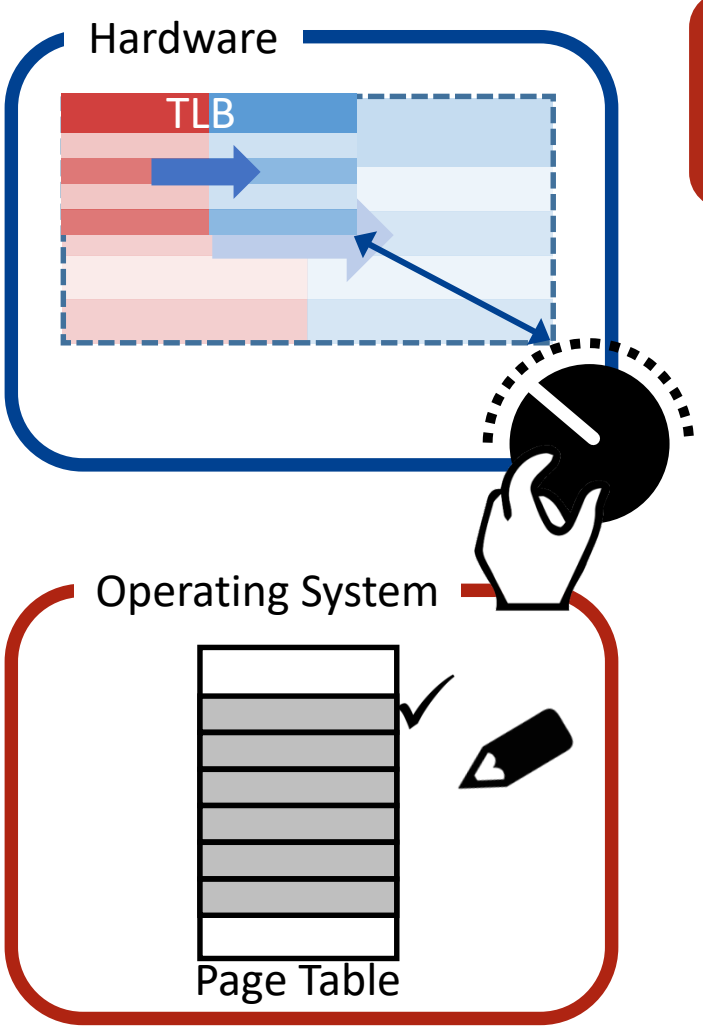
Hybrid TLB Coalescing



We propose a TLB with adjustable coverage

- **HW**-**SW** Joint Effort
- **HW** offers adjustable TLB coverage
  - Number of TLB entries fixed
  - Coverage of entry adjustable
- **OS** decides best TLB coverage
  - Adjusts TLB coverage per process
- **OS** identifies contiguous chunks
  - Marks onto process page table

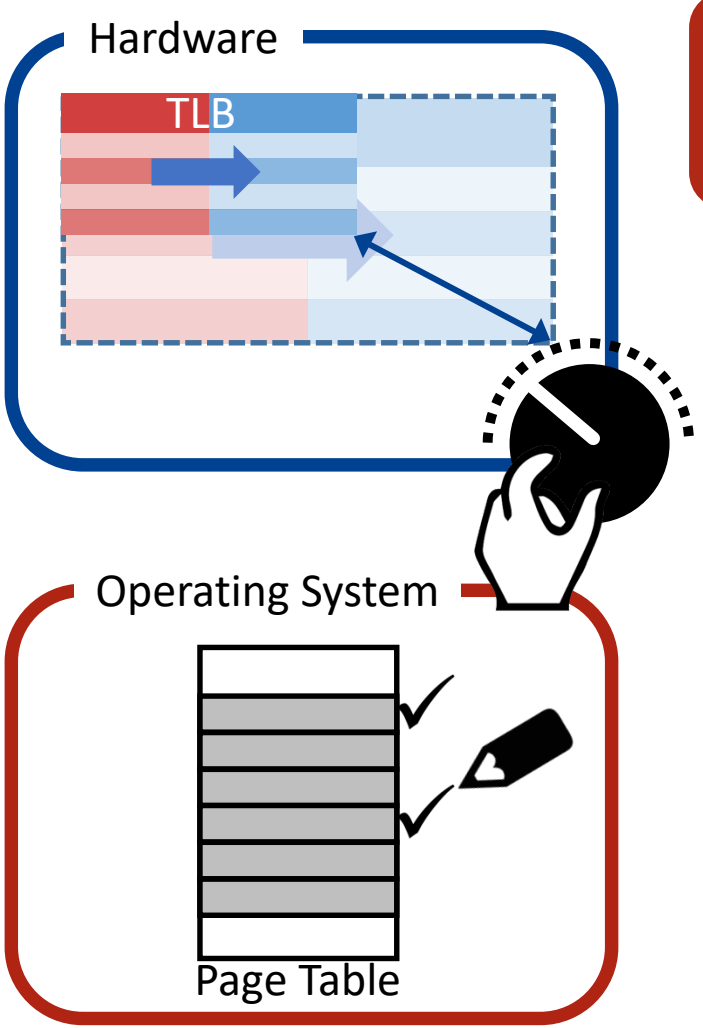
Hybrid TLB Coalescing



We propose a TLB with adjustable coverage

- **HW**-**SW** Joint Effort
- **HW** offers adjustable TLB coverage
  - Number of TLB entries fixed
  - Coverage of entry adjustable
- **OS** decides best TLB coverage
  - Adjusts TLB coverage per process
- **OS** identifies contiguous chunks
  - Marks onto process page table

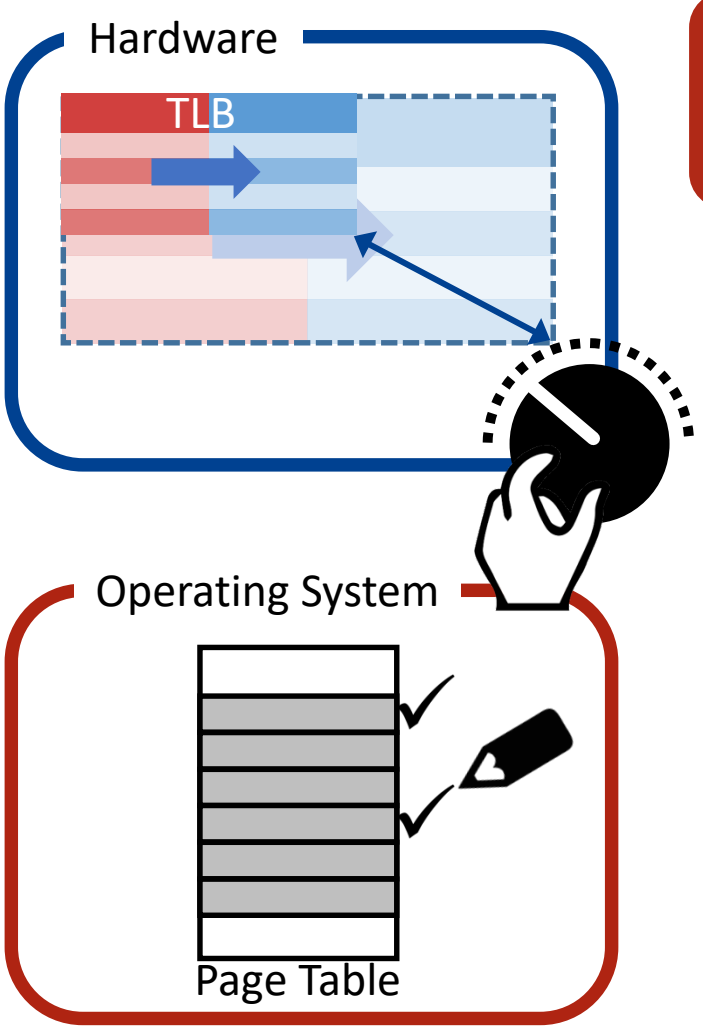
Hybrid TLB Coalescing



We propose a TLB with adjustable coverage

- **HW**-**SW** Joint Effort
- **HW** offers adjustable TLB coverage
  - Number of TLB entries fixed
  - Coverage of entry adjustable
- **OS** decides best TLB coverage
  - Adjusts TLB coverage per process
- **OS** identifies contiguous chunks
  - Marks onto process page table

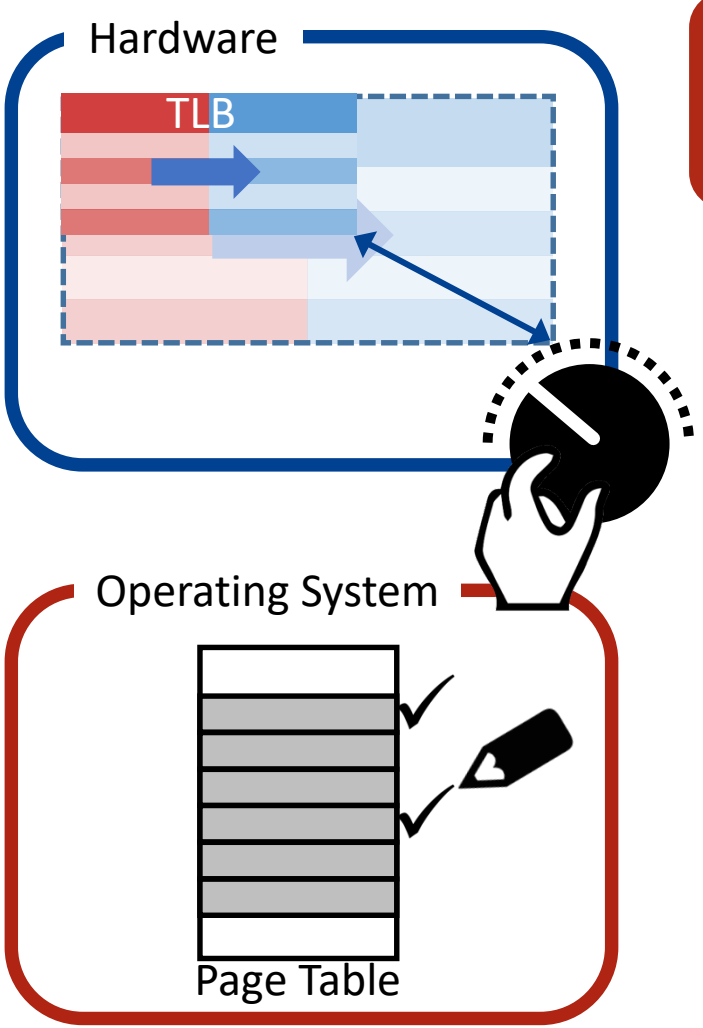
Hybrid TLB Coalescing



We propose a TLB with adjustable coverage

- **HW**-**SW** Joint Effort
- **HW** offers adjustable TLB coverage
  - Number of TLB entries fixed
  - Coverage of entry adjustable
- **OS** decides best TLB coverage
  - Adjusts TLB coverage per process
- **OS** identifies contiguous chunks
  - Marks onto process page table

Hybrid TLB Coalescing

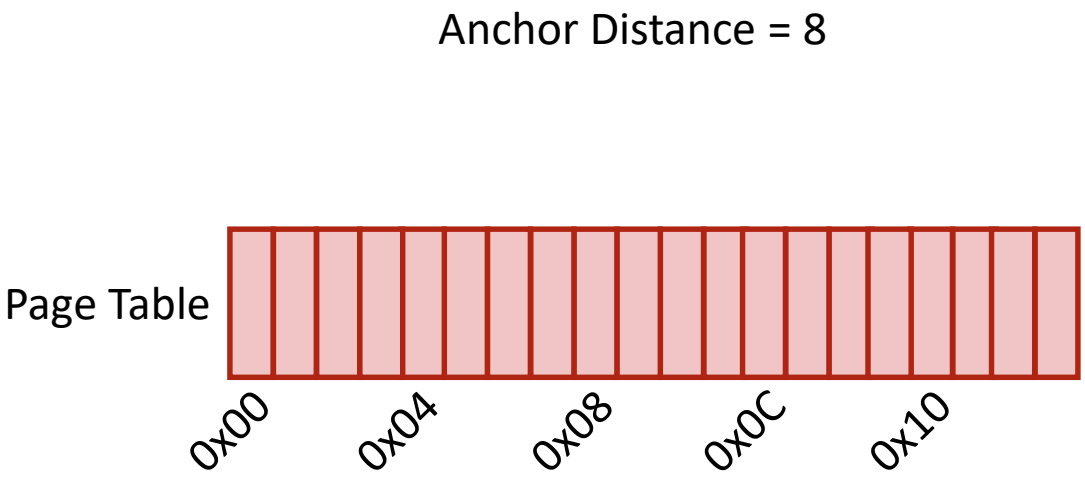


We propose a TLB with adjustable coverage

- **HW**-**SW** Joint Effort
- **HW** offers adjustable TLB coverage
  - Number of TLB entries fixed
  - Coverage of entry adjustable
- **OS** decides best TLB coverage
  - Adjusts TLB coverage per process
- **OS** identifies contiguous chunks
  - Marks onto process page table

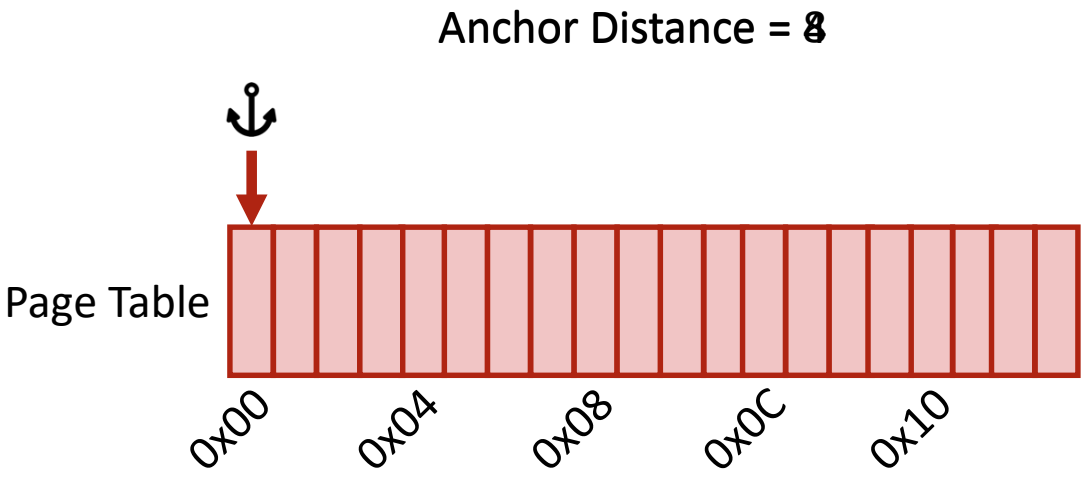
Anchor

- Anchors are special entries in the page table
  - Placed at every alignments of anchor distance
  - Anchor distance is a power of 2 (for encoding efficiency)
  - Anchor distance configurable by OS



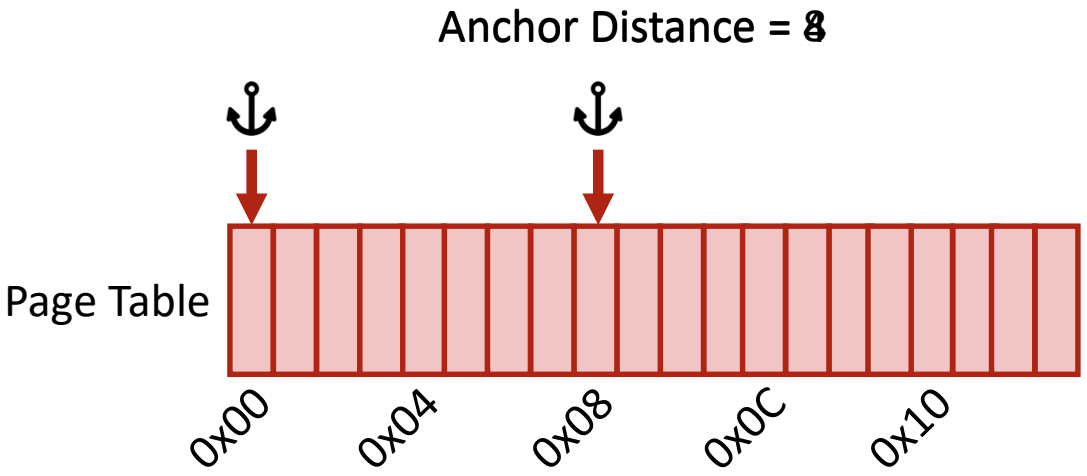
Anchor

- Anchors are special entries in the page table
  - Placed at every alignments of anchor distance
  - Anchor distance is a power of 2 (for encoding efficiency)
  - Anchor distance configurable by OS



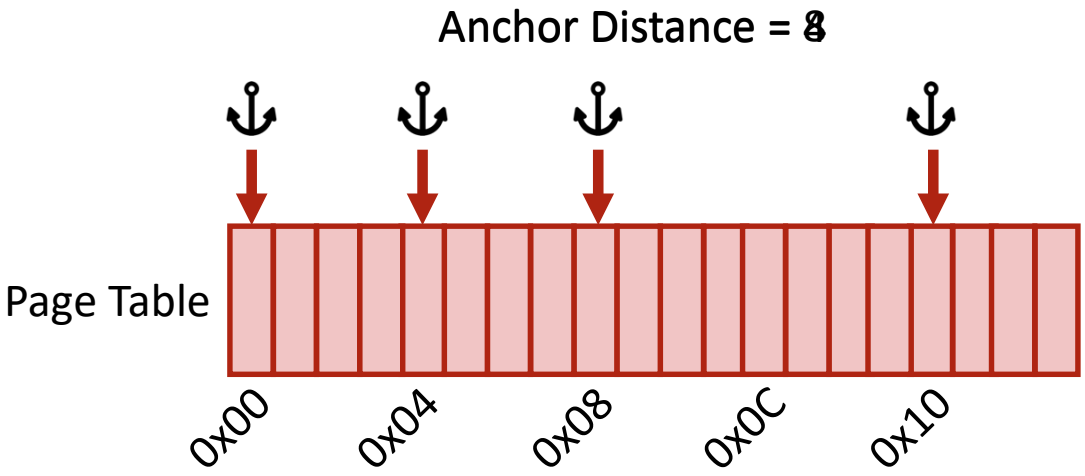
Anchor

- Anchors are special entries in the page table
  - Placed at every alignments of anchor distance
  - Anchor distance is a power of 2 (for encoding efficiency)
  - Anchor distance configurable by OS



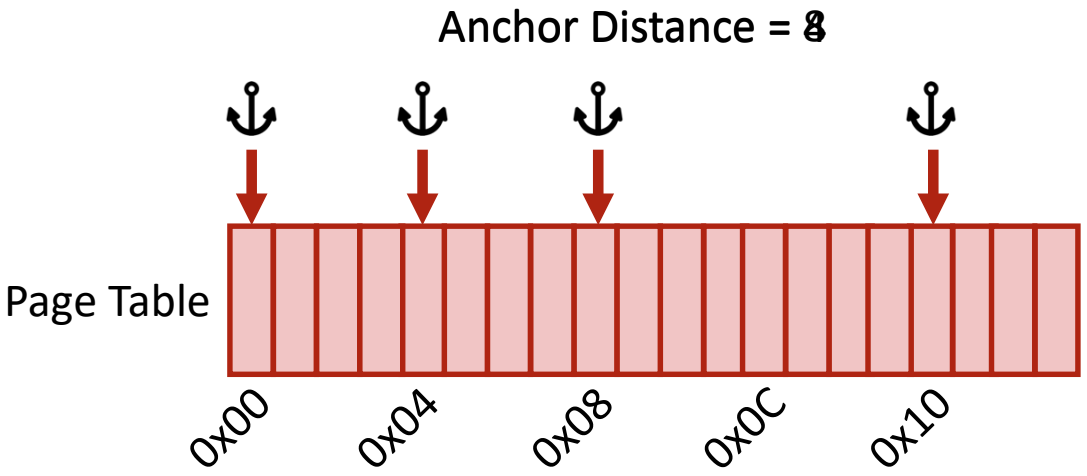
Anchor

- Anchors are special entries in the page table
  - Placed at every alignments of anchor distance
  - Anchor distance is a power of 2 (for encoding efficiency)
  - Anchor distance configurable by OS



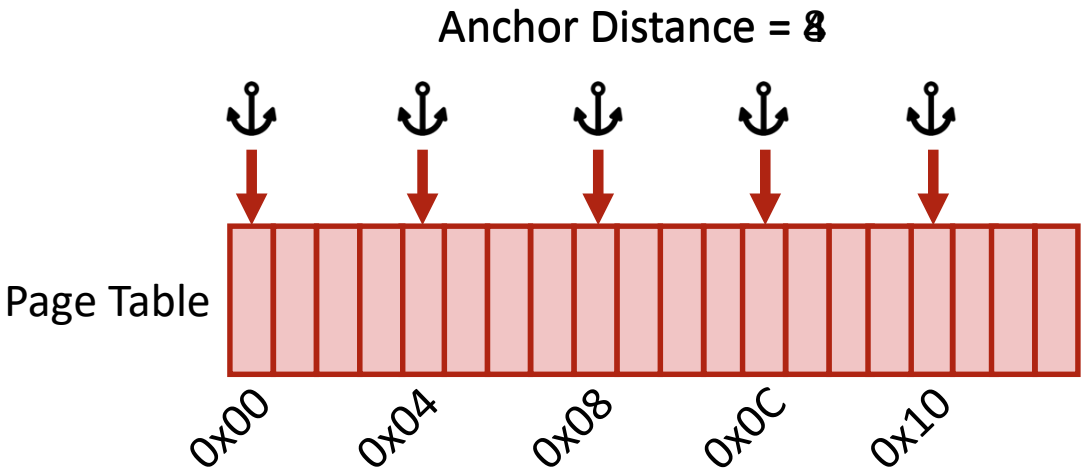
Anchor

- Anchors are special entries in the page table
  - Placed at every alignments of anchor distance
  - Anchor distance is a power of 2 (for encoding efficiency)
  - Anchor distance configurable by OS



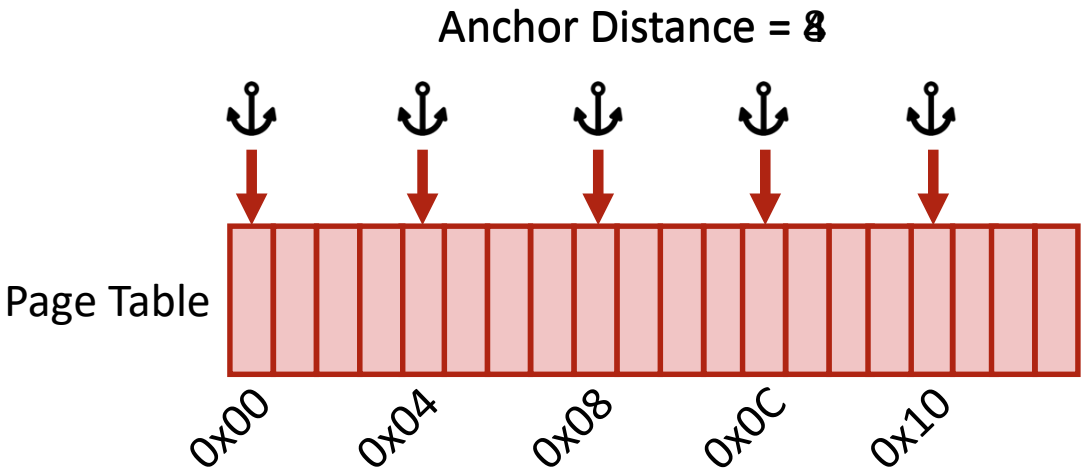
Anchor

- Anchors are special entries in the page table
  - Placed at every alignments of anchor distance
  - Anchor distance is a power of 2 (for encoding efficiency)
  - Anchor distance configurable by OS



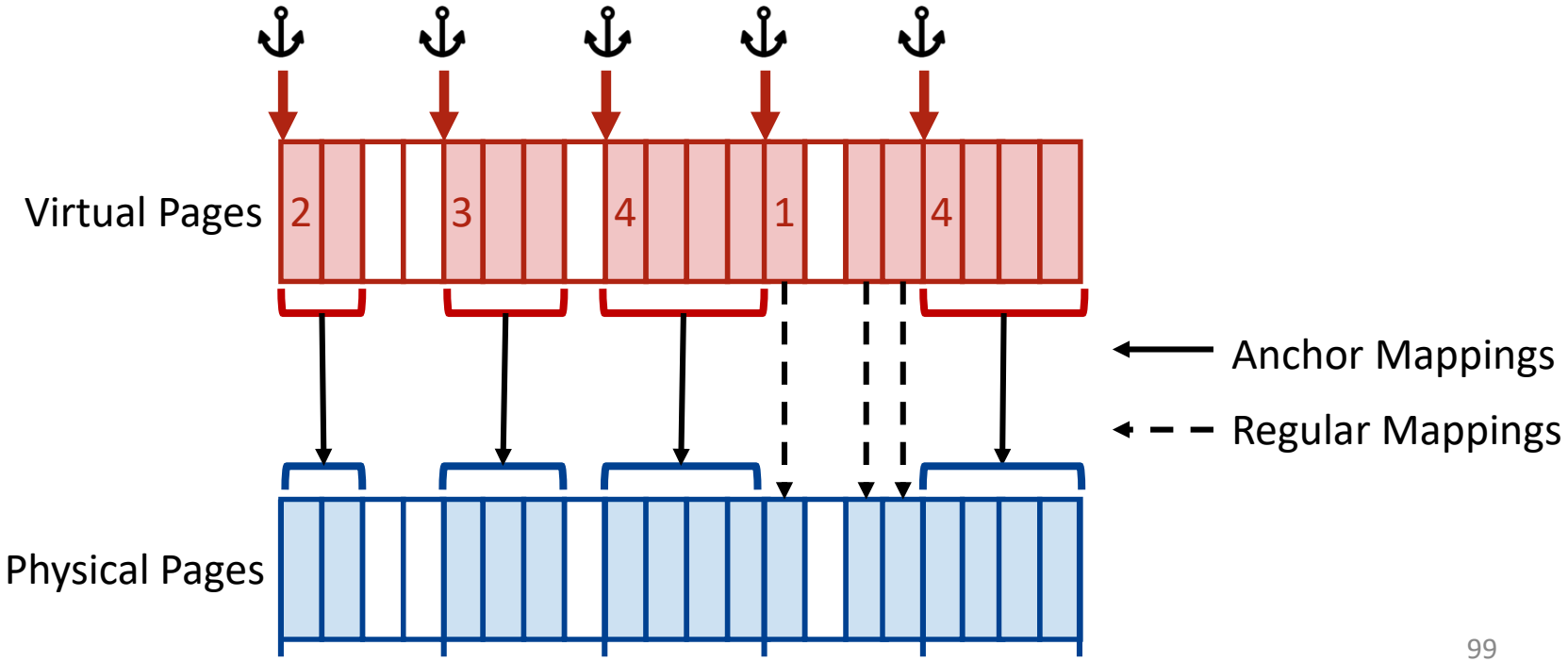
Anchor

- Anchors are special entries in the page table
  - Placed at every alignments of anchor distance
  - Anchor distance is a power of 2 (for encoding efficiency)
  - Anchor distance configurable by OS



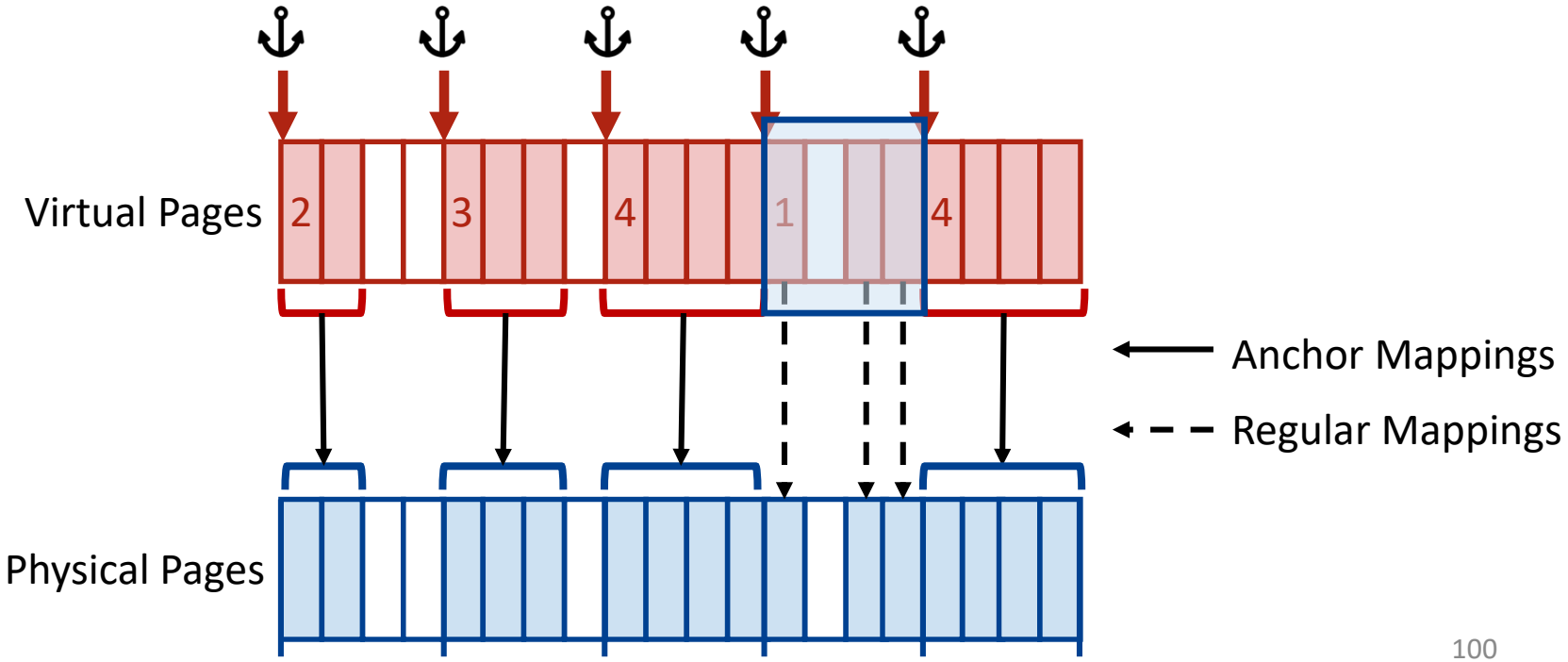
Anchor Page Table

- Uses the Page Table
- Anchor covers up to distance(4) contiguous pages
  - Each anchor represents contiguity that begins at anchor
- OS marks contiguity onto the anchor page table



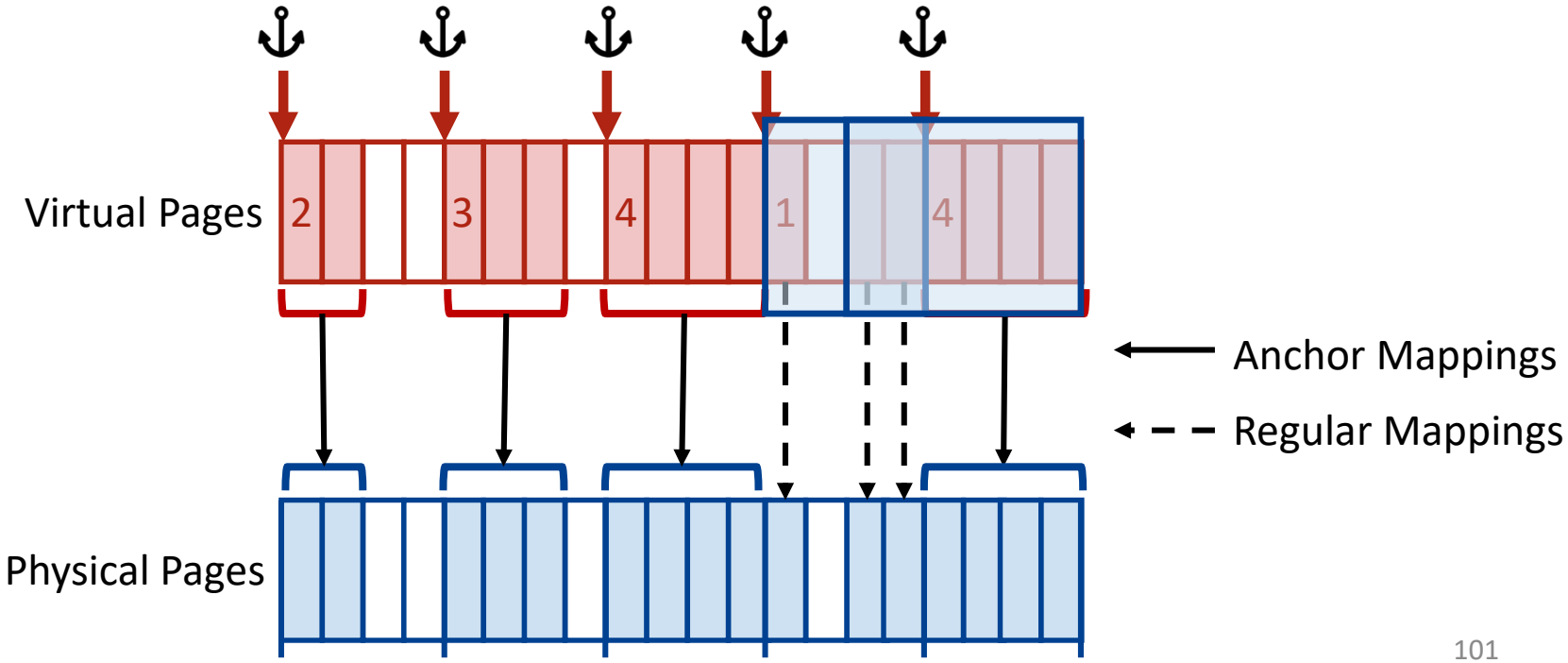
Anchor Page Table

- Uses the Page Table
- Anchor covers up to distance(4) contiguous pages
  - Each anchor represents contiguity that begins at anchor
- OS marks contiguity onto the anchor page table



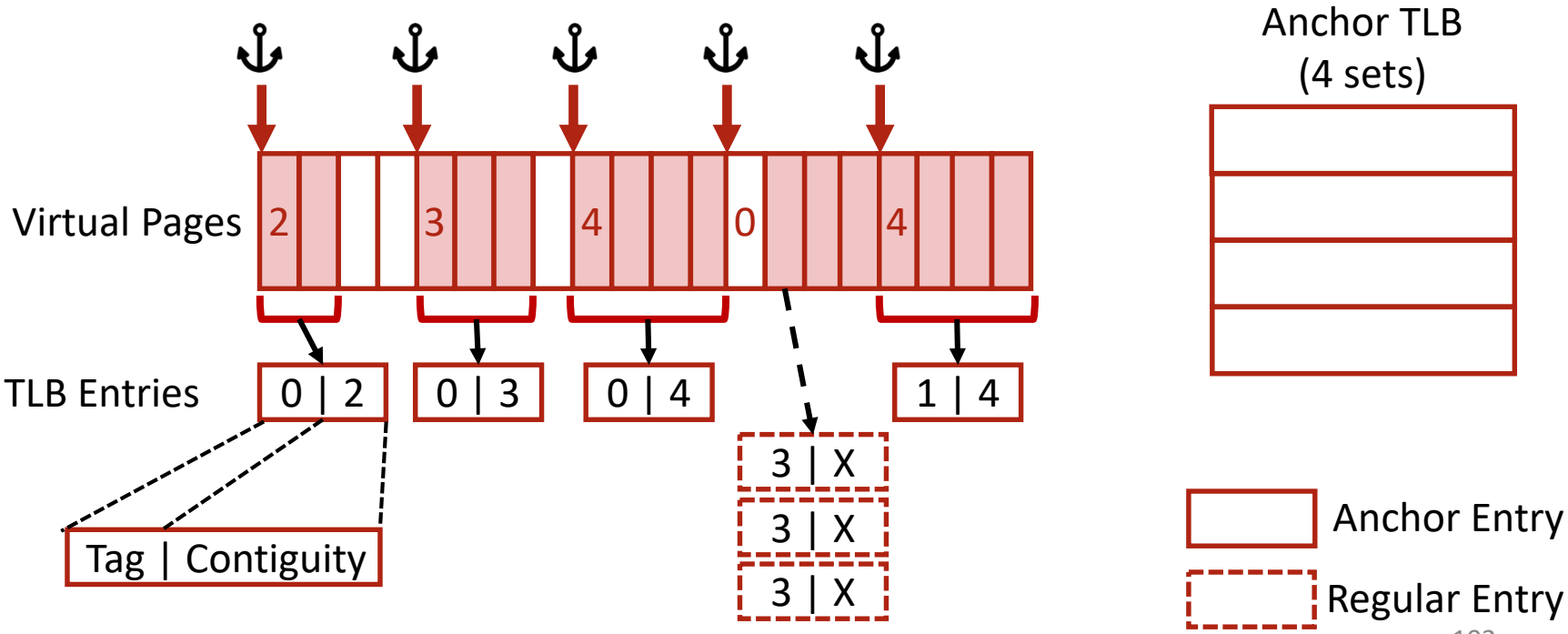
Anchor Page Table

- Uses the Page Table
- Anchor covers up to distance(4) contiguous pages
  - Each anchor represents contiguity that begins at anchor
- OS marks contiguity onto the anchor page table



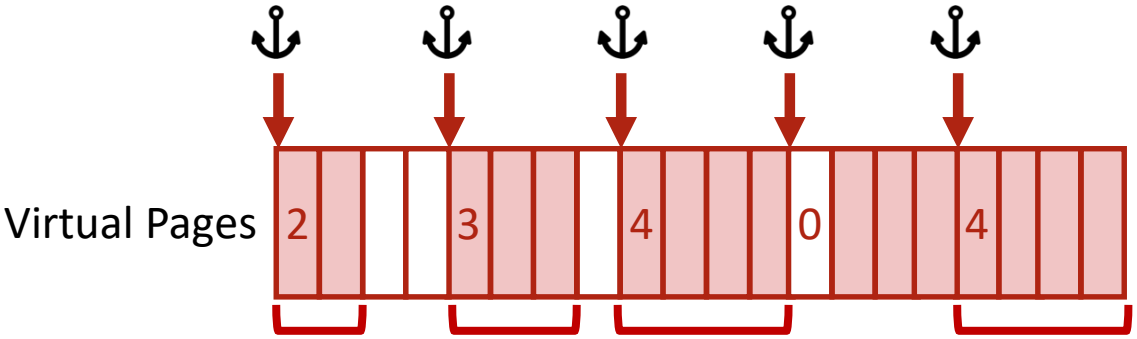
Anchor TLB

- Integrated into the L2 TLB
  - L1 keeps regular entries
- Caches both regular and anchor page table entries
  - Regular and anchor indexed differently



Anchor TLB Lookup

- On L1 TLB Miss Anchor TLB looks up
  - Regular TLB first
  - Anchor TLB next



Anchor TLB
(4 sets)

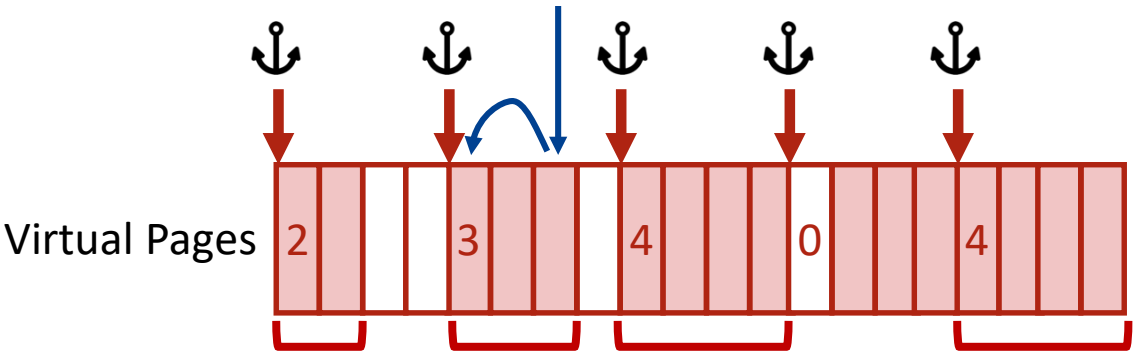
0 | 2	1 | 4
0 | 3	3 | X
0 | 4	3 | X
3 | X	

Anchor Entry

Regular Entry

Anchor TLB Lookup

- On L1 TLB Miss Anchor TLB looks up
  - Regular TLB first
  - Anchor TLB next



Anchor TLB
(4 sets)

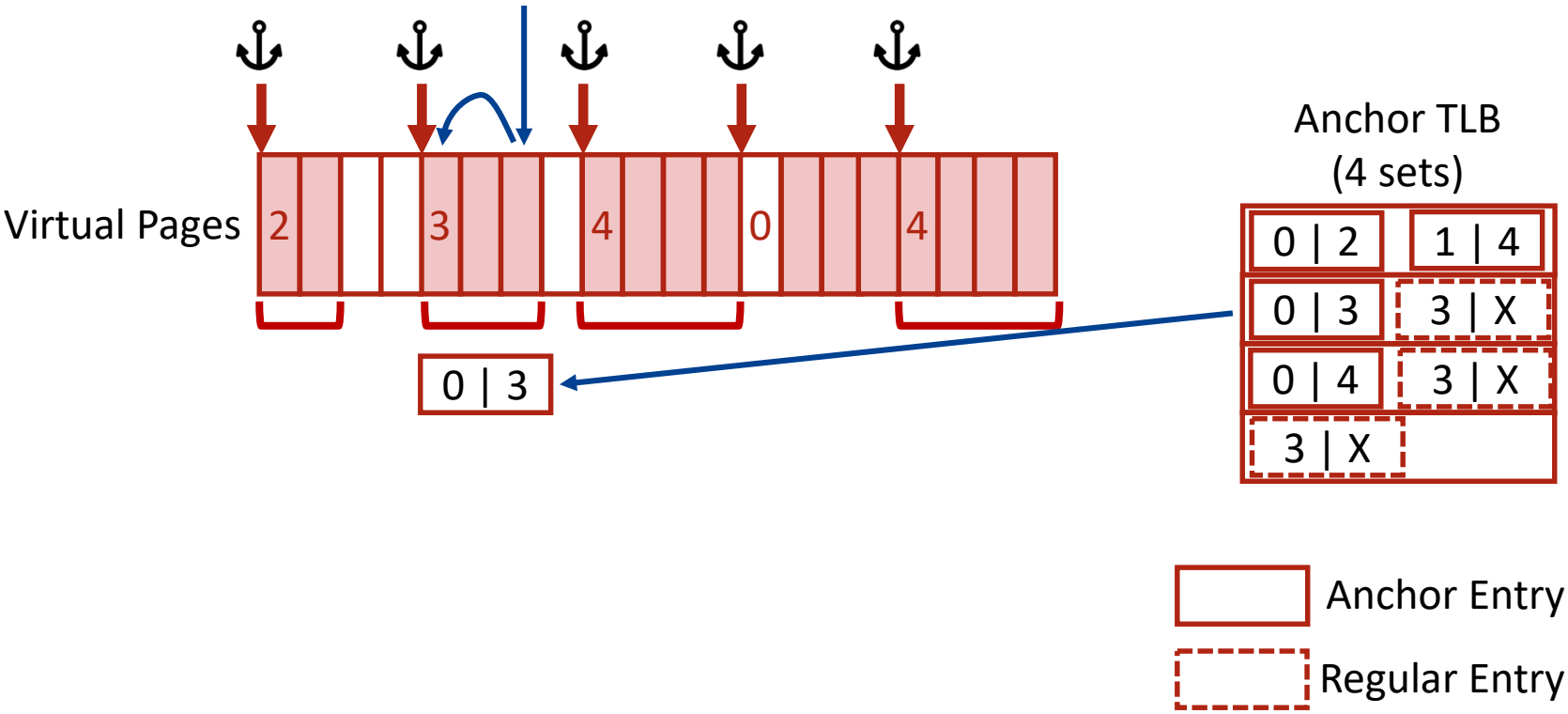
0 | 2	1 | 4
0 | 3	3 | X
0 | 4	3 | X
3 | X	

Anchor Entry

Regular Entry

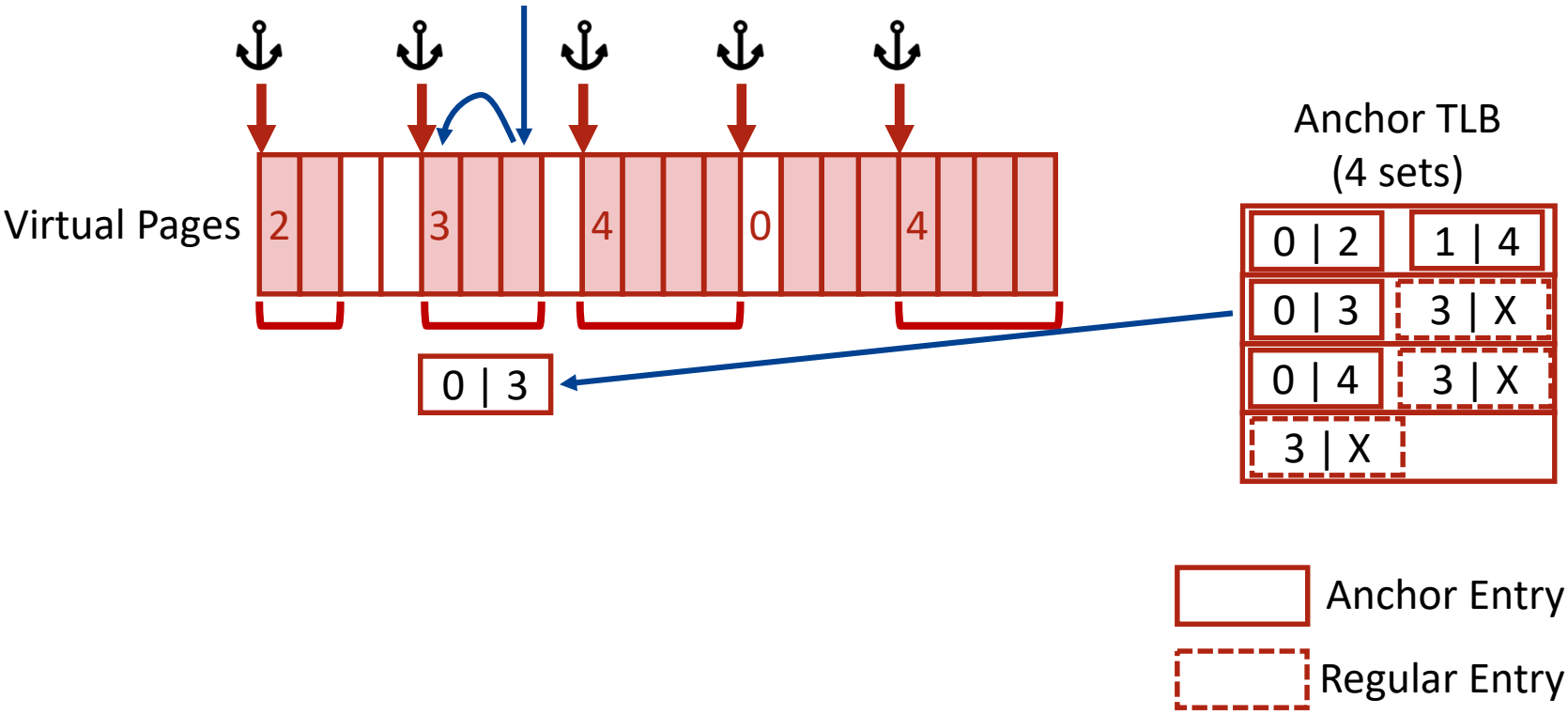
Anchor TLB Lookup

- On L1 TLB Miss Anchor TLB looks up
  - Regular TLB first
  - Anchor TLB next



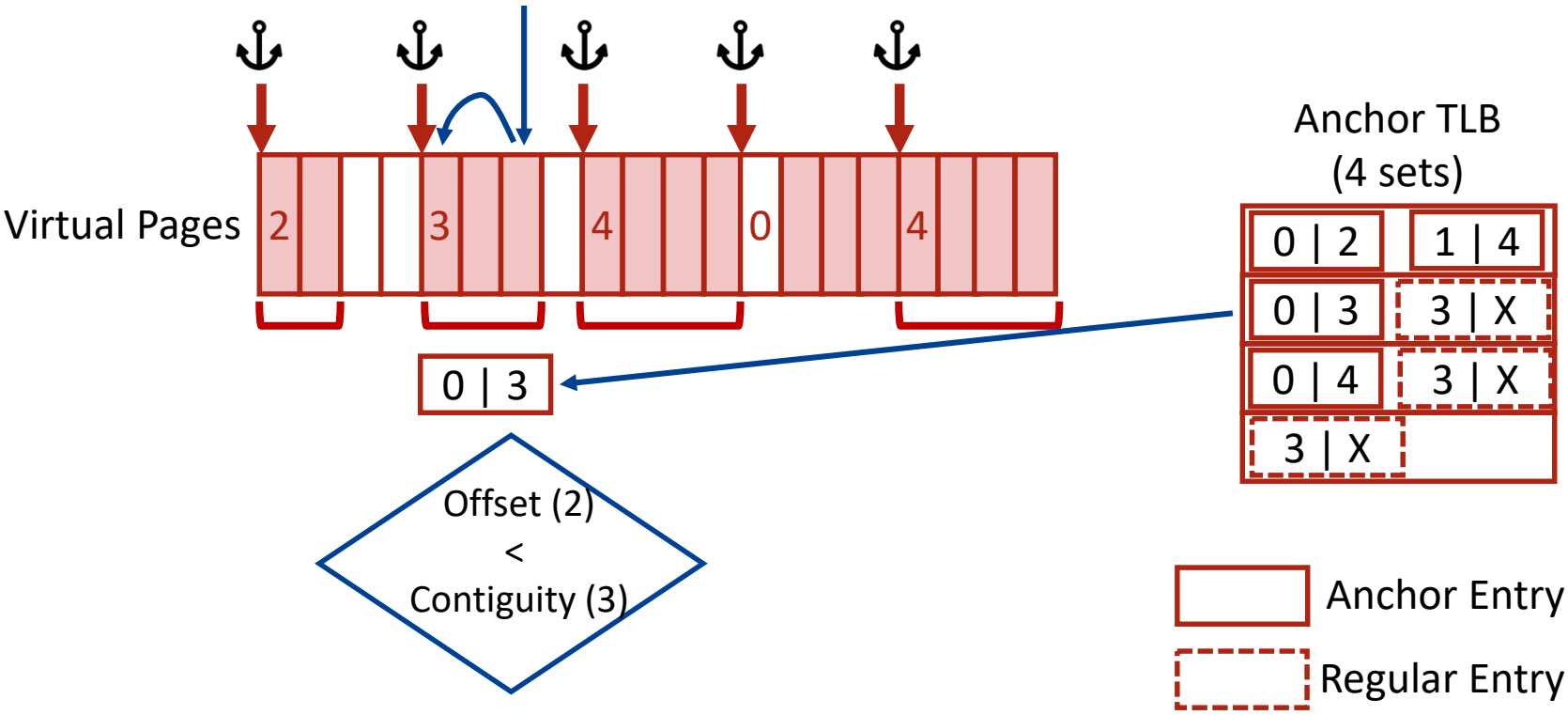
Anchor TLB Lookup

- On L1 TLB Miss Anchor TLB looks up
  - Regular TLB first
  - Anchor TLB next



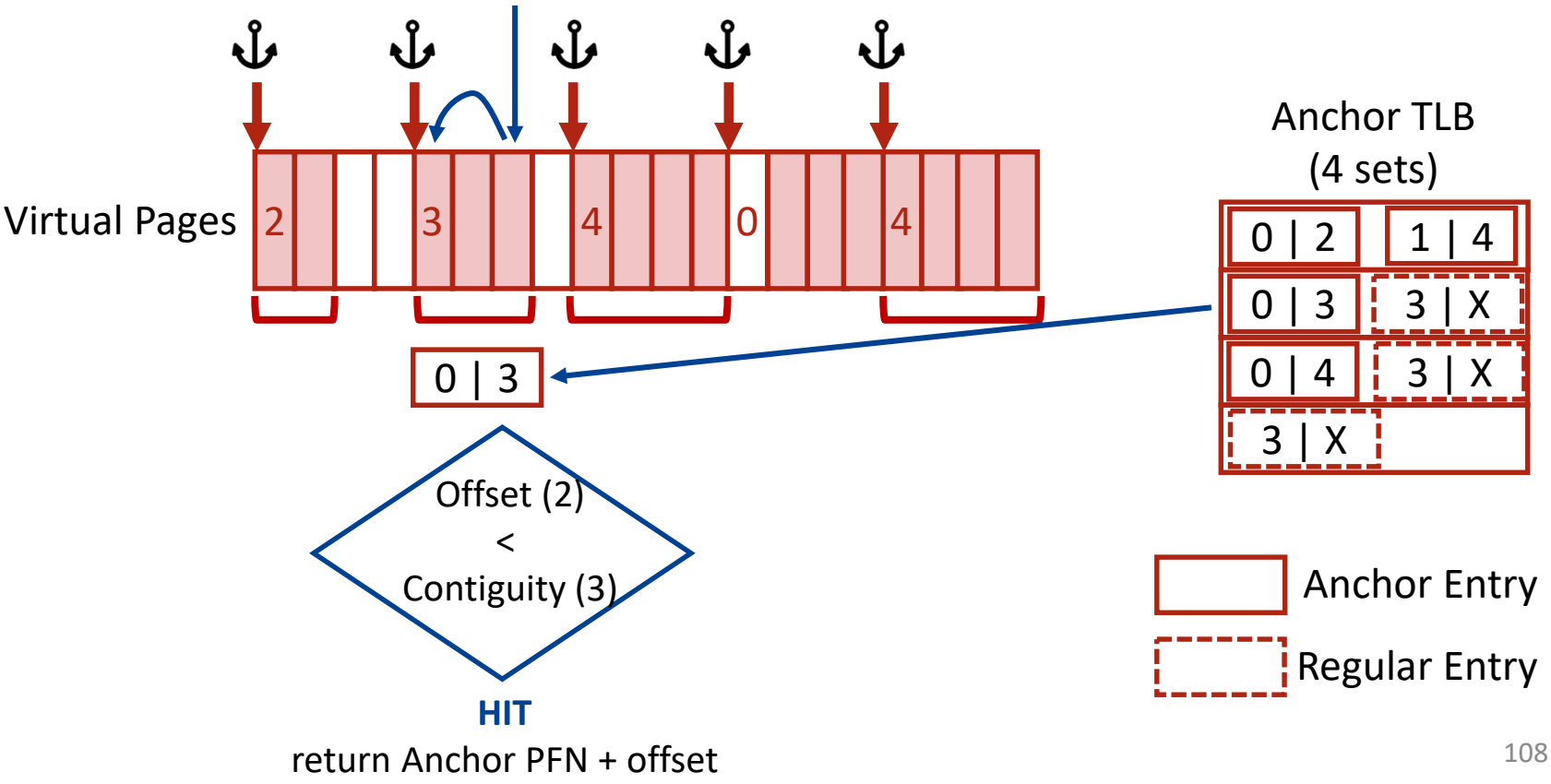
Anchor TLB Lookup

- On L1 TLB Miss Anchor TLB looks up
  - Regular TLB first
  - Anchor TLB next



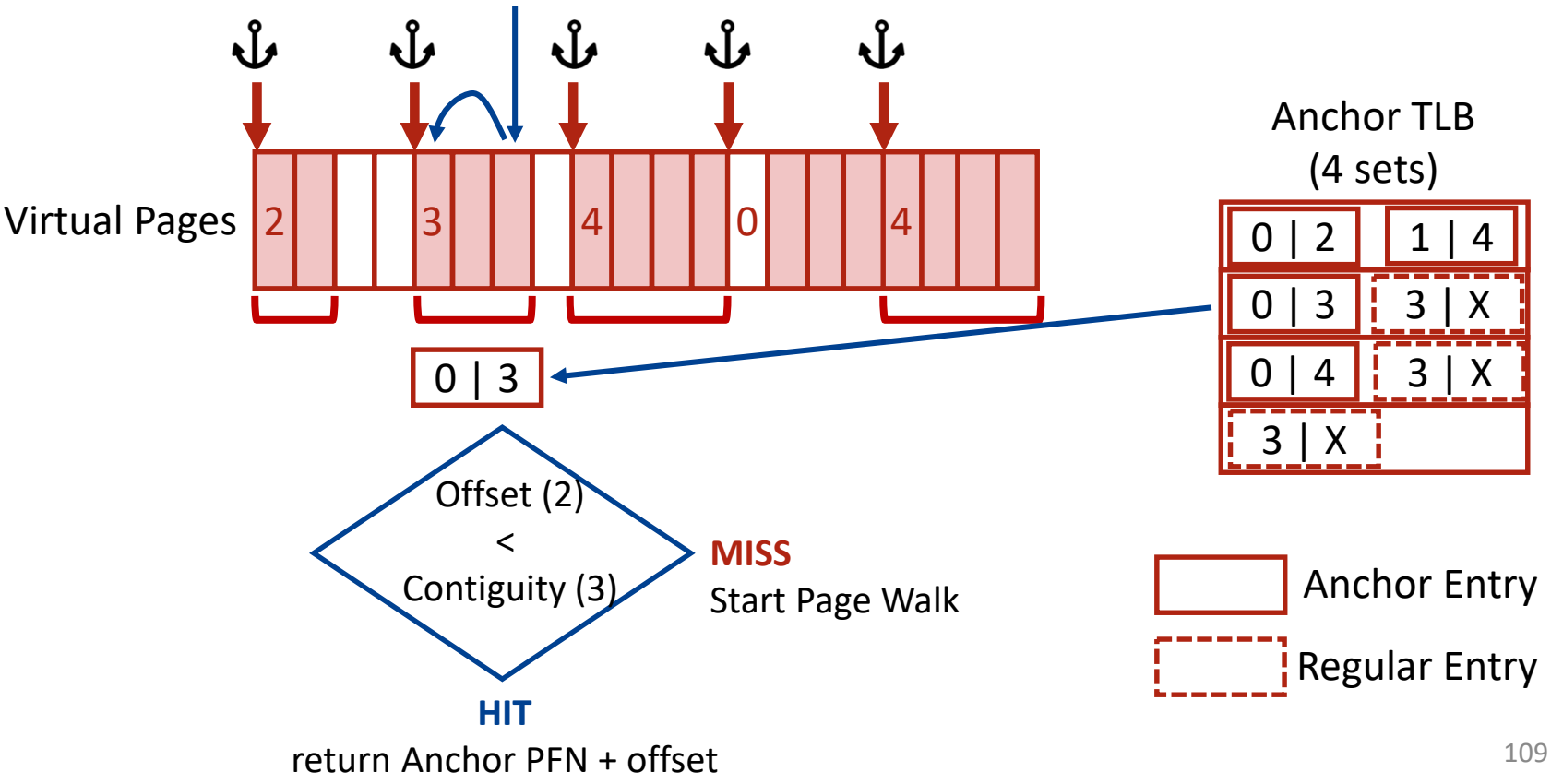
Anchor TLB Lookup

- On L1 TLB Miss Anchor TLB looks up
  - Regular TLB first
  - Anchor TLB next



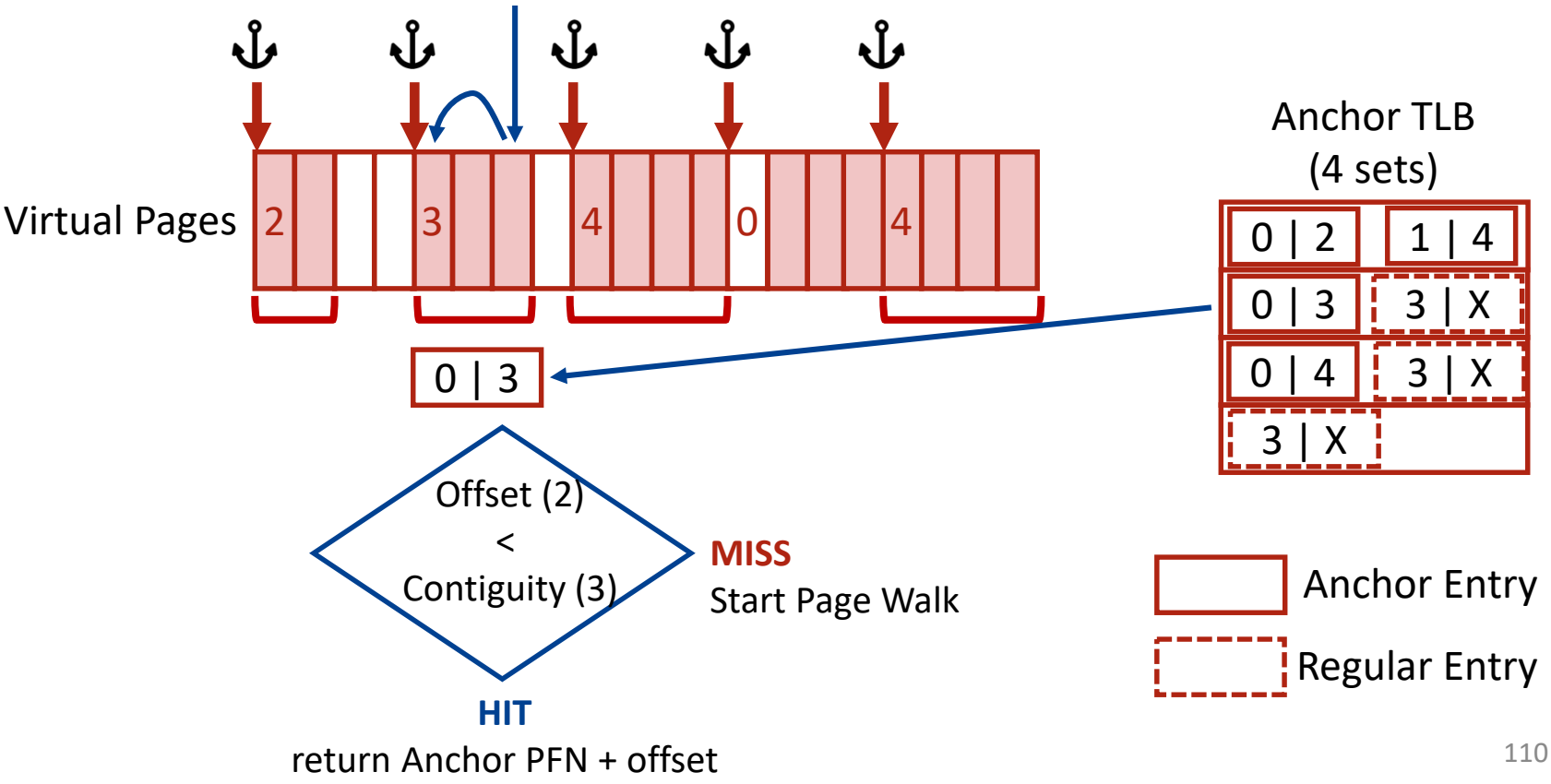
Anchor TLB Lookup

- On L1 TLB Miss Anchor TLB looks up
  - Regular TLB first
  - Anchor TLB next



Anchor TLB Lookup

- On L1 TLB Miss Anchor TLB looks up
  - Regular TLB first
  - Anchor TLB next



Operating System Responsibilities

- OS periodically selects process anchor distance
  - Heuristic algorithm to minimize TLB entry count
- OS adjusts anchor distance
  - Anchor distance based on selection algorithm
- OS marks mapping contiguity
  - Memory mapping contiguity in anchor page table entry

Simulation Methodology

- Trace based TLB simulator (Based on Intel Haswell)

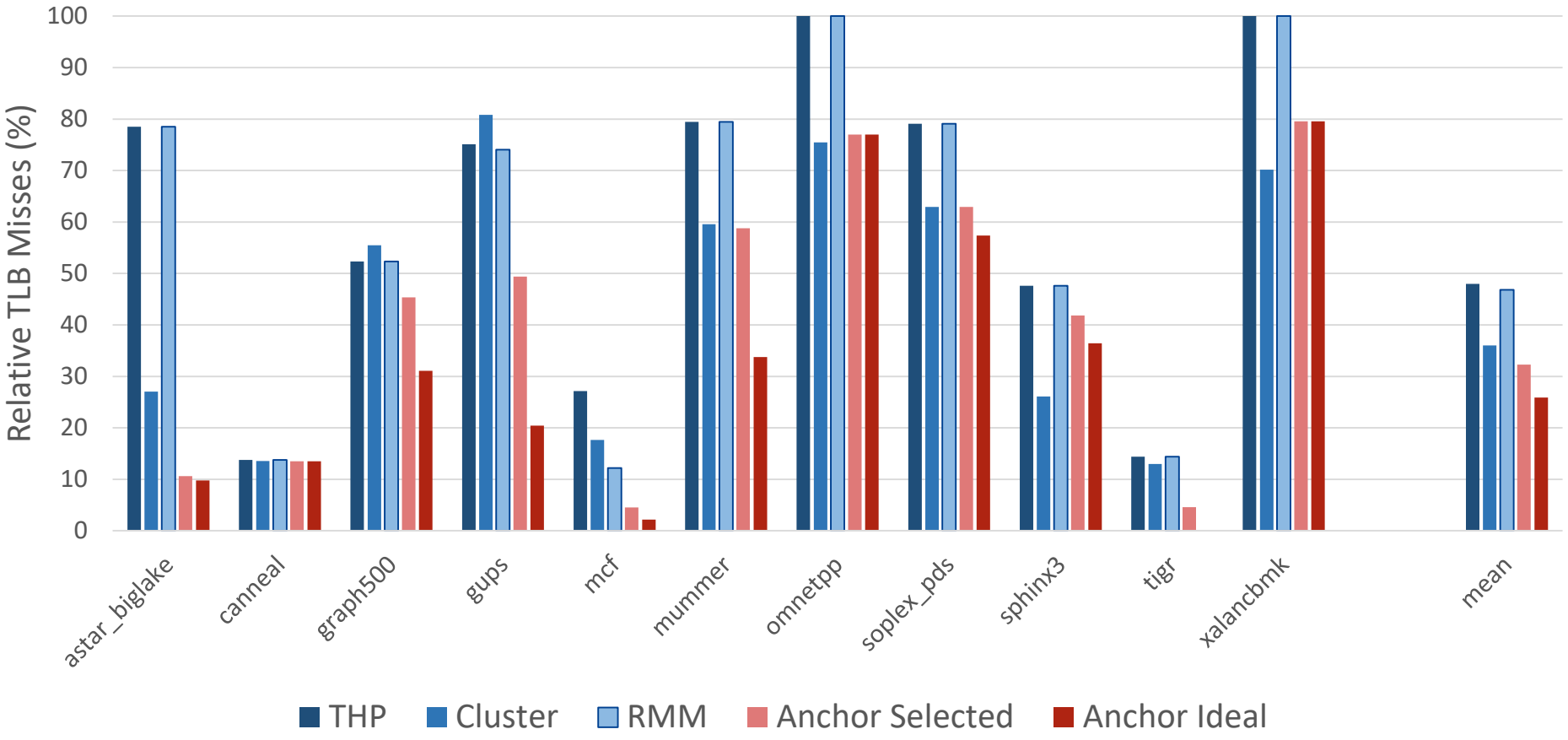
	TLB Configuration	
Common L1	4KB: 2MB:	64 entry, 4 way 32 entry, 4 way
Baseline L2 / THP	4KB/2MB:	1024 entry, 8 way
Cluster	Regular (4KB/2MB): Cluster-8:	768 entry, 6 way 320 entry, 5 way
RMM (Multiple segments)	Baseline L2 TLB + RMM:	 32 entry, fully-assoc.
Anchor (Selected/Static Ideal)	4KB/2MB/anchor:	1024 entry, 8 way

Memory Mapping Scenarios

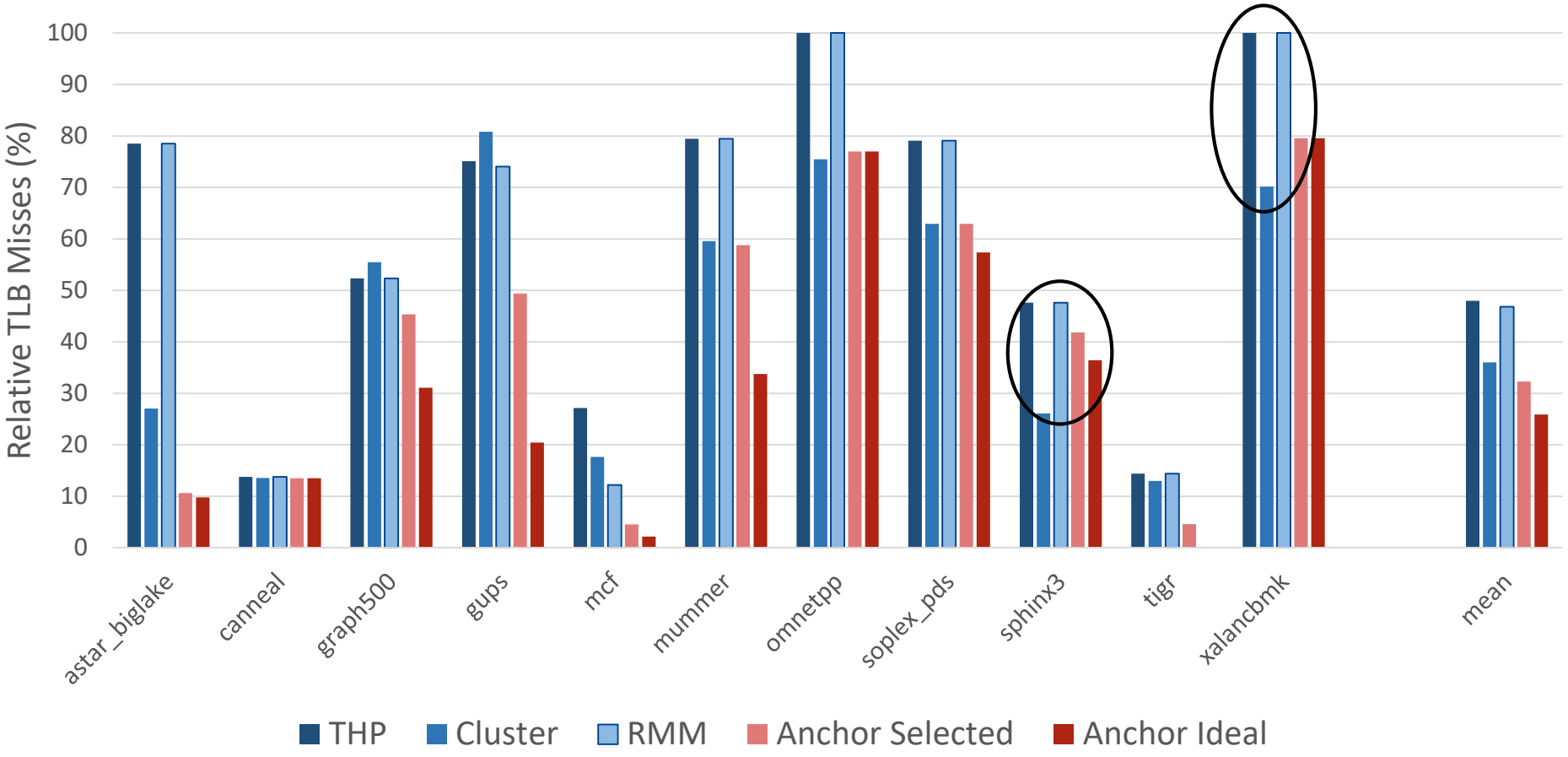
- Two class of memory mapping scenarios
  - Two real system memory mappings
  - Four synthetic memory mappings

Name	Trace information
demand	Default Linux memory mapping
eager	‘Eager’ allocation
low	1– 16 pages (4KB – 64KB)
medium	1 – 512 pages (4KB – 2MB)
high	512 – 64K pages (2MB – 256MB)
max	Maximum contiguity

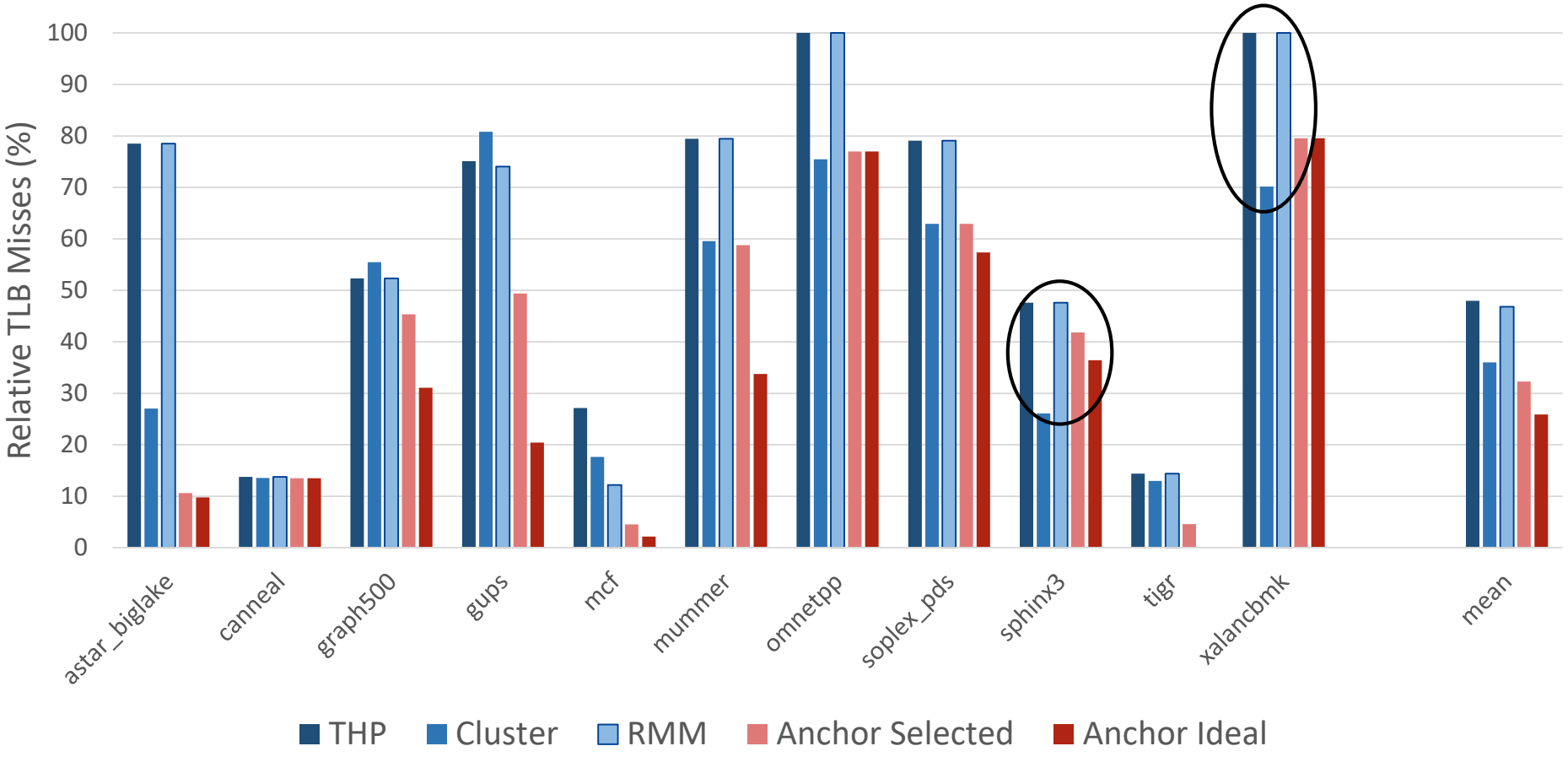
Evaluation – TLB Misses of demand **mapping**



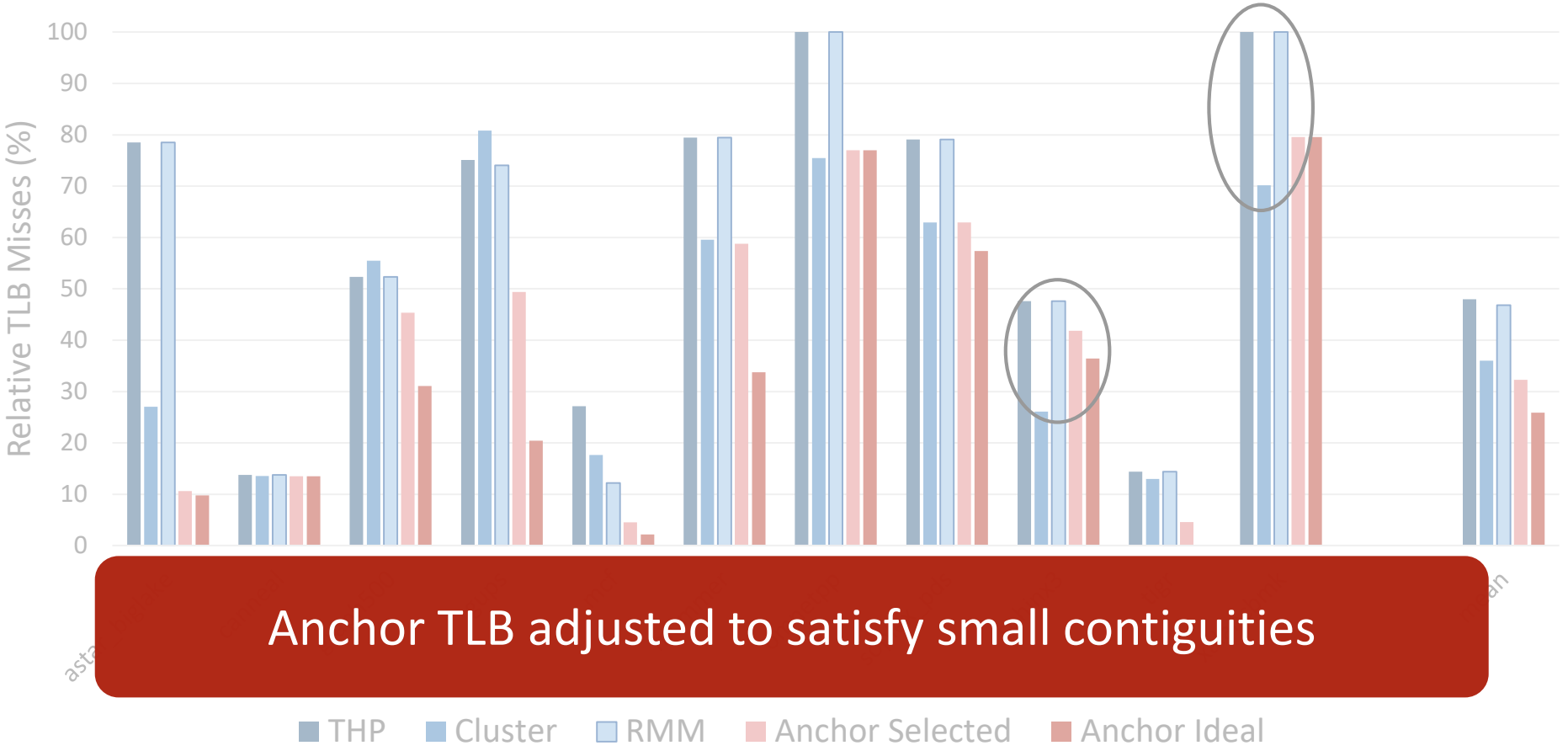
Evaluation – TLB Misses of demand **mapping**



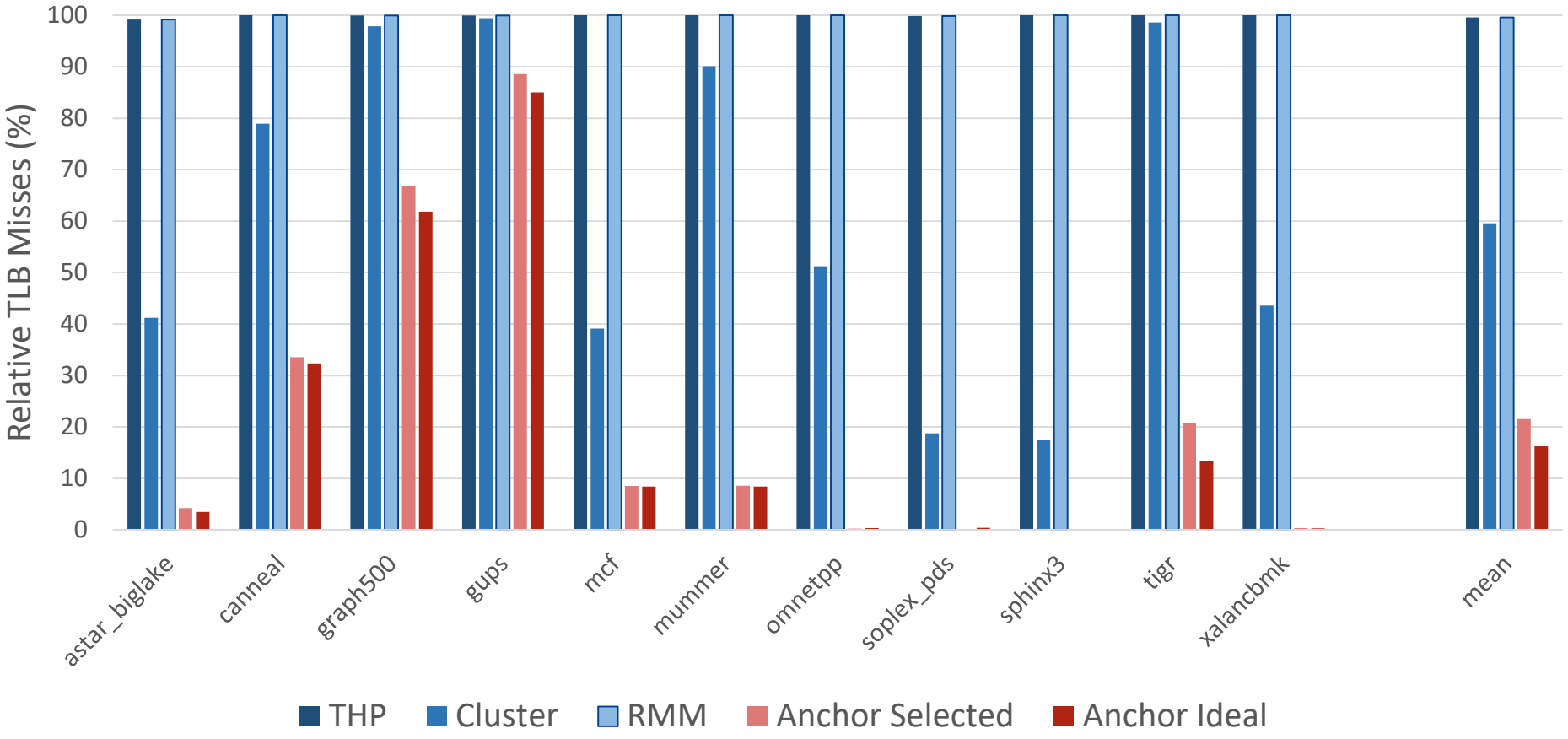
Evaluation – TLB Misses of demand **mapping**



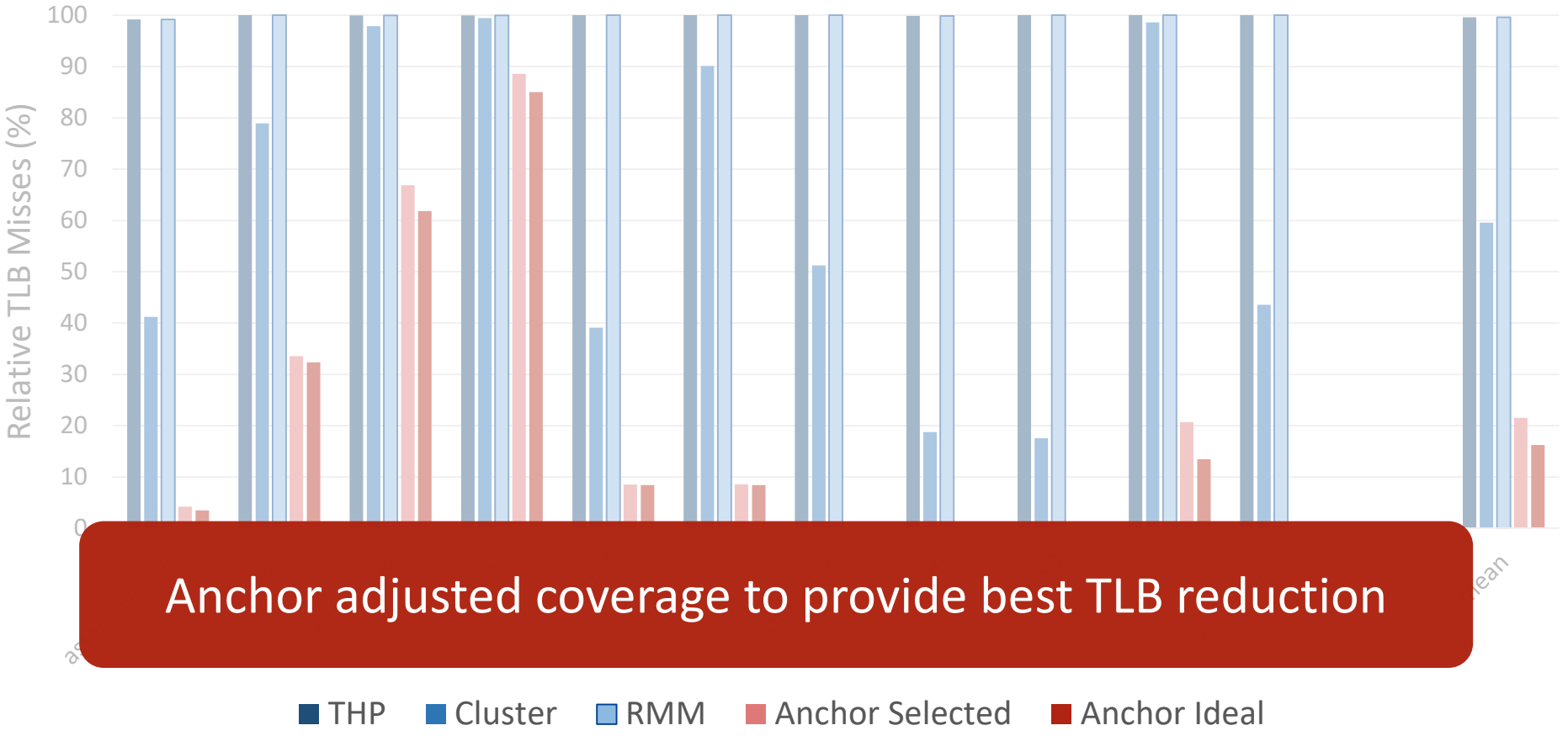
Evaluation – TLB Misses of demand **mapping**



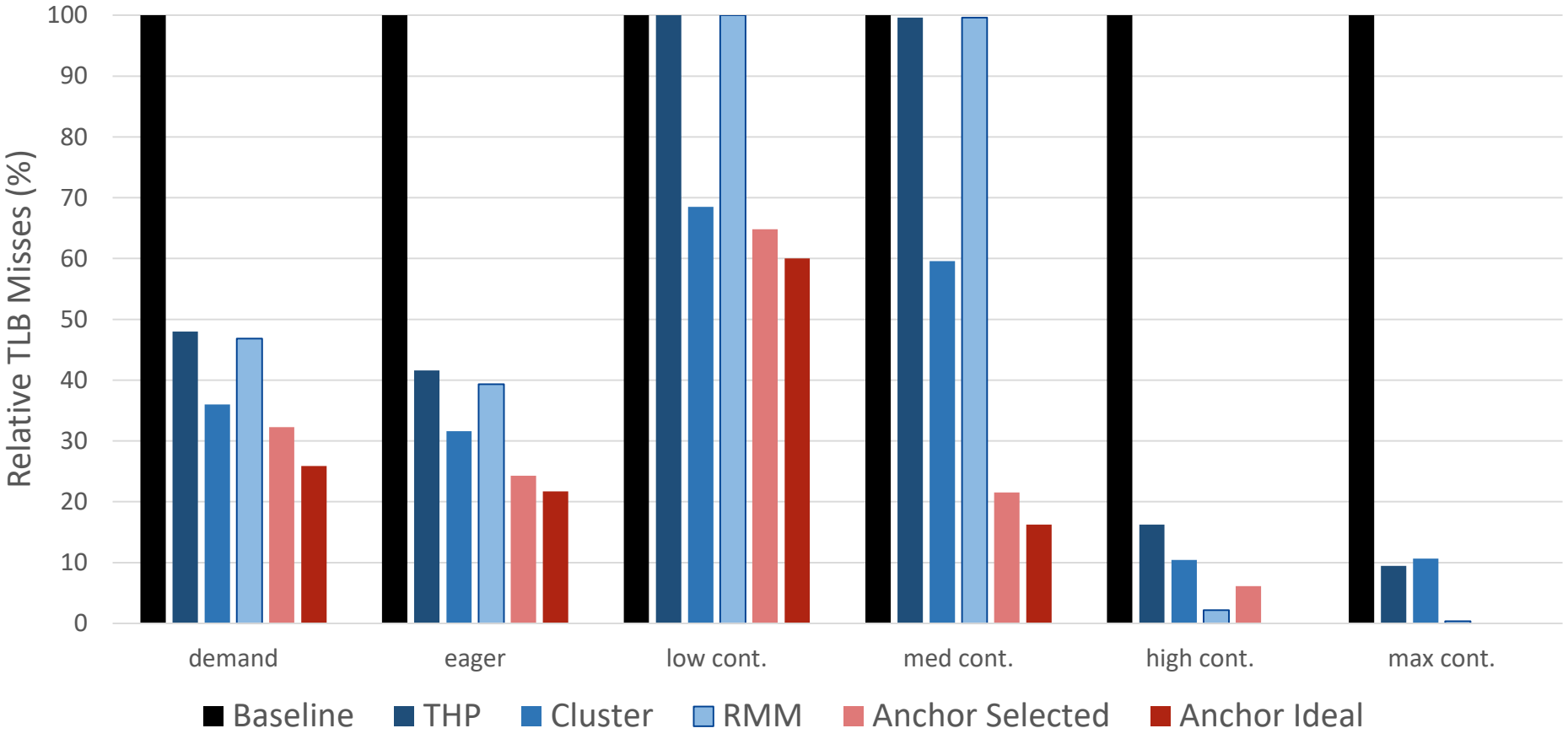
Evaluation – TLB Misses of medium **mapping**



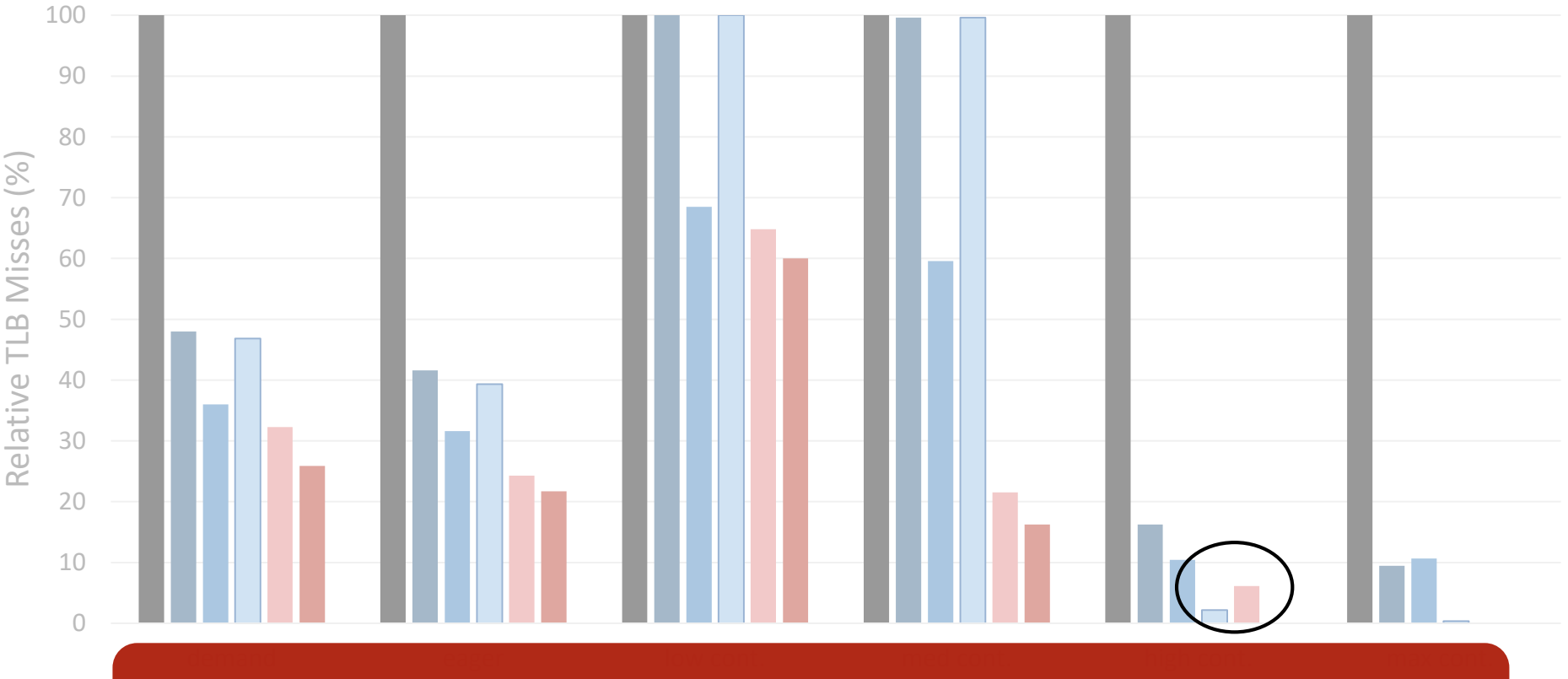
Evaluation – TLB Misses of medium **mapping**



Evaluation – TLB Misses of all mapping

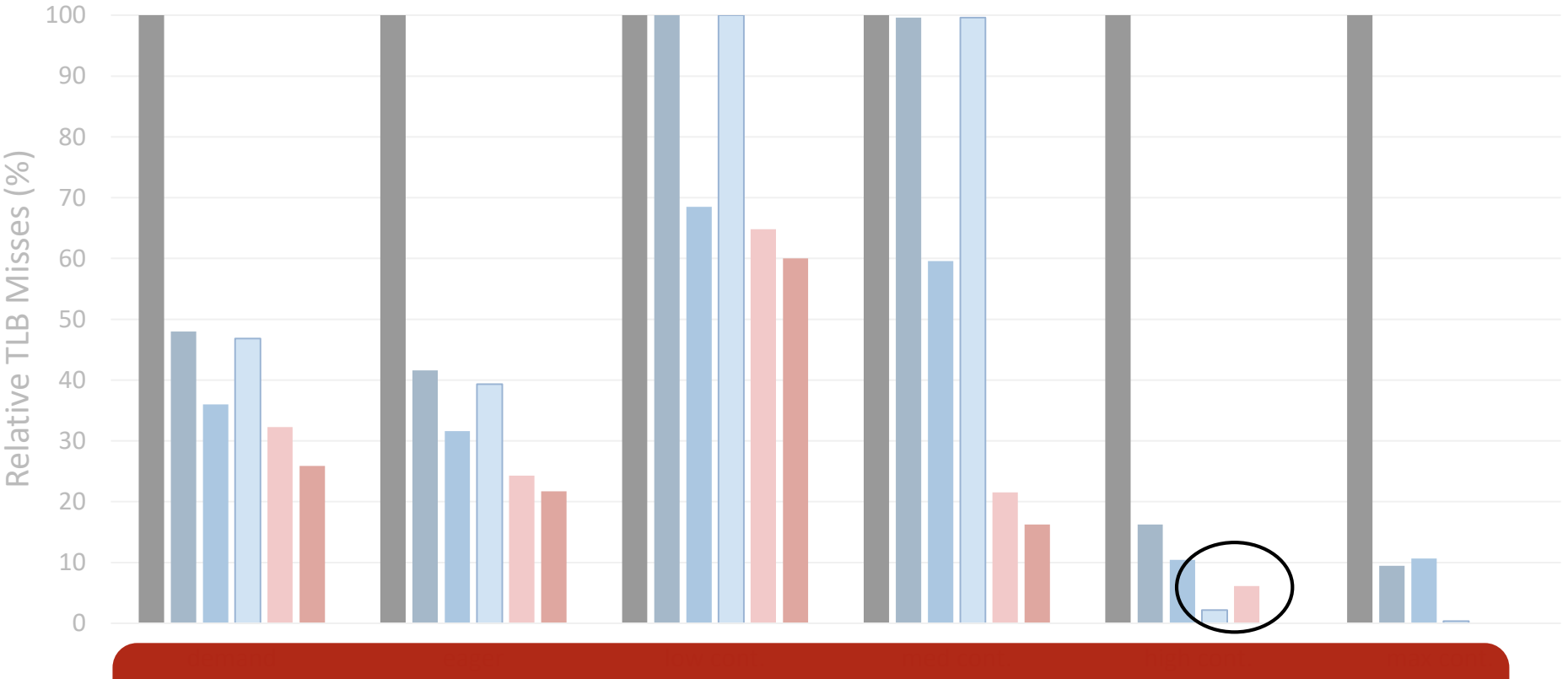


Evaluation – TLB Misses of all mapping



Anchor TLB performs well for diverse mapping scenarios

Evaluation – TLB Misses of all mapping



Anchor TLB performs well for diverse mapping scenarios

Conclusion

- Hybrid TLB Coalescing is a HW-SW joint effort
- Anchor TLB provides adjustable coverage
  - TLB entry coverage grows and shrinks dynamically
- OS provides contiguity hint using the page table
- OS picks adequate contiguity per-process

- Hybrid TLB Coalesce performs:
  - Best for Small-Intermediate contiguities
  - Similar to best prior scheme for Large contiguities