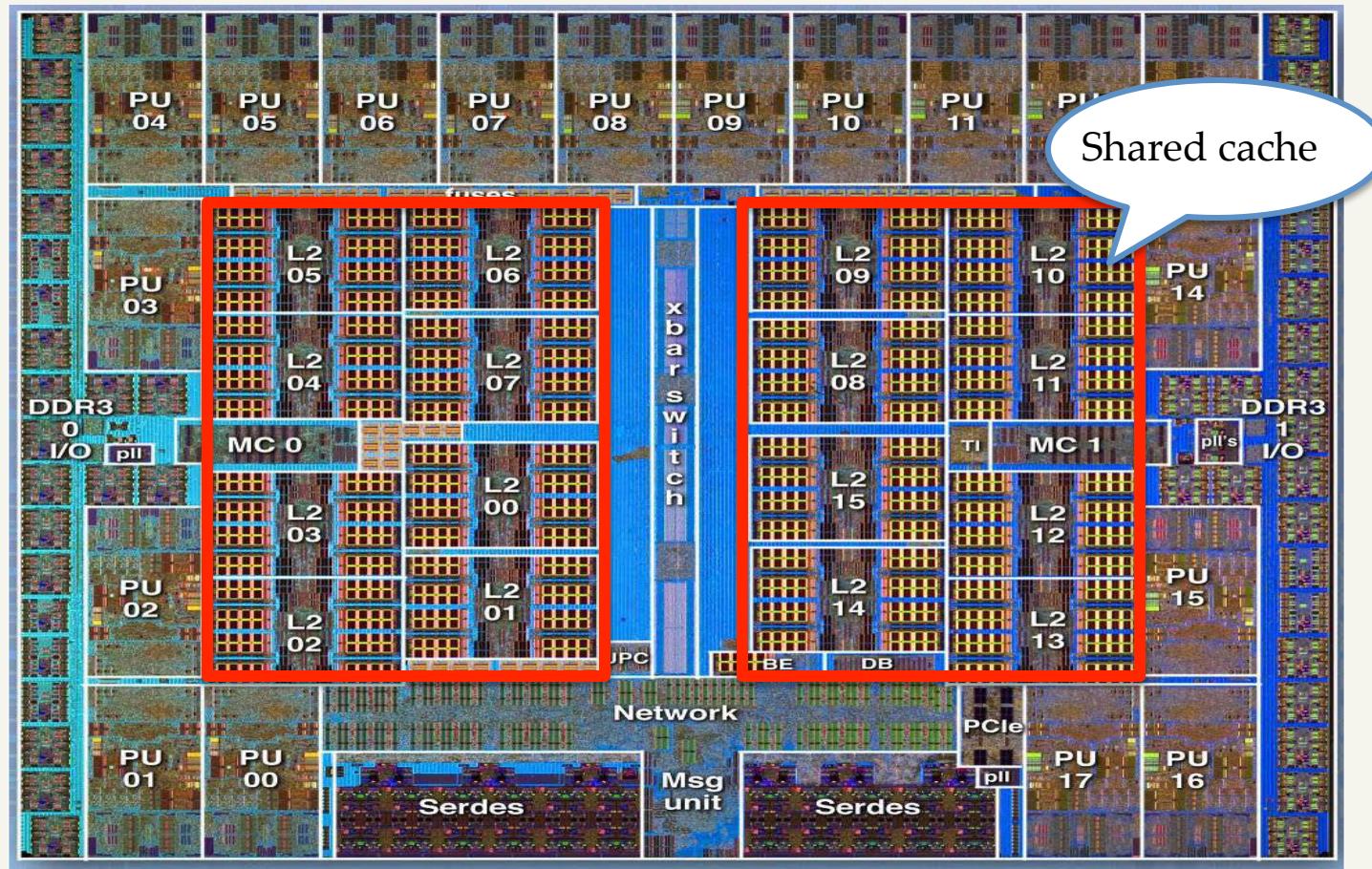




Cornell University
Computer Systems Laboratory

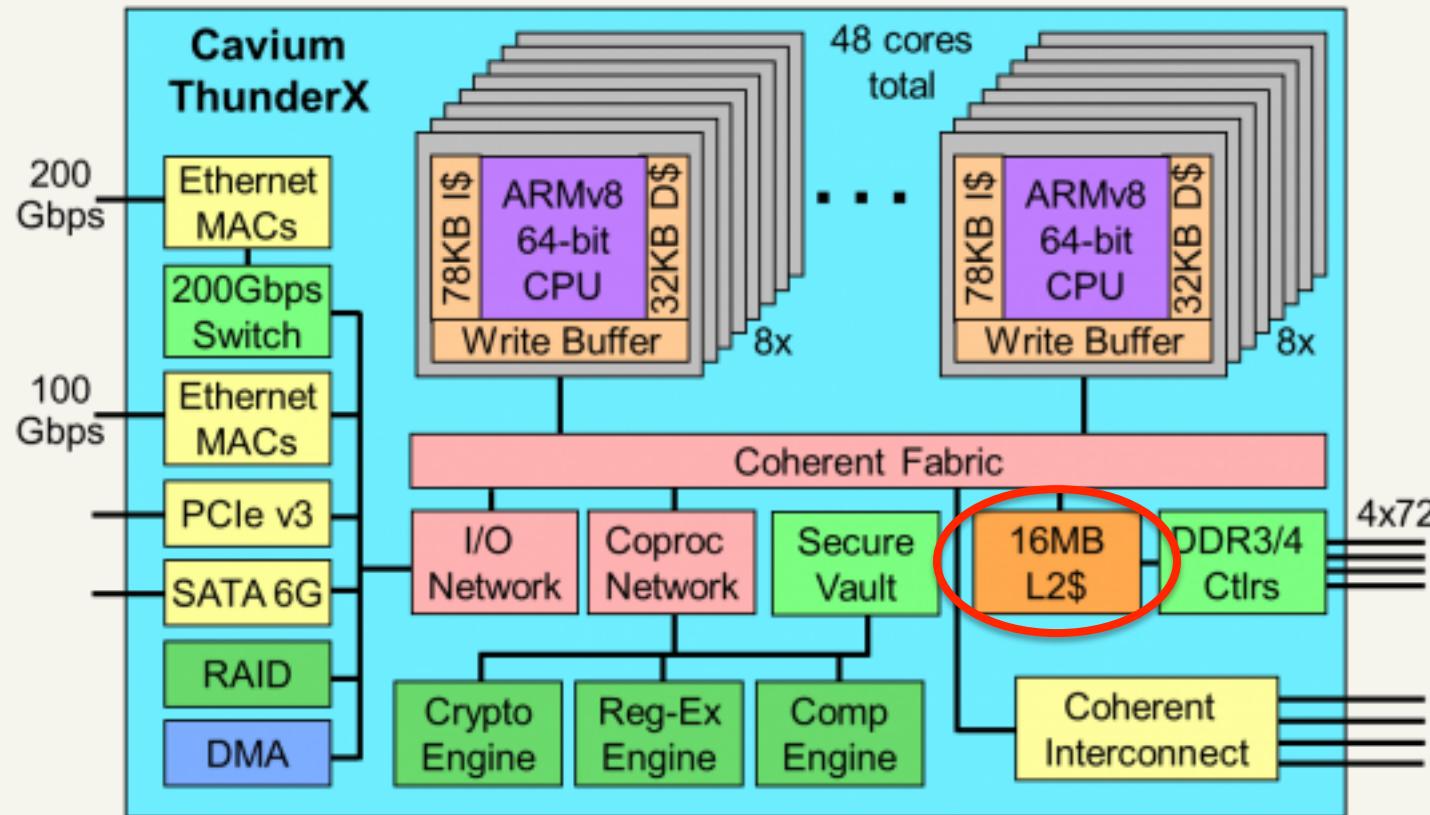
■ IBM Blue Gene/Q



Source: IBM



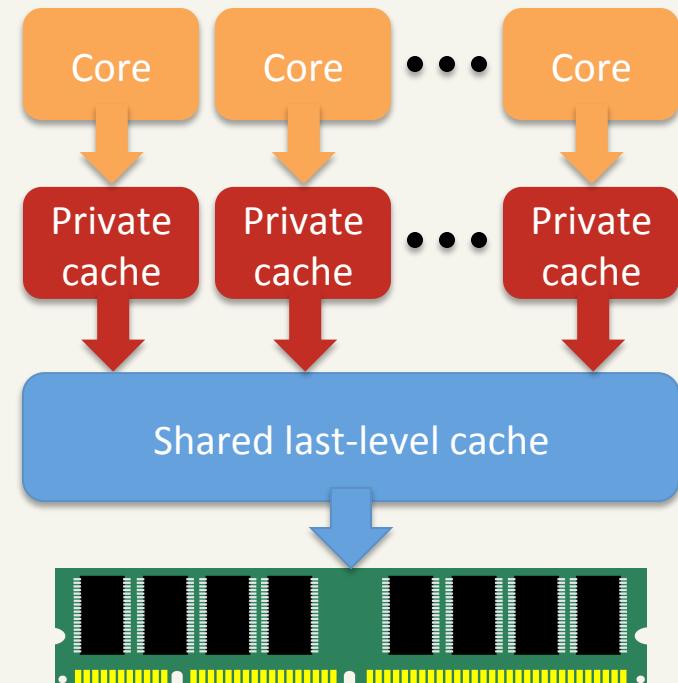
■ Cavium ThunderX® 48-core CMP



Source: Cavium



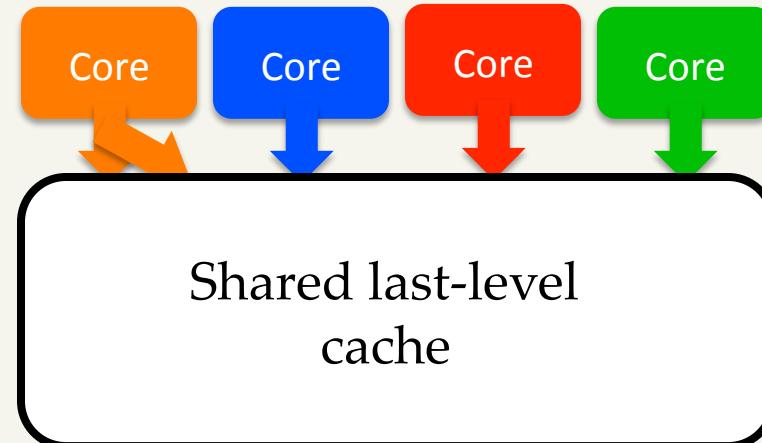
- **Last-level cache is critical to system performance**
 - ~50% chip area
- **Performance isolation in shared cache**
 - Improve system throughput
 - Guarantee QoS of latency-critical workloads
 - Eliminate timing channels



BACKGROUND

■ Cache way partition

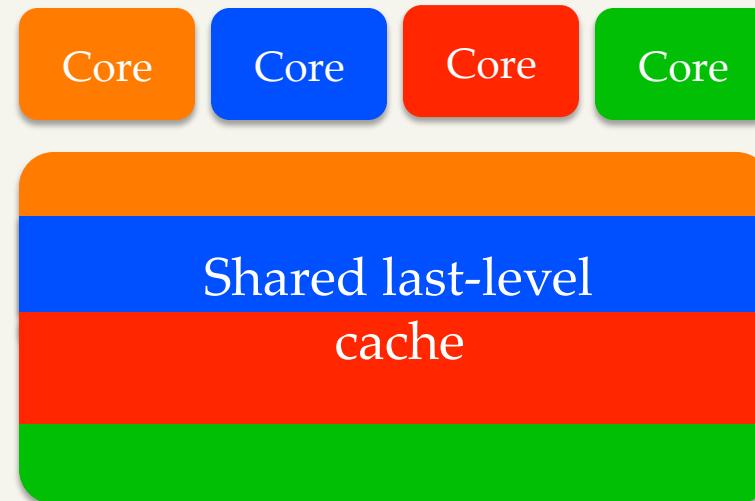
- Assign different cache ways to different cores
- Perfect isolation
- Coarse-grained
 - 16 cache ways in ThunderX 48-core processor
 - Associativity lost



BACKGROUND

■ Page coloring

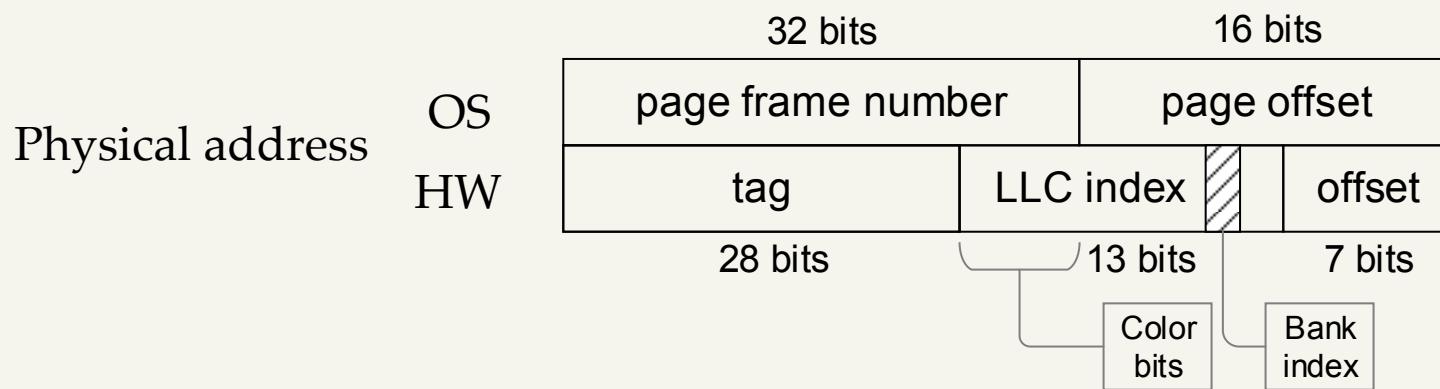
- Assign different cache sets to different cores
- Perfect isolation
- OS-level software technique



BACKGROUND

■ Page coloring

- Assign different cache sets to different cores
- Perfect isolation
- High repartition overhead
- Coarse-grained: the number of page colors is limited
 - 4 color bits, 16 colors in ThunderX 48-core processor



- **Fine-grained cache partitioning [1, 2, 3, 4]**
 - Probabilistically guarantee the size of partitions at the granularity of cache lines
 - Requires non-trivial hardware changes
 - No clear boundary across partitions: isolation is not strict

[1] Xie and Loh, ISCA' 09

[2] Sanchez and Kozyrakis, ISCA' 11

[3] Manikantan et al., ISCA' 12

[4] Wang and Chen, MICRO' 14

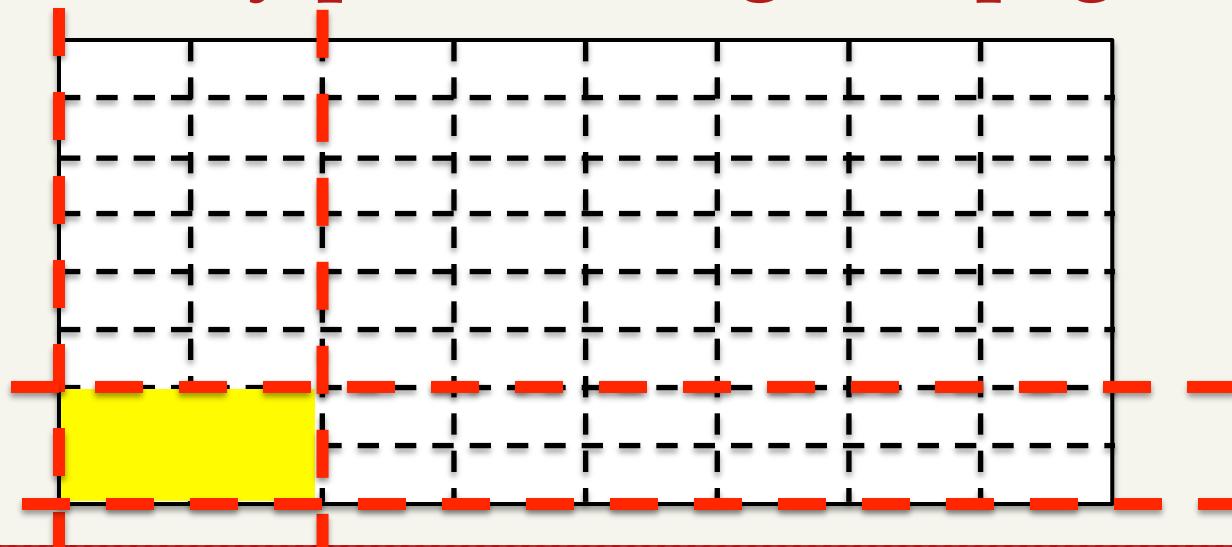


COMPARISON OF SCHEMES

	Way partitioning	Page coloring	Probabilistic partitioning	SWAP
Perfect isolation	Yes	Yes	Probabilistic	Yes
Fine-grain	No	No	Yes	Yes
Hardware overhead	Low	No	High	Low
Real system	Yes	Yes	No	Yes
Repartition overhead	Low	High	Low	Median

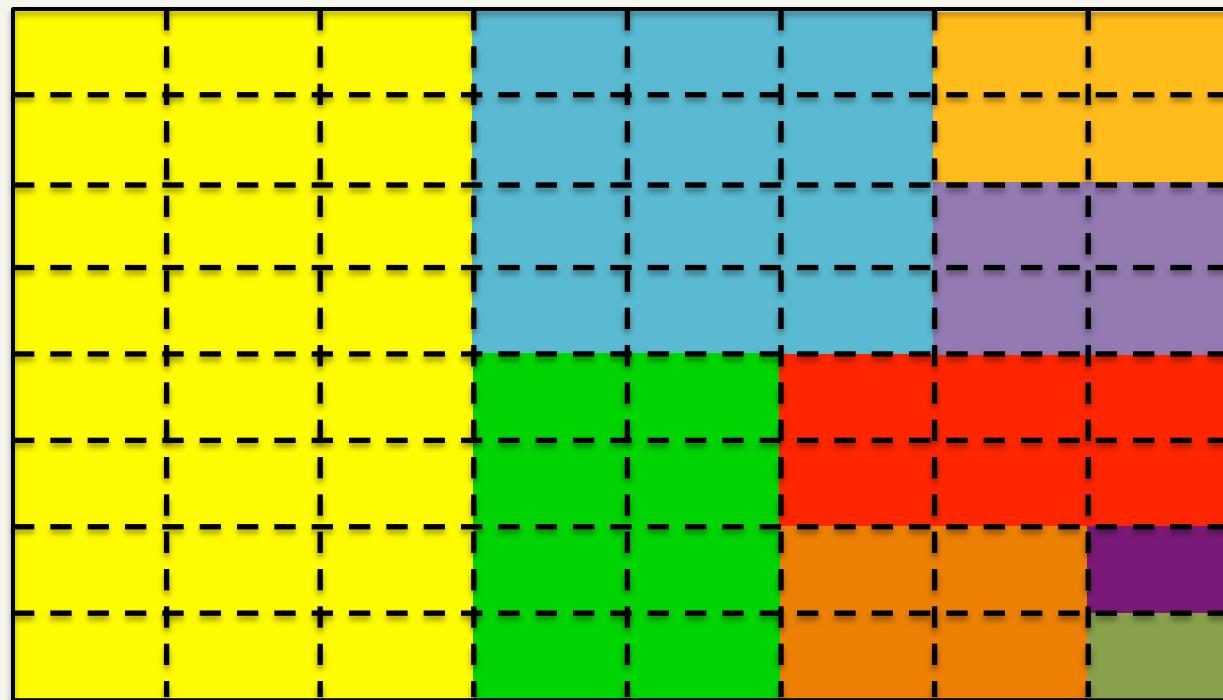


- Way partitioning vertically divides the cache
 - 16 cache ways in ThunderX for 48 cores
- Page coloring horizontally divides the cache
 - 16 page colors in ThunderX for 48 cores
- Combine way partitioning and page coloring



■ Combine way partitioning and page coloring

- Divide the cache in a 2-dimensional manner
- Maximum 256 partitions w/ 16 cache ways and 16 page colors, fine-grained enough for 48 cores



■ Contribution

- Combine way partitioning and page coloring that enables fine-grain cache partition in real systems

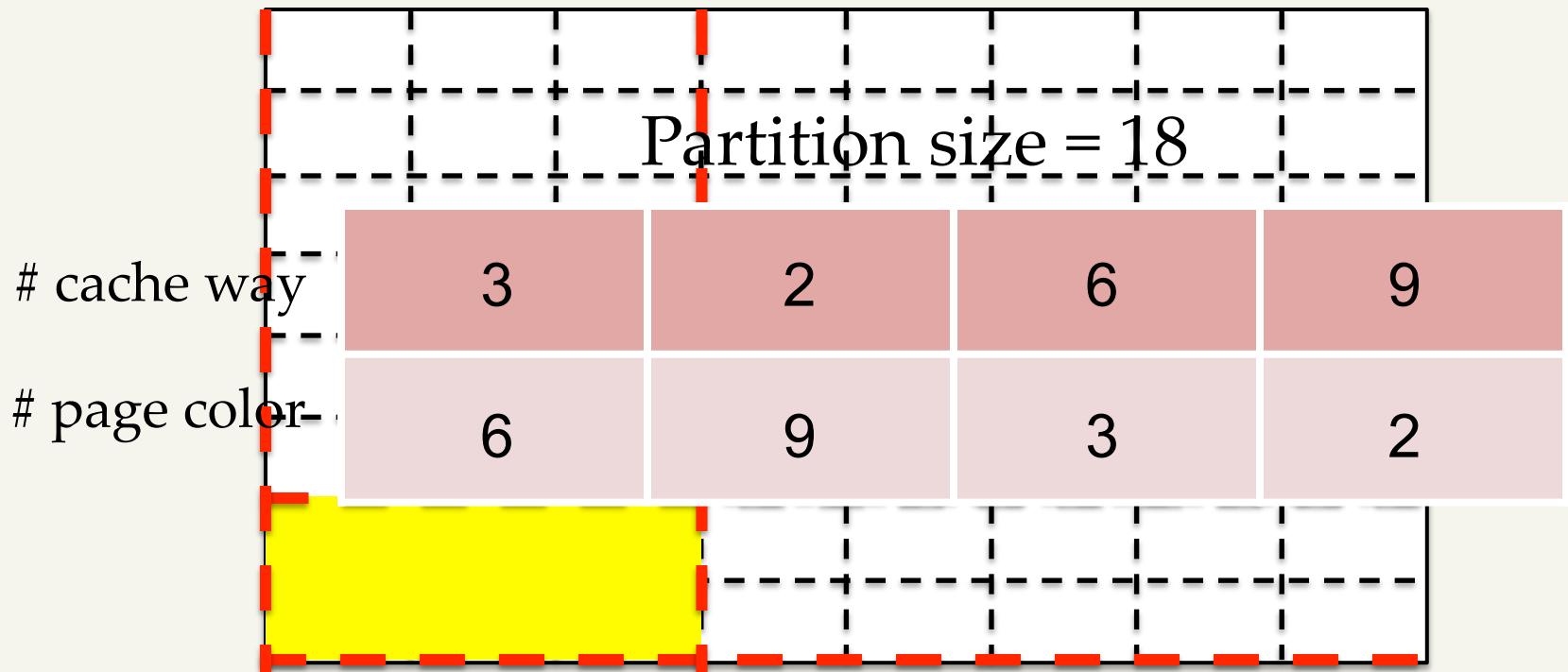
■ Challenges

- What's the shape of the partition?
- How are partitions placed with each other?
- How to minimize repartition overhead?



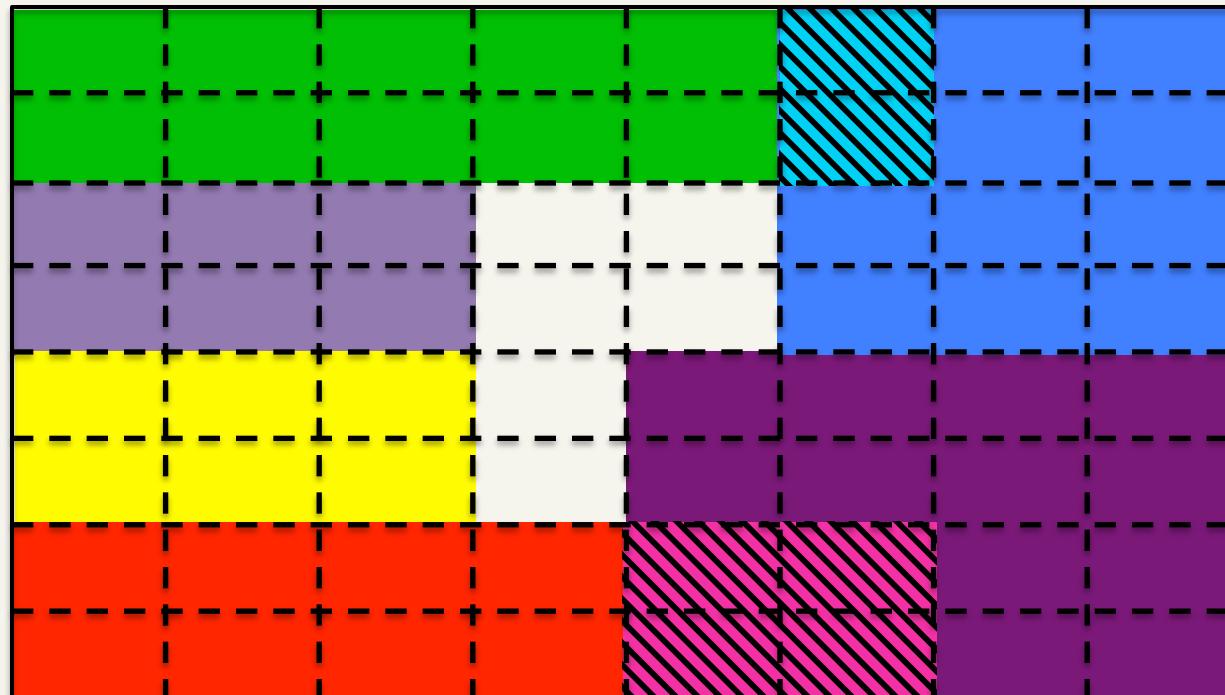
■ Partition shape

- Given the partition size, how many cache ways and pages colors should the partition have?



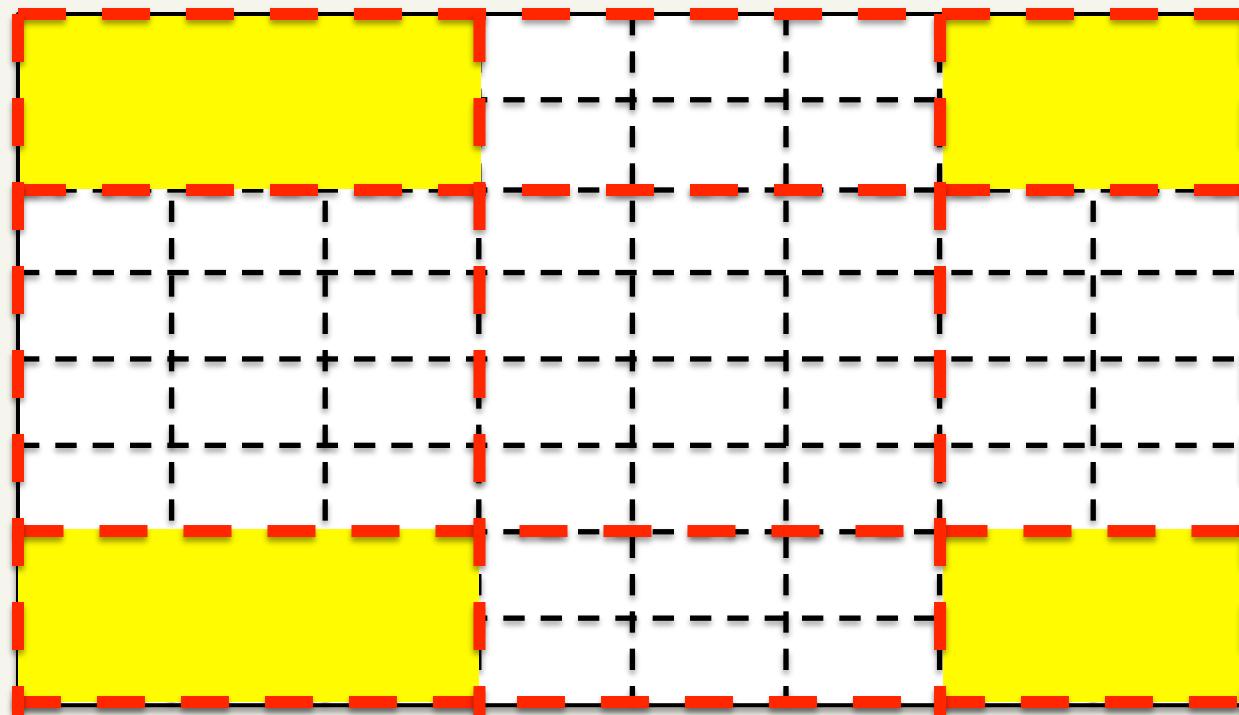
■ Partition Placement

- Partitions do not overlap (interference-free)
- No cache space is wasted



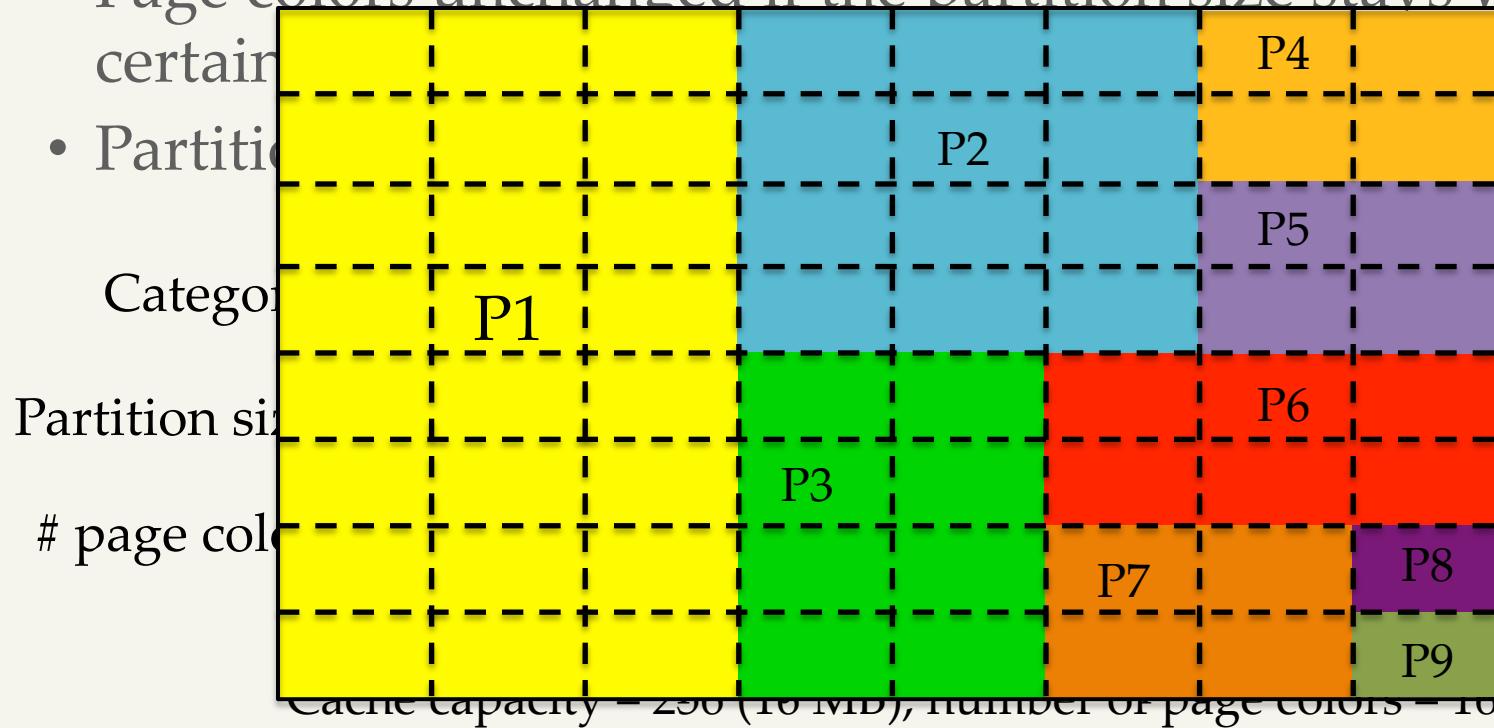
■ Partition shape

- Partitions cannot simply expand to occupy unused area



■ Partition shape

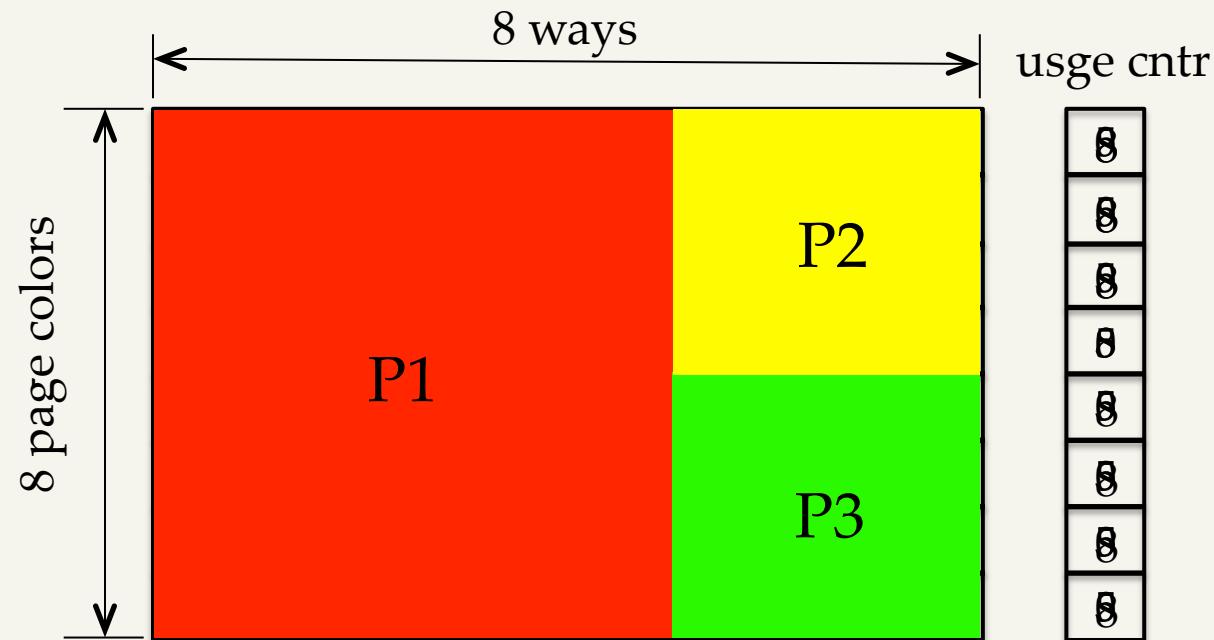
- Given the partition size, we classify the partitions into different categories
- Page colors unchanged if the partition size stays within a certain range
- Partition size is the number of pages assigned to a partition



■ Partition Placement

- Start with large partitions (with more colors)
- Assign the partition with page colors that have most cache ways left

	Size
P1	40
P2	12
P3	12



▪ Reduce repartition overhead

- Adjust cache way assignment incurs low overhead
 - » Write way permission register
- Adjust page color assignment is cumbersome
 - » Migrate the page from the old color to the new color



▪ Reduce repartition overhead

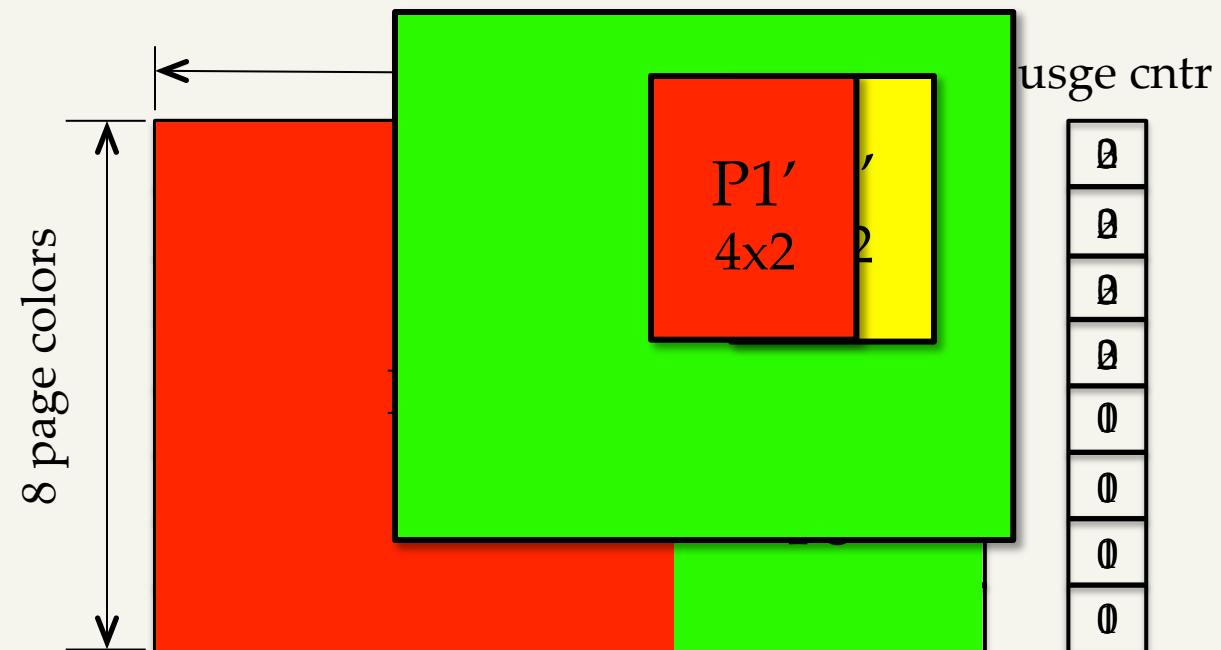
- Key: reduce page re-coloring
- Classify the partitions as before
 - » Same: keep the original colors
 - » Downgrade: use partial original colors
 - » Upgrade: may use any colors
- Estimate the cache way usage before placement



▪ Reduce repartition overhead

- Classify the partitions as before
- Estimate the cache way usage

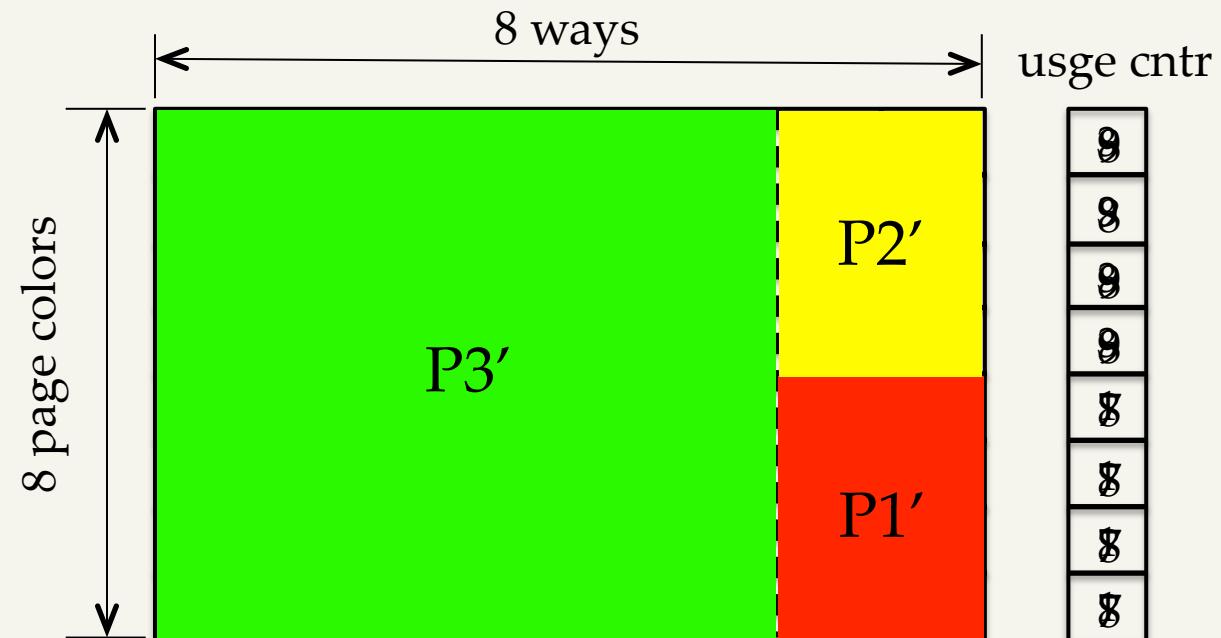
	Before	After	
P1	40	8	Down
P2	12	8	
P3	12	48	Up



▪ Reduce repartition overhead

- Start with large partitions (with more colors)
- Assign the partition with page colors that have most cache ways left

	Before	After	
P1	40	8	Down
P2	12	8	
P3	12	48	Up



OTHER ISSUES

- Cache miss-ratio curve
 - Profiling
- Lookahead algorithm [1] decides partition sizes
- Other issues
 - Hashed indexing
 - Superpage

[1] Qureshi and Patt, MICRO' 06



EXPERIMENTAL SETUP

■ Cavium ThunderX Processor

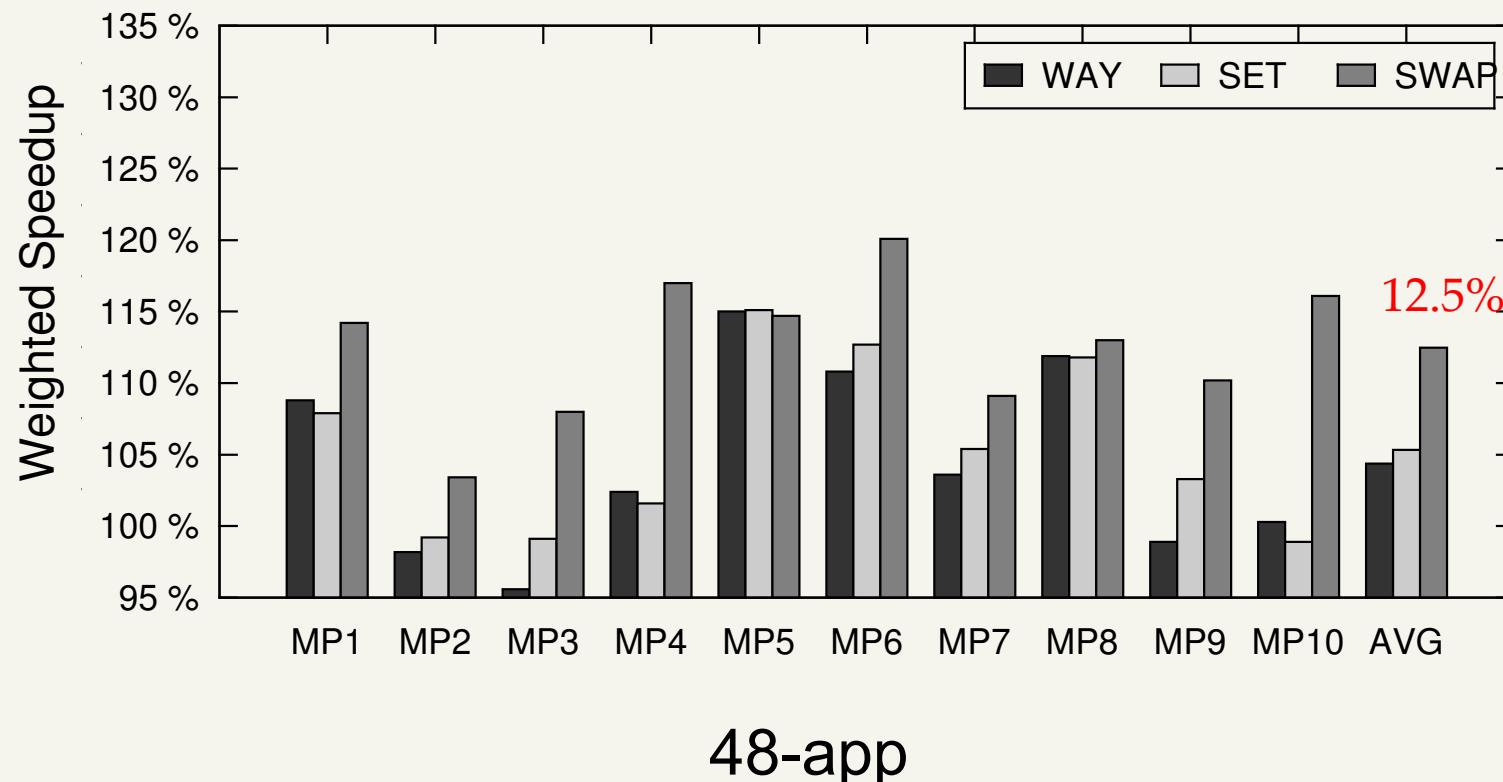
- 48-core CMP, 1.9GHz
- 16MB shared last-level cache
- 64GB DDR4-2133, 4 channels
- Ubuntu Linux 3.18

■ Performance analysis

- Mix of SPEC2000 and SPEC2006 multi-programmed workloads
- Latency critical workload memcached

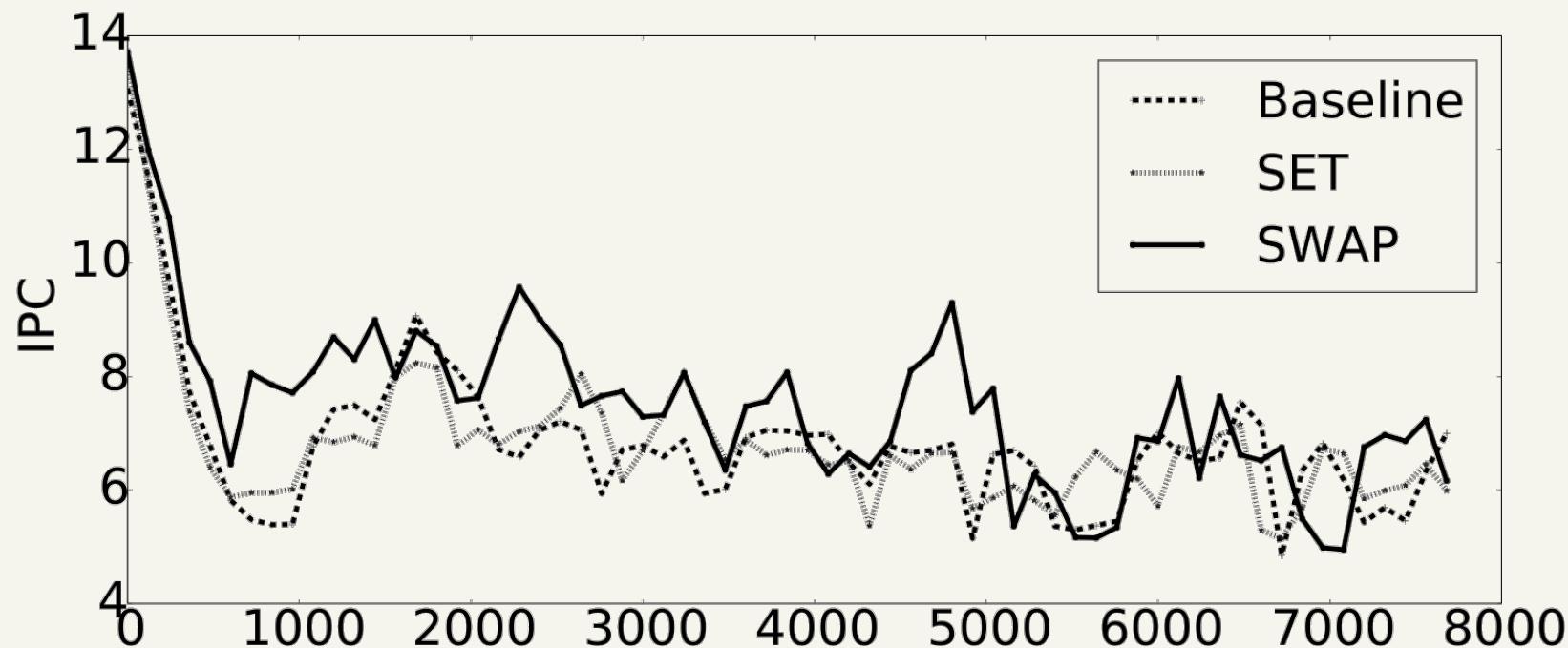


■ Running application bundle



■ Running application sequence

- The next application in the sequence replaces the finished one; cache partitions change dynamically



■ Running application sequence

- The next application in the sequence replaces the finished one; cache partitions change dynamically

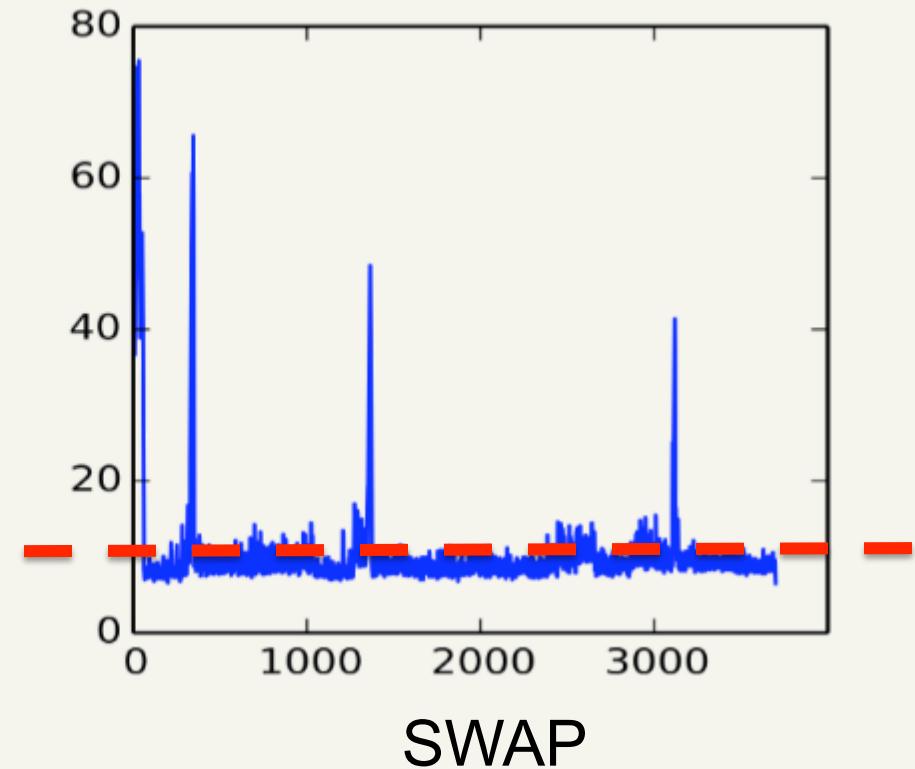
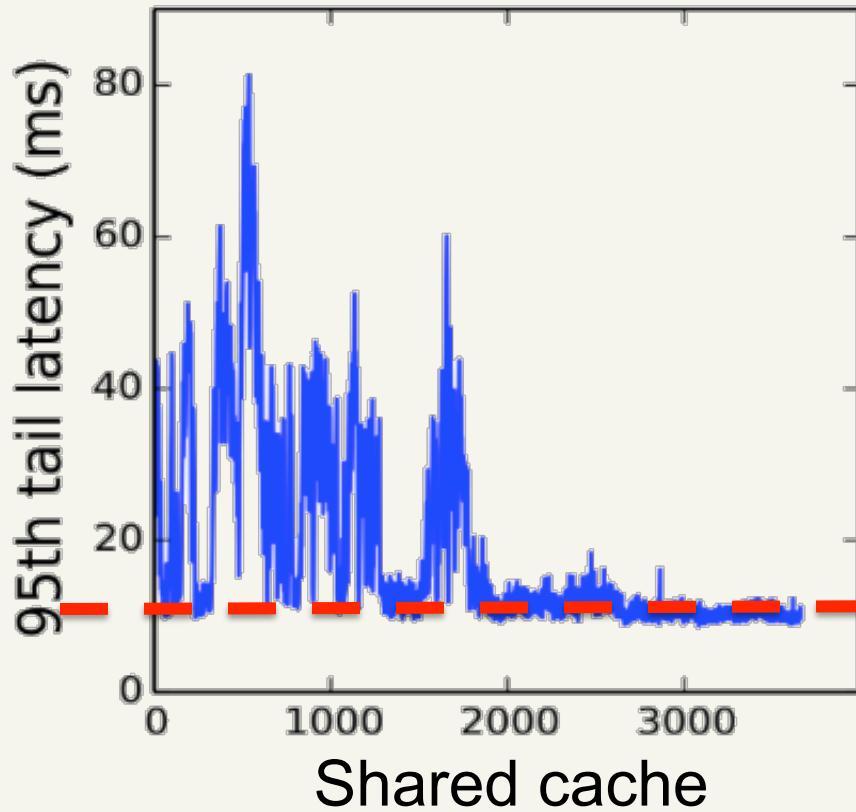
Cores	Seq	WAY	SET	SWAP	Avg. Inj interval
16	1	1.04x	1.02x	1.08x	46s
	2	1.11x	1.04x	1.17x	41s
32	1	0.97x	1.04x	1.11x	31s
	2	1.04x	1.02x	1.20x	25s
48	1	0.92x	0.99x	1.11x	34s
	2	1.00x	1.03x	1.15x	25s

SWAP

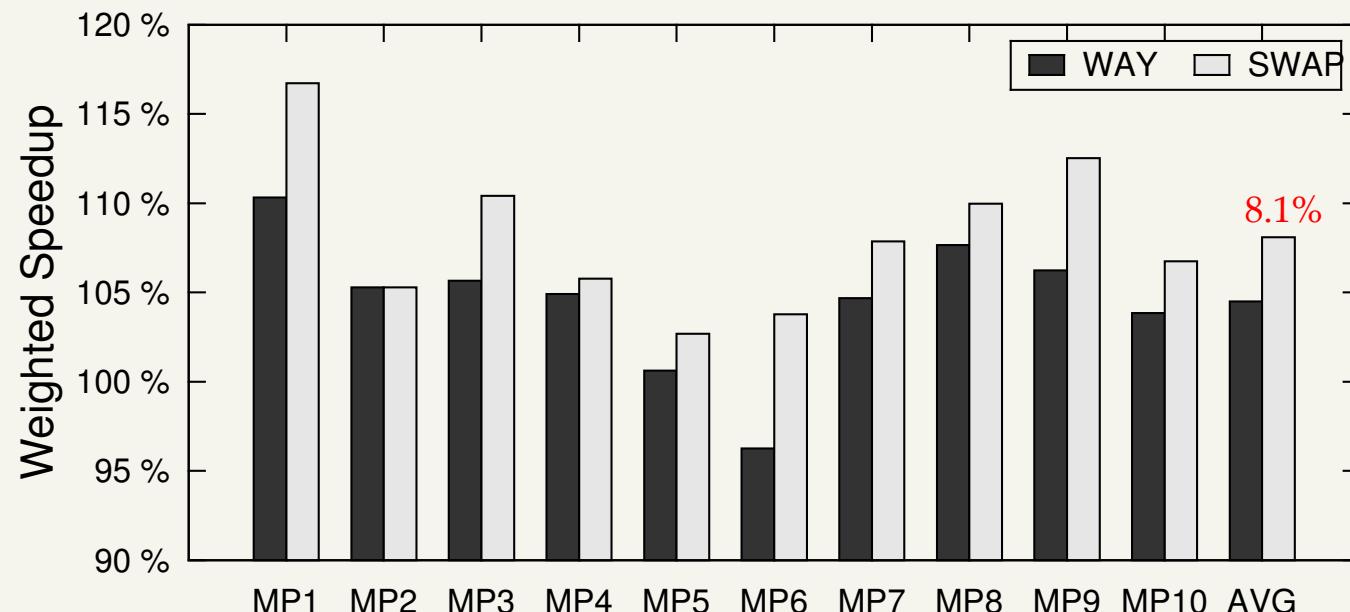
SWAP • Evaluations • Conclusions



- Latency workload memcached co-located with background multi-programmed workloads



- Latency workload memcached co-located with background multi-programmed workloads



16-app SPEC bundle

SWAP

SWAP • Evaluations • Conclusions



Cornell University
Computer Systems Laboratory

CONCLUSIONS

■ A real system implementation of fine-grain cache partitioning in large CMP systems

- Combine cache way partitioning and page coloring
- Delivers superior system throughput
- Guarantee QoS of latency-critical workloads



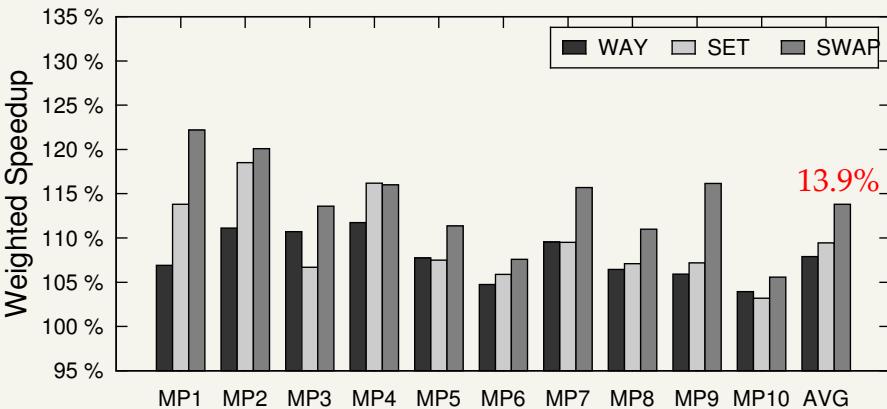


SWAP: EFFECTIVE FINE-GRAIN MANAGEMENT OF SHARED LAST-LEVEL CACHES WITH MINIMUM HARDWARE SUPPORT

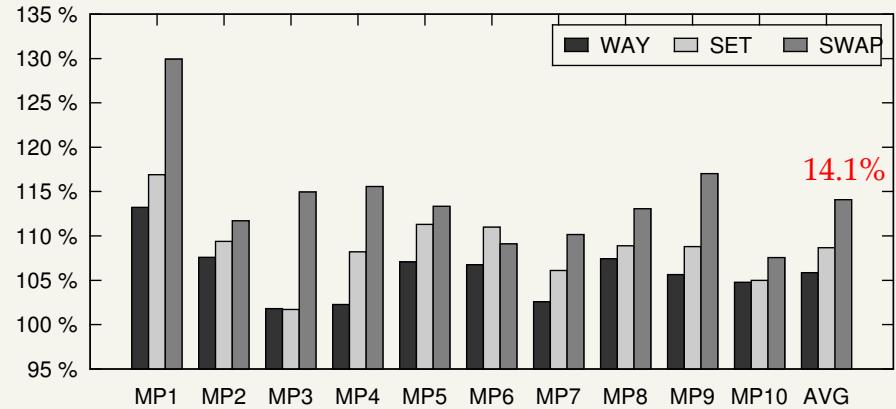
Xiaodong Wang, Shuang Chen, Jeff Setter,
and José F. Martínez

Computer Systems Lab
Cornell University

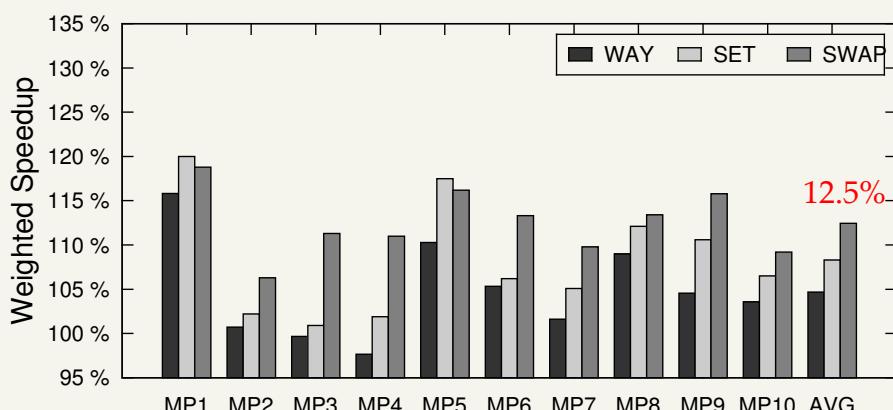
STATIC PARTITIONING



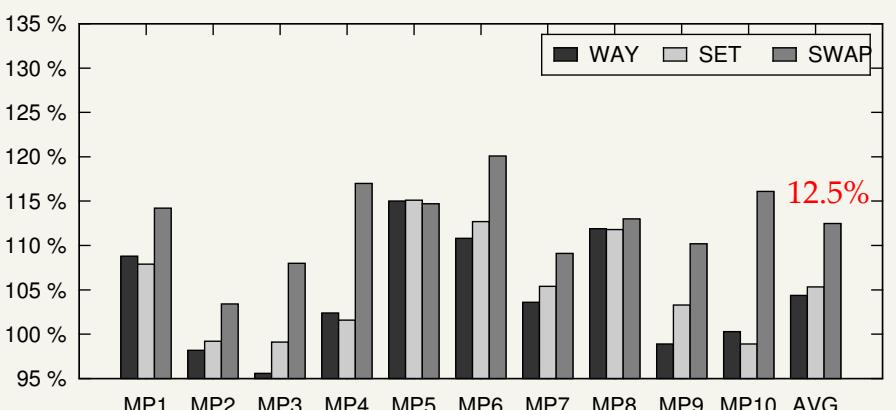
16-core



24-core



32-core



48-core

SWAP

