M3 Architecture Research Group
Cornell University

HPCA 2017
Xiaodong Wang, Shuang Chen,
Jeff Setter, José Martínez

SWAP: Effective Fine-Grain Management of Shared Last-Level Caches with Minimum Hardware Support

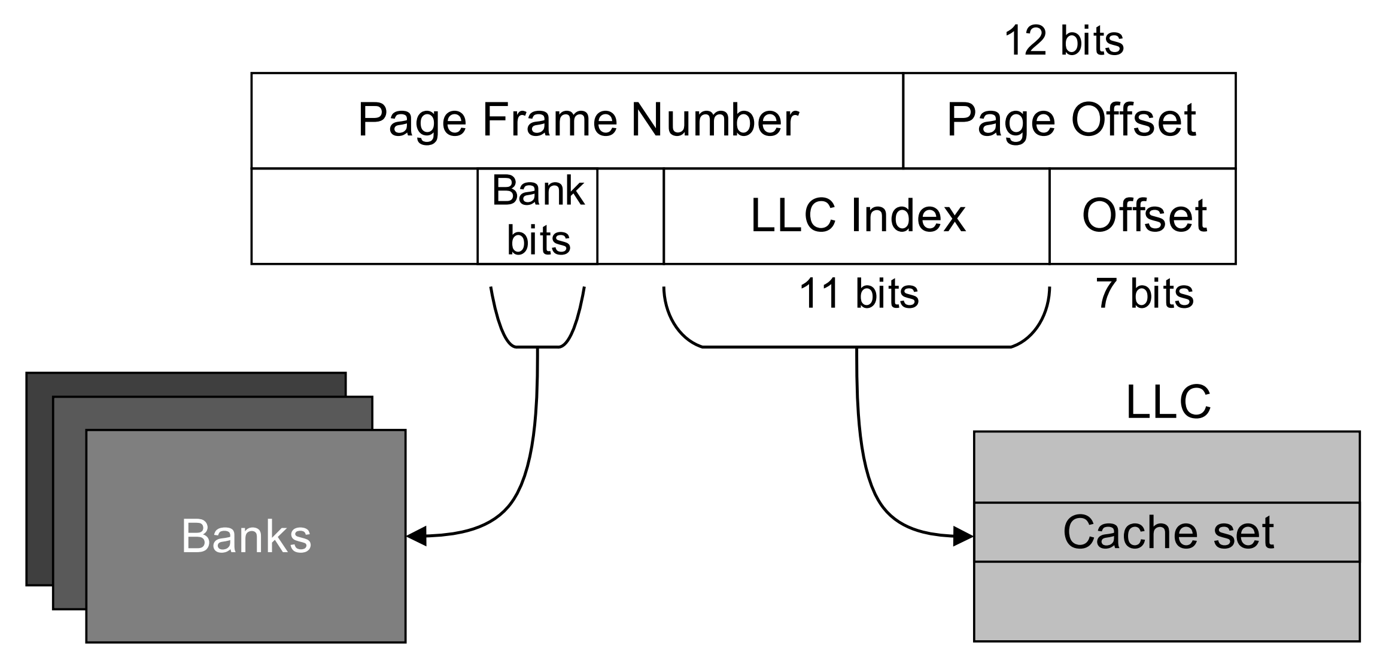
Introduction

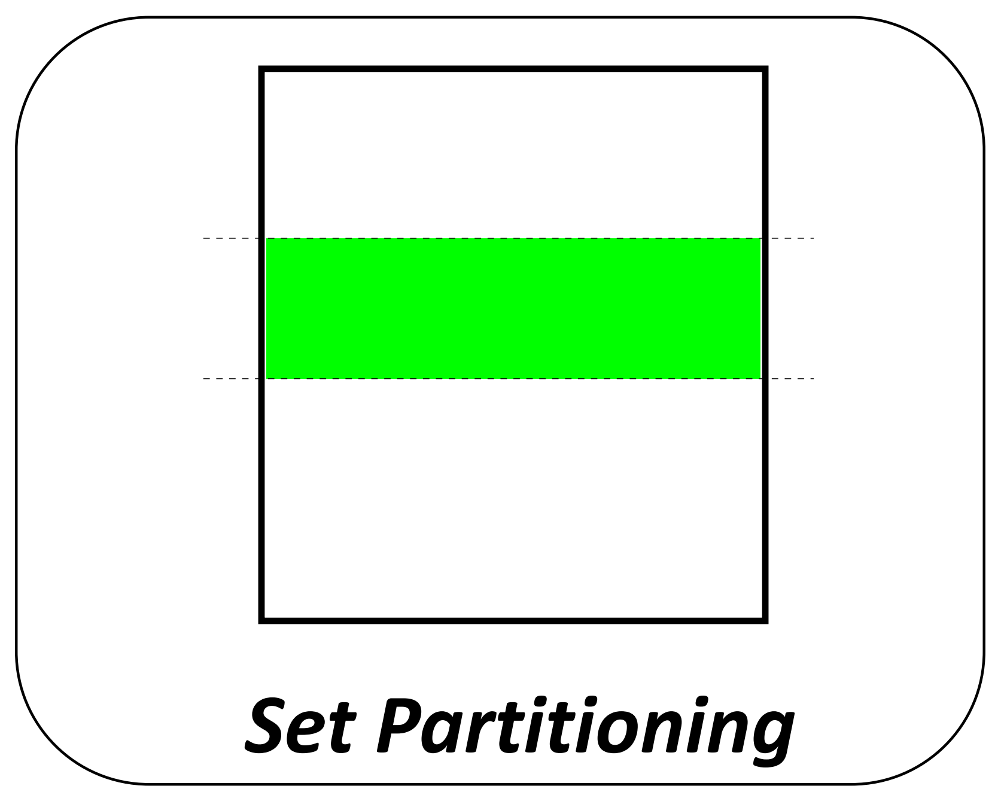
Performance isolation is an important goal in server-class environments. Partitioning the last-level cache of a chip multiprocessor (CMP) across co-running applications has proven useful in this regard. Two popular approaches are (a) hardware support for way partitioning, or (b) operating system support for set partitioning through page coloring. Unfortunately, neither approach by itself is scalable beyond a handful of cores without incurring in significant performance overheads.

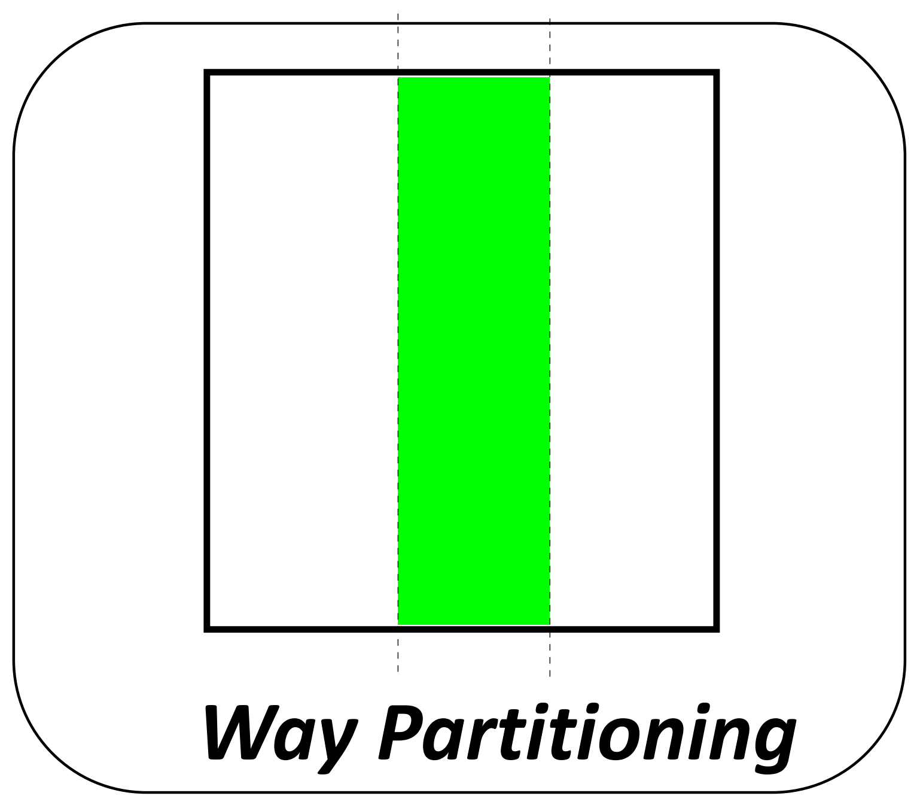
We propose SWAP, a scalable and fine-grained cache management technique that seamlessly combines set and way partitioning. By cooperatively managing cache ways and sets, SWAP (“Set and WAy Partitioning”) can successfully provide hundreds of fine-grained cache partitions for the manycore era.

SWAP requires no additional hardware beyond way parti- tioning. In fact, SWAP can be readily implemented in existing commercial servers whose processors do support hardware way partitioning. In this paper, we prototype SWAP on a 48-core Cavium ThunderX platform running Linux, and show average speedups over no cache partitioning are twice as large as those attained with hardware way partitioning alone.

Background

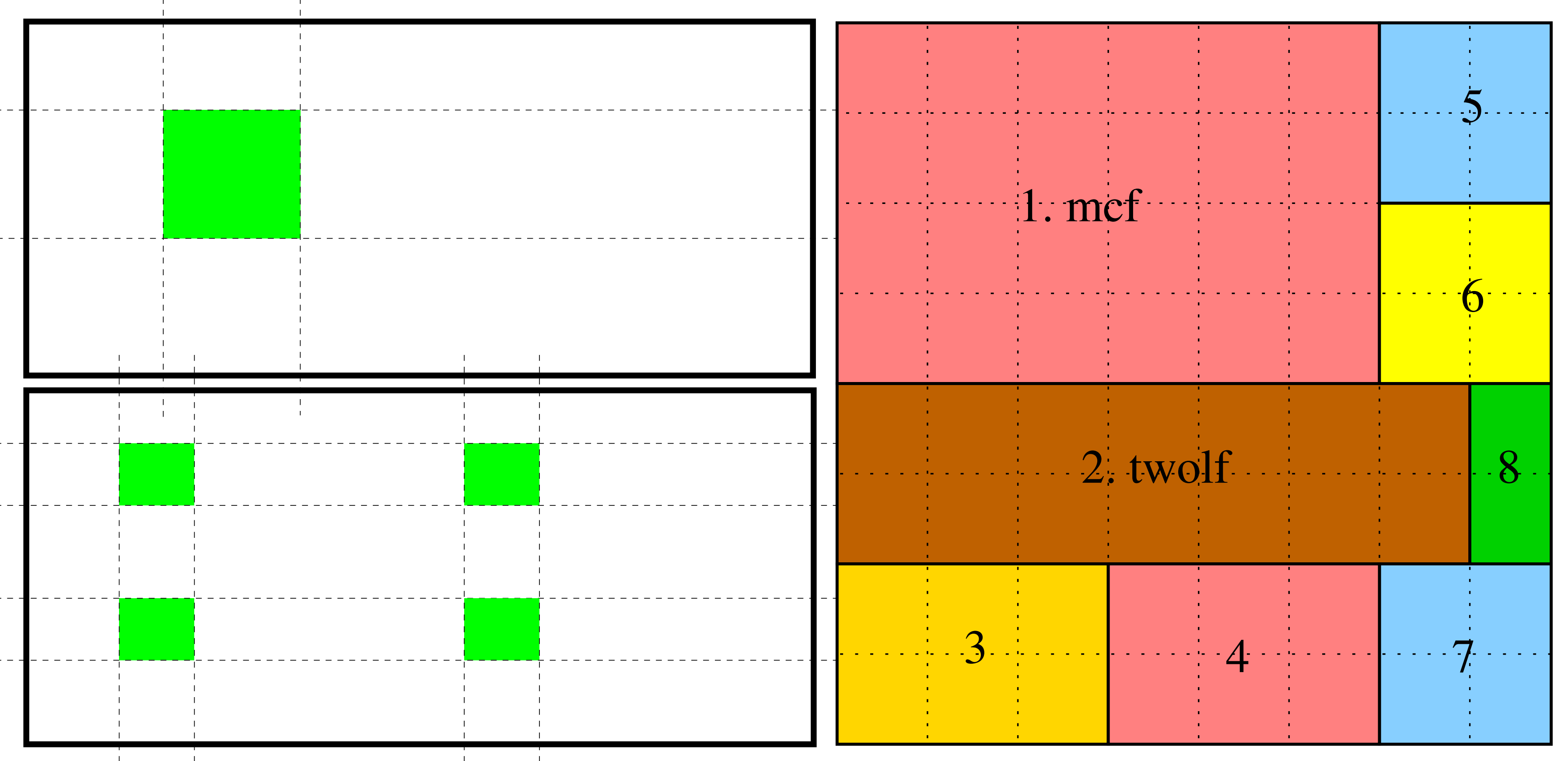






Neither set or way partitioning is fine-grained enough.

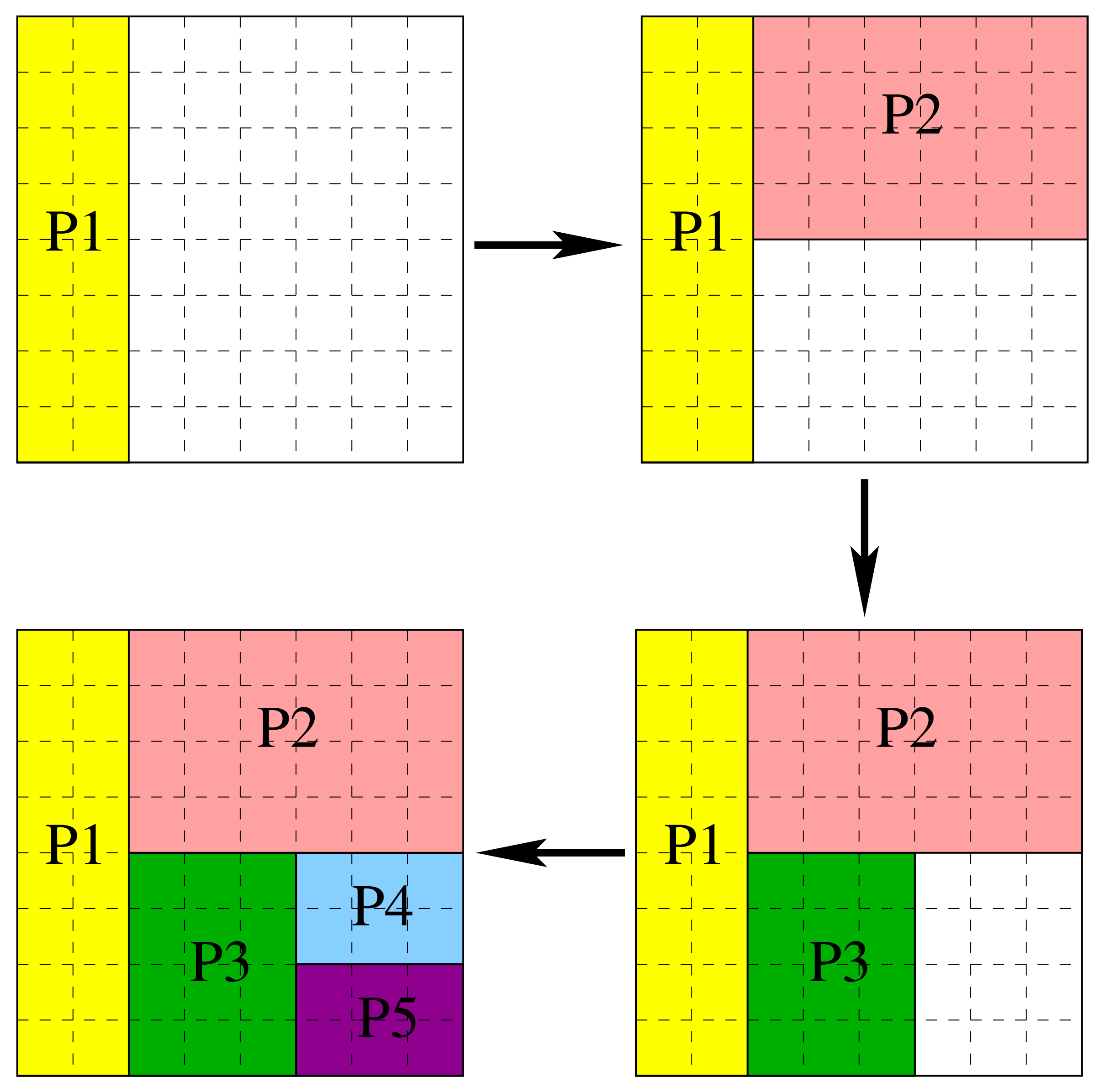
Combine Set and Way Partitioning



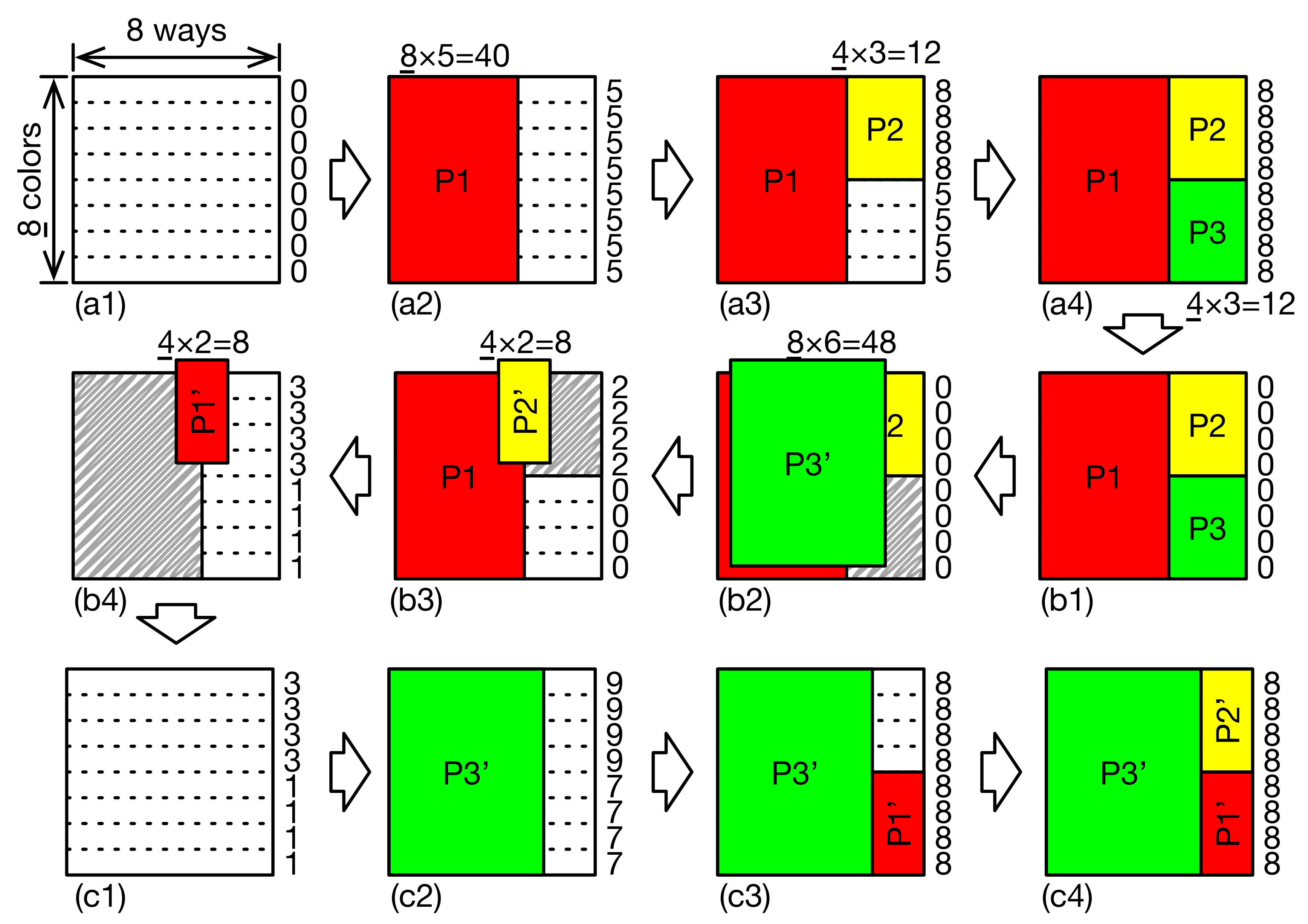
* Partition placement * Memory Pressure * Recolor overhead
* Increased conflict misses in way partitioning.

SWAP Design

Static Partitioning:
1. Offline/online profiling to get cache sensitivity curve of each app.
2. Run lookahead algorithm to find allocation size.
3. Sort and place applications by their classes as follows:



Dynamic Repartitioning:
1. ReRun lookahead algorithm to find allocation size.
2. Reset usage counters of each color.
3. Try reusing previous colors as much as possible.



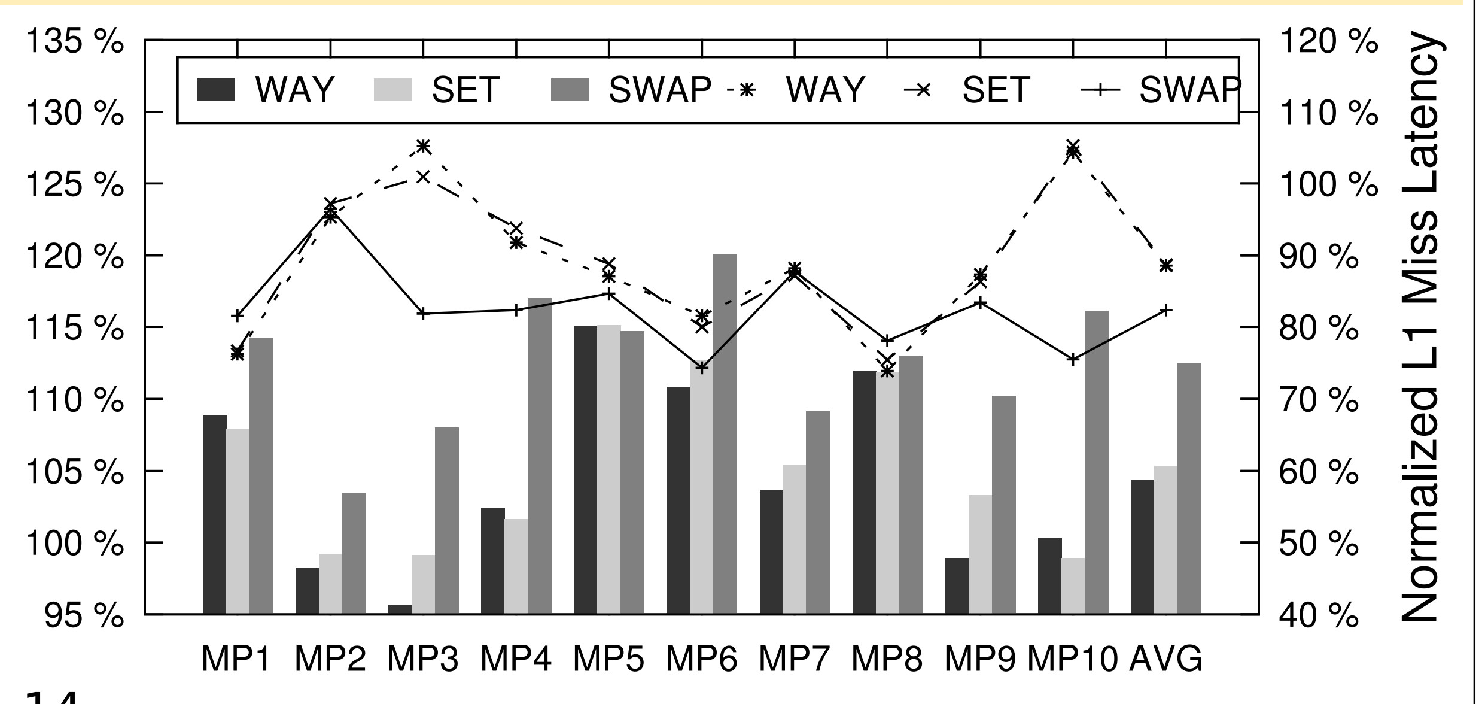
Evaluation

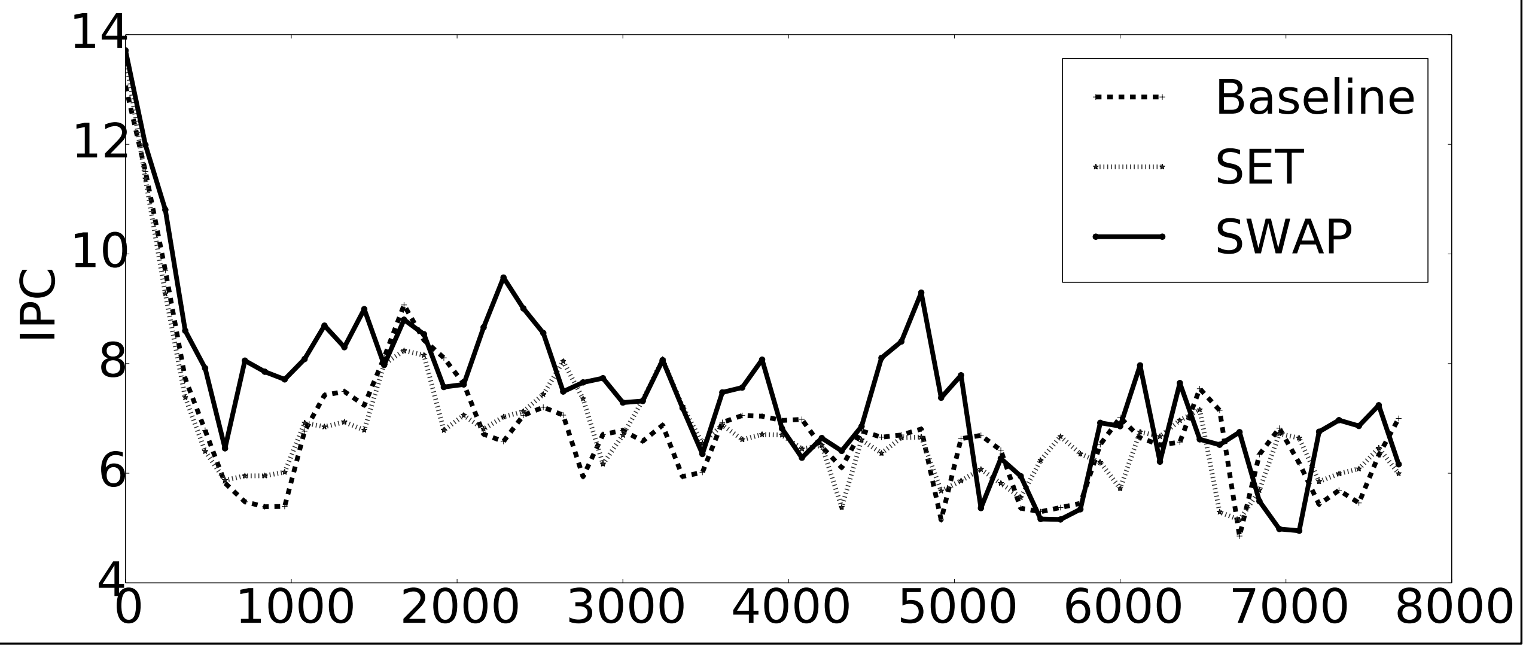
Platform: 48-core Cavium ThunderX

Results: Improve system throughput by 12-14.5%.

SWAP overhead: negligible

More potentials: guarantee QoS and improve


WAY
SET
SWAP
WAY
SET
SWAP
MP1
MP2
MP3
MP4
MP5
MP6
MP7
MP8
MP9
MP10
AVG
Normalized L1 Miss Latency


IPC
Baseline
SET
SWAP