THE DYNAMIC GRANULARITY MEMORY SYSTEM

Doe Hyun Yoon

IIL, HP Labs

Michael Sullivan
Min Kyu Jeong
Mattan Erez

ECE, UT Austin

MEMORY ACCESS GRANULARITY

- The size of block for accessing main memory
  - Often, equal to last-level cache line size

- Modern systems use **coarse-grained (CG)** memory access
  - 64B or larger
  - Amortize control & ECC overhead
  - Prefetching

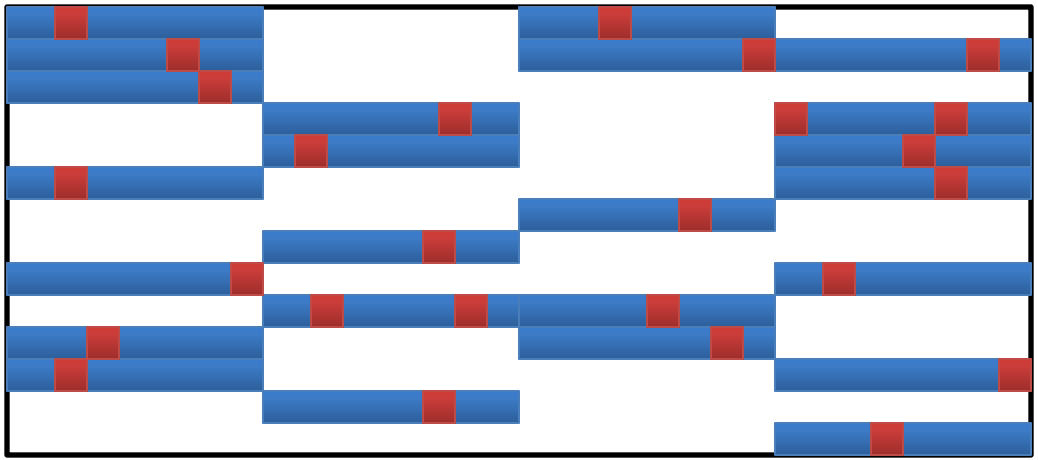
CG ACCESS MAY WASTE BW

GUPS microbenchmark

```
for( i=0; i<N; i++ ) {
  a[ b[i] ] += x;
}
```

Initialized with random numbers

Buffer a



- **Waste BW on unused data**

CAN WE WASTE BW?

- CG access often improves performance
  - Large cache lines reduce miss rate due to prefetching
- Off-chip BW doesn't scale with # cores
- Power is the limiting factor
- **We shouldn't waste the finite off-chip BW**

HOW TO EFFICIENTLY UTILIZE OFF-CHIP BW?

- Prior work: AGMS [ISCA'11]
  - Combine CG and FG accesses
  - **Need SW help for ECC support**
    - Source code, compiler, OS, virtual memory, …

- **DGMS**
  - **HW-only** variant of AGMS
  - **Truly dynamic** granularity adaptation

ADAPTIVE GRANULARITY MEMORY SYSTEM [ISCA'11]

AGMS [ISCA'11]

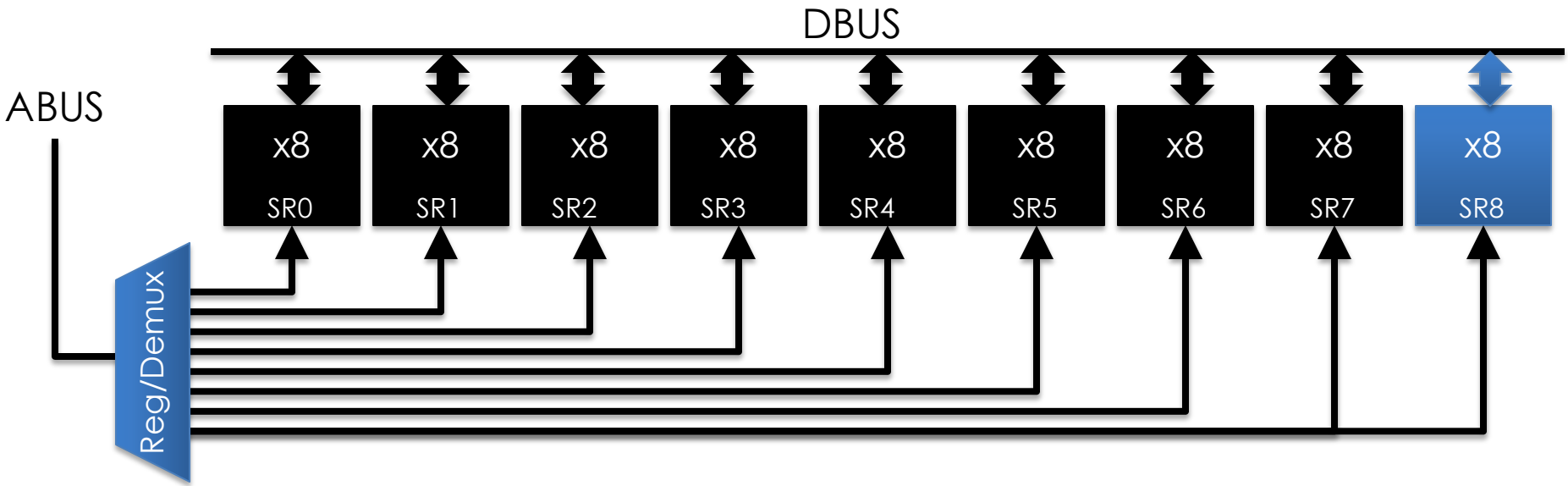
- Combine coarse-grained (CG) and fine-grained (FG) accesses

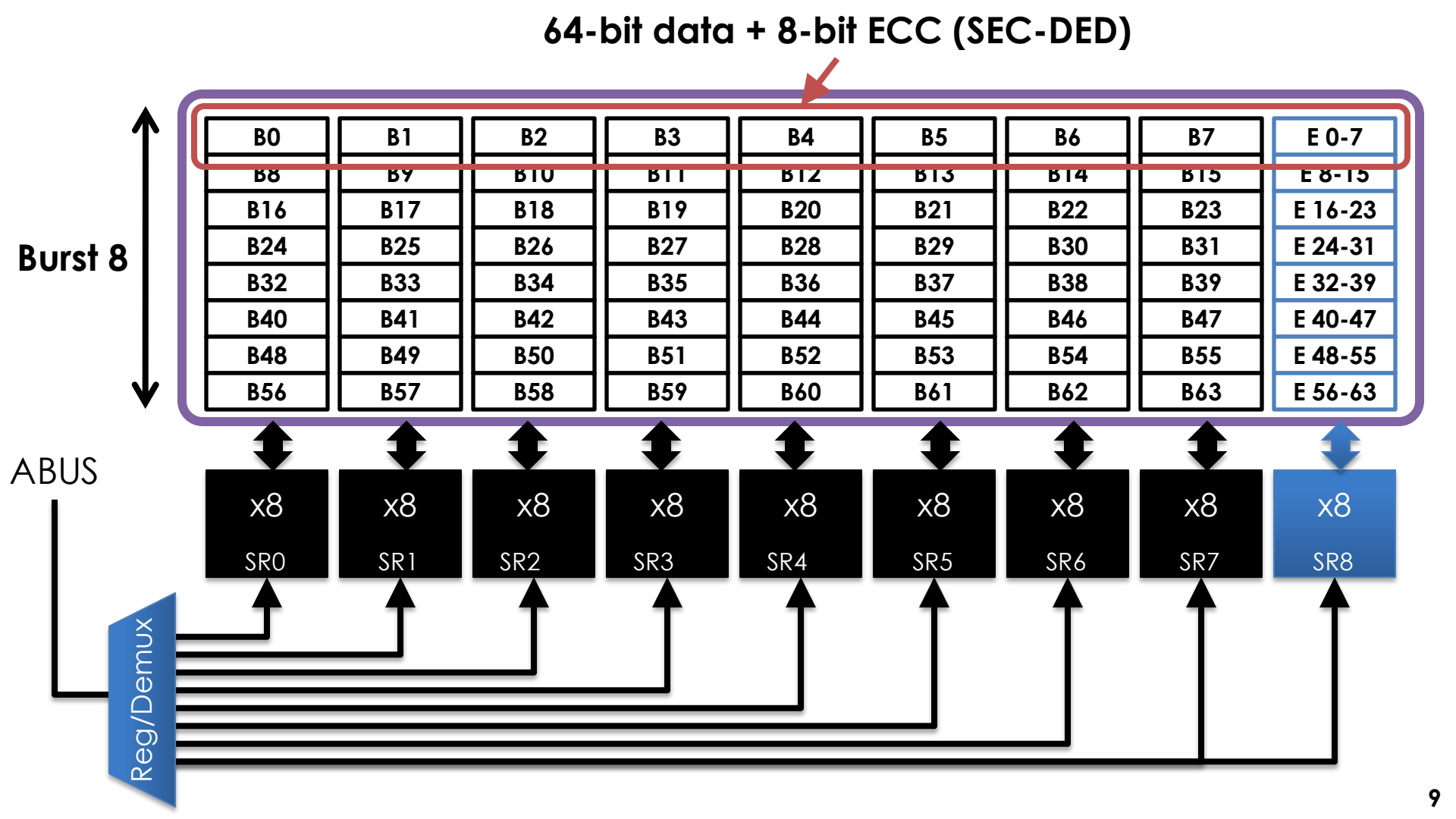
- **CG for high spatial locality regions**
- **FG for low spatial locality regions**

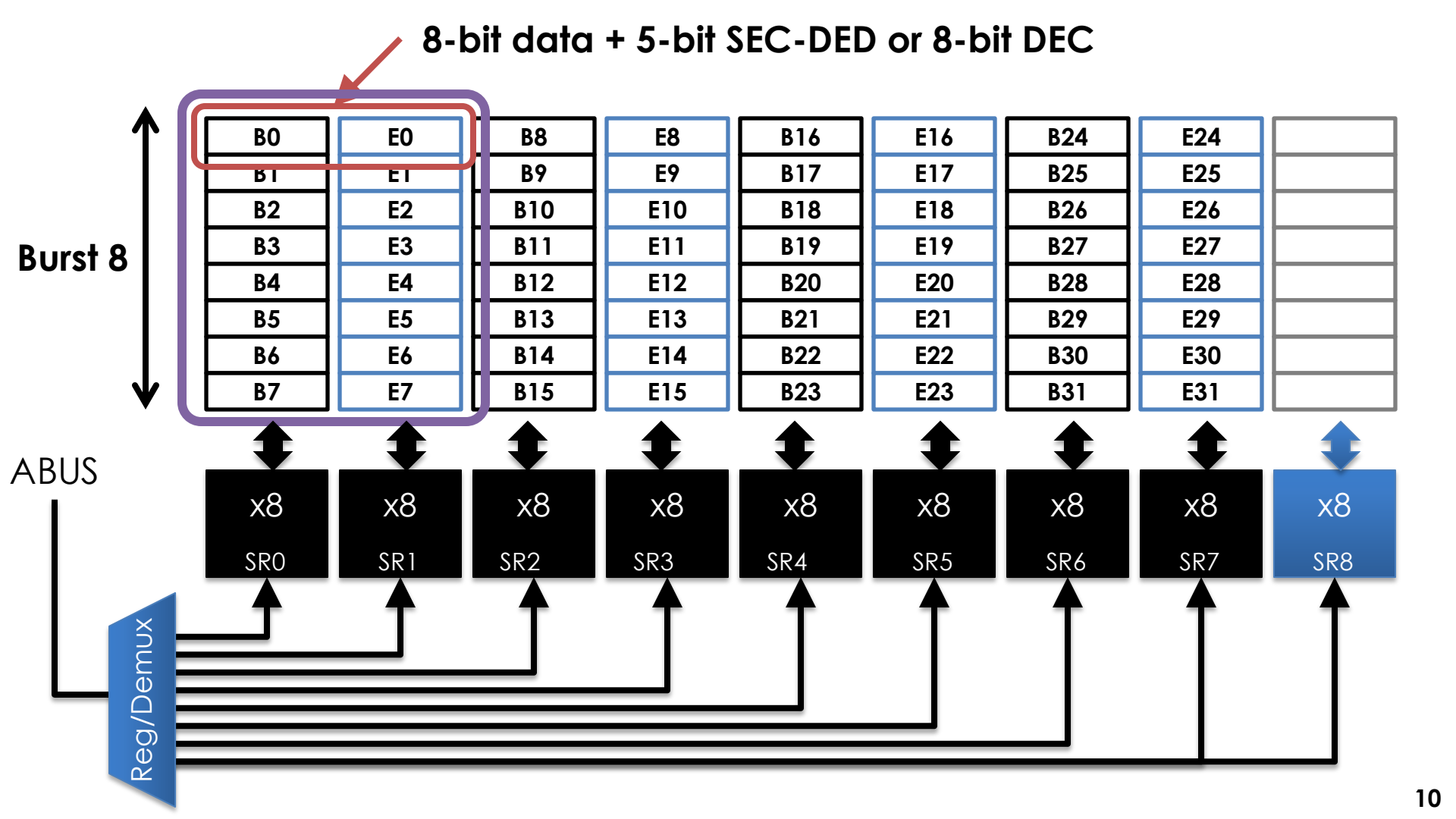
- Higher throughput
- Lower DRAM power

SUB-RANKED DRAM MODULE

- **Independently control individual DRAM chips**
- Access granularity = 8bit x burst 8 = 8B




64-bit data + 8-bit ECC (SEC-DED)
Burst 8
B0 B1 B2 B3 B4 B5 B6 B7 E 0-7
B8 B9 B10 B11 B12 B13 B14 B15 E 8-15
B16 B17 B18 B19 B20 B21 B22 B23 E 16-23
B24 B25 B26 B27 B28 B29 B30 B31 E 24-31
B32 B33 B34 B35 B36 B37 B38 B39 E 32-39
B40 B41 B42 B43 B44 B45 B46 B47 E 40-47
B48 B49 B50 B51 B52 B53 B54 B55 E 48-55
B56 B57 B58 B59 B60 B61 B62 B63 E 56-63
ABUS
x8 SR0
x8 SR1
x8 SR2
x8 SR3
x8 SR4
x8 SR5
x8 SR6
x8 SR7
x8 SR8
Reg/Demux


8-bit data + 5-bit SEC-DED or 8-bit DEC
Burst 8
B0 B1 B2 B3 B4 B5 B6 B7
E0 E1 E2 E3 E4 E5 E6 E7
B8 B9 B10 B11 B12 B13 B14 B15
E8 E9 E10 E11 E12 E13 E14 E15
B16 B17 B18 B19 B20 B21 B22 B23
E16 E17 E18 E19 E20 E21 E22 E23
B24 B25 B26 B27 B28 B29 B30 B31
E24 E25 E26 E27 E28 E29 E30 E31
ABUS
x8 SR0
x8 SR1
x8 SR2
x8 SR3
x8 SR4
x8 SR5
x8 SR6
x8 SR7
x8 SR8
Reg/Demux

SOFTWARE SUPPORT IN AGMS

- **Different data/ECC layouts** for CG & FG

- Requires software help
  - Extend virtual memory interface
  - OS/runtime manages CG&FG pages
  - Programmer/compiler annotates preferred granularity

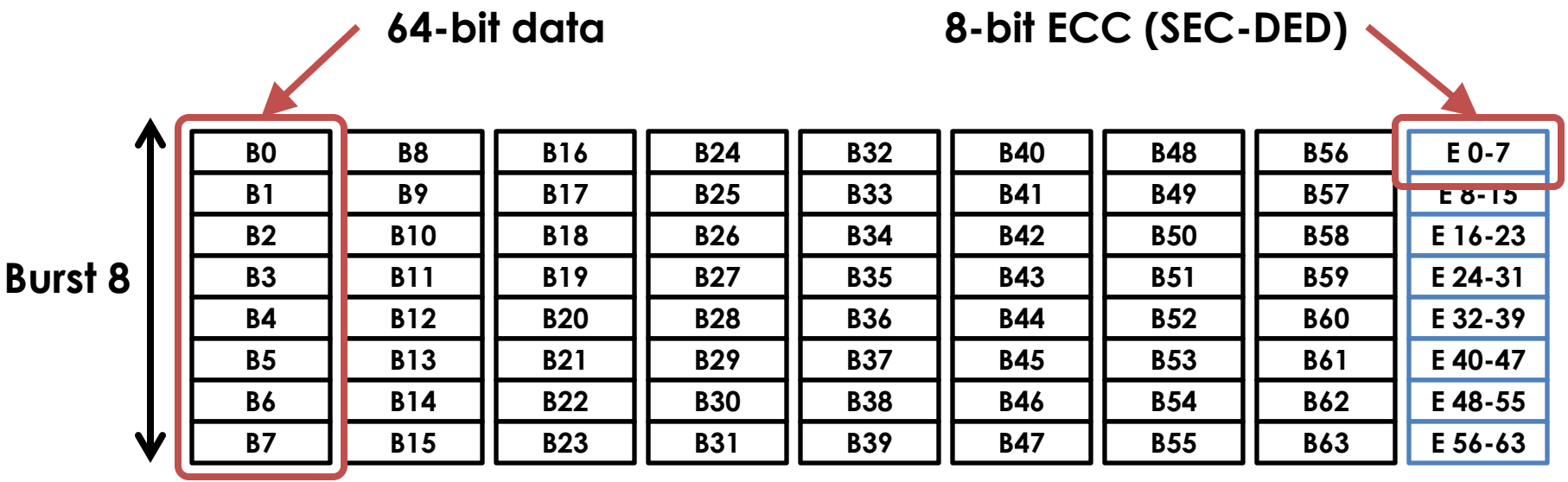
- **Need to change every level of system hierarchy!**

DYNAMIC GRANULARITY MEMORY SYSTEM

DGMS

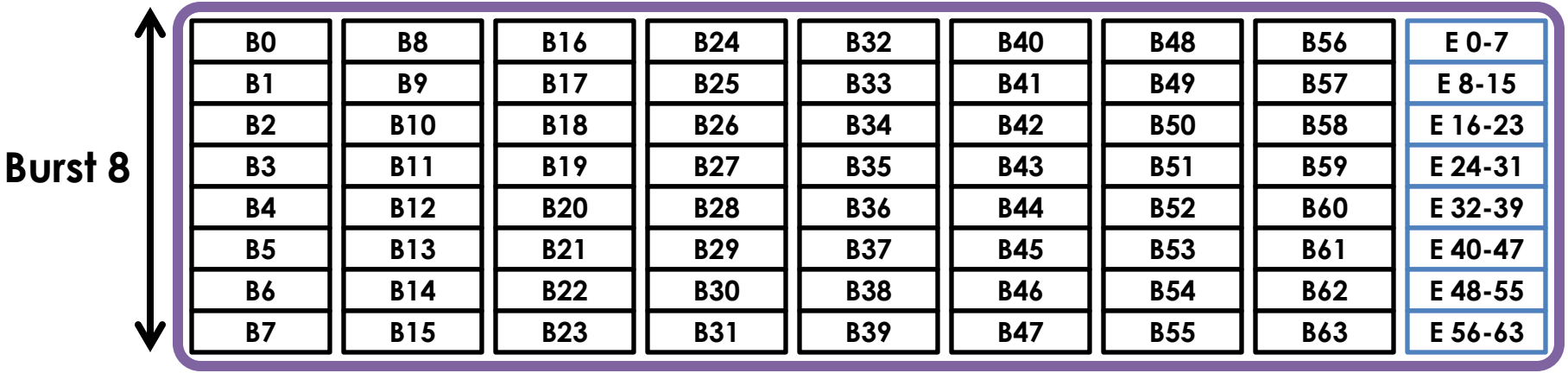
- **Unified data/ECC layout** for CG & FG
  - No SW support
- HW-only variant of AGMS
  - Comparable or better performance
  - **Easier to implement**
- Challenge:
  - **How to predict access granularity dynamically?**

UNIFIED DATA/ECC LAYOUT



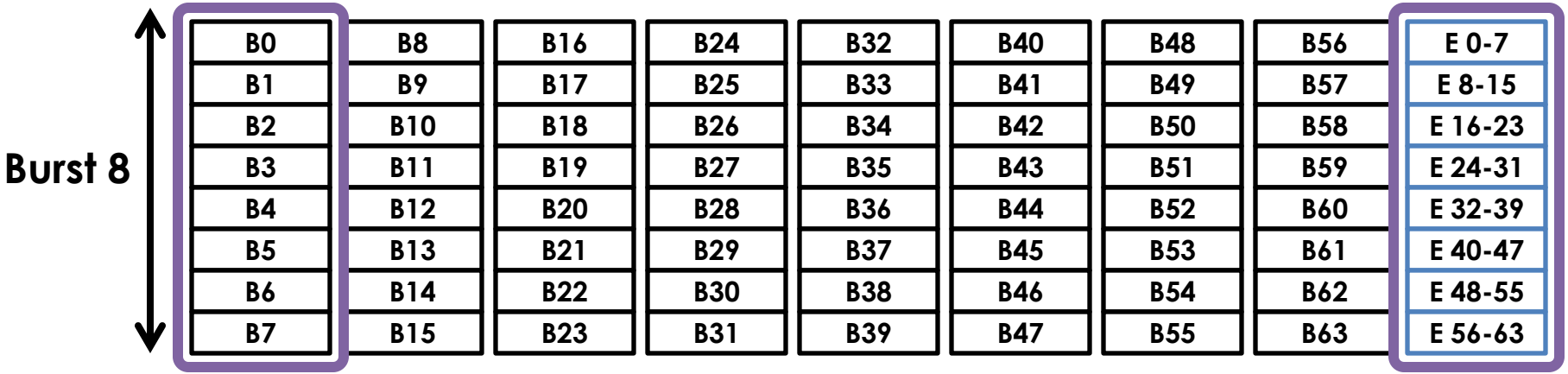
CG ACCESS

- Access the whole 72B

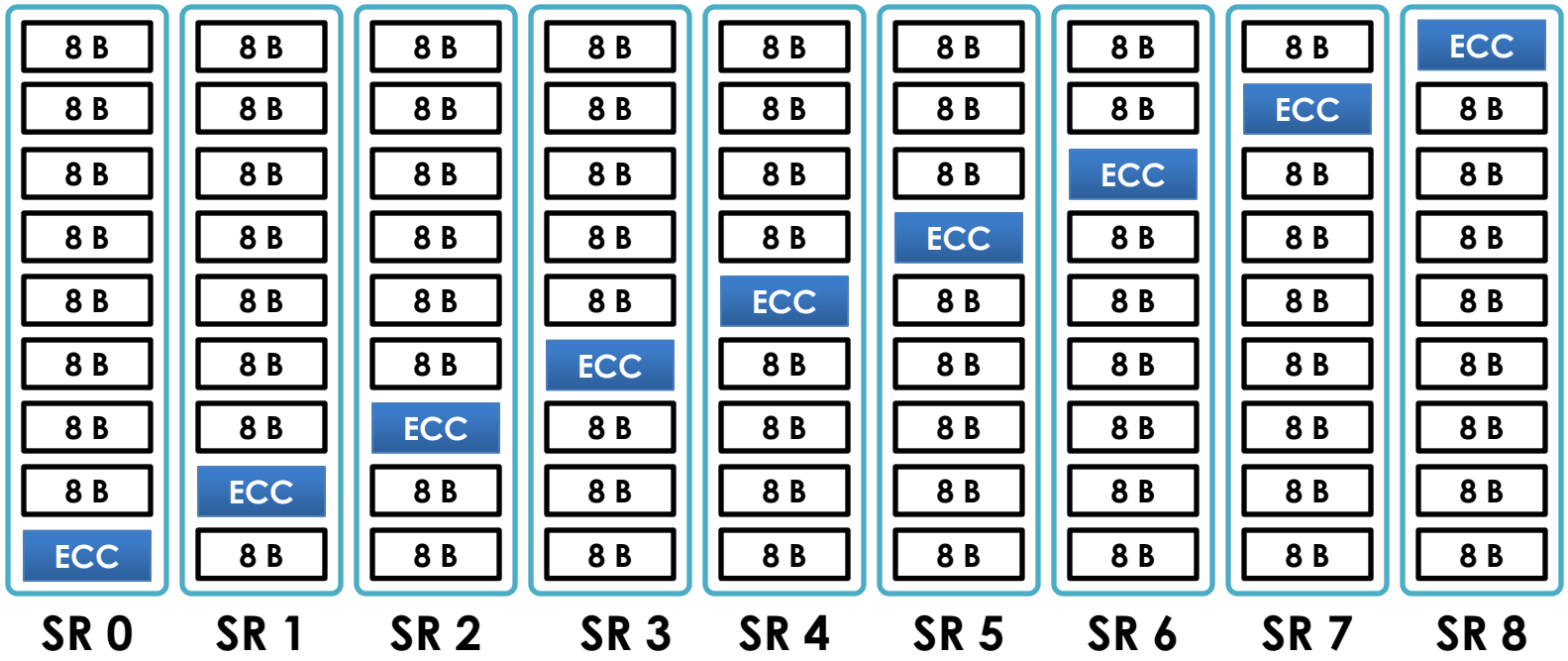


FG ACCESS

- Access 8B data and 8B ECC

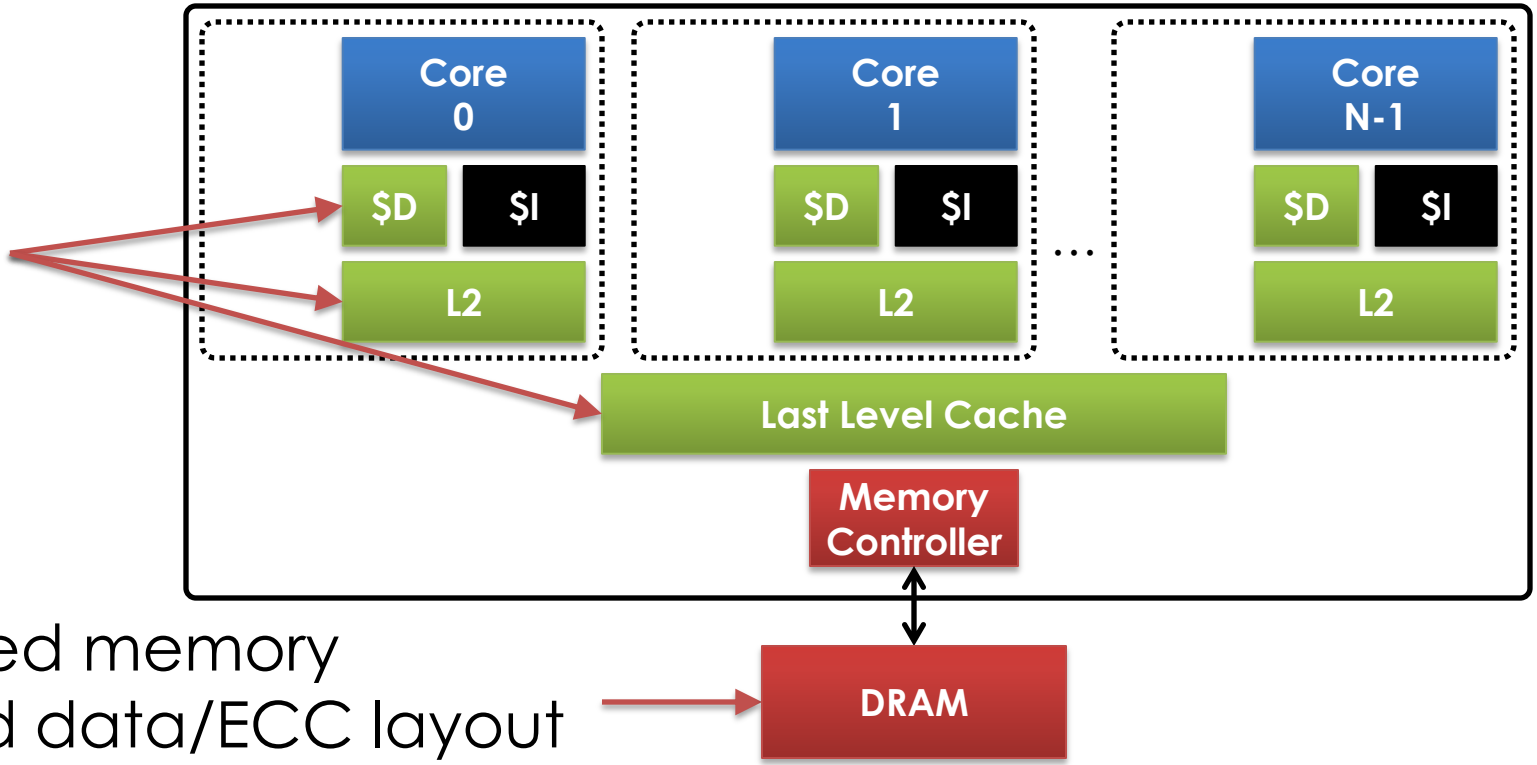


AVOIDING CONTENTION ON ECC DRAM



DGMS DESIGN

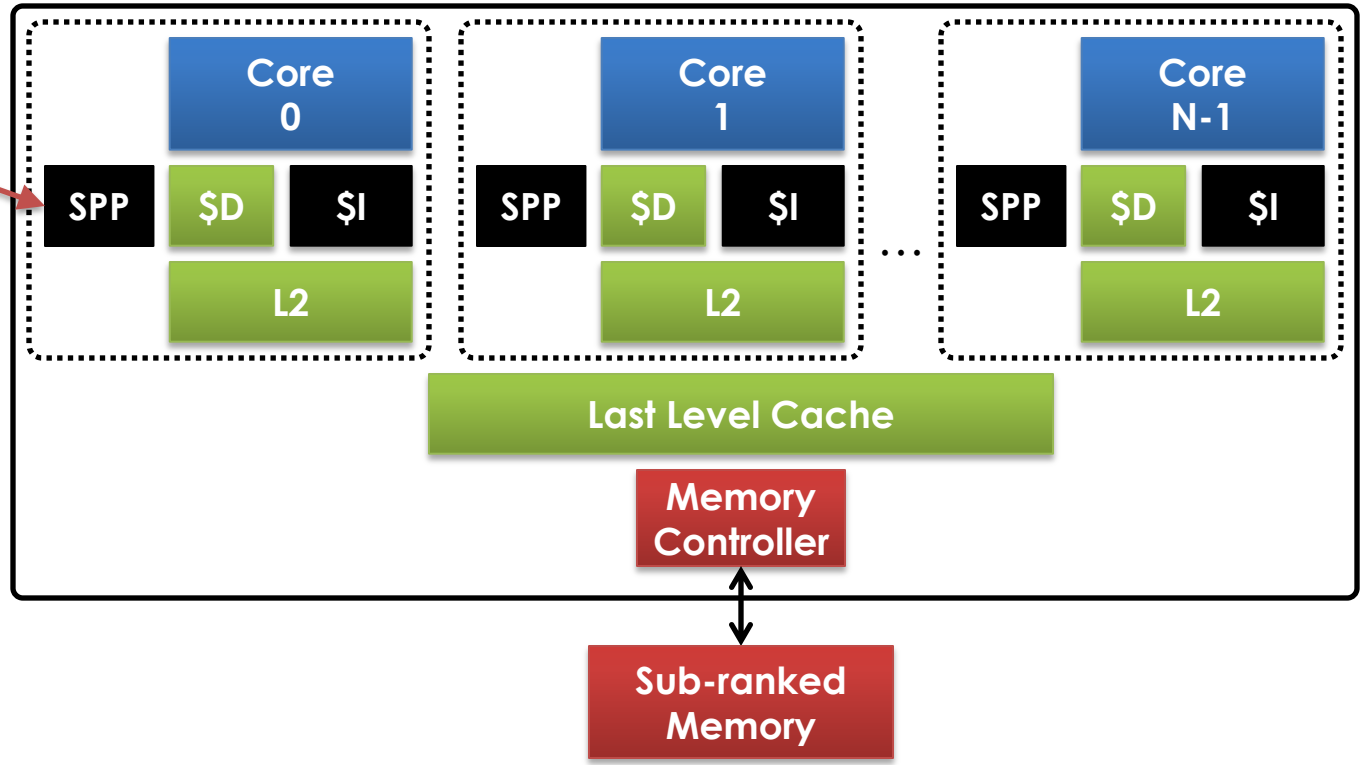
Sector cache



Sub-ranked memory w/ unified data/ECC layout

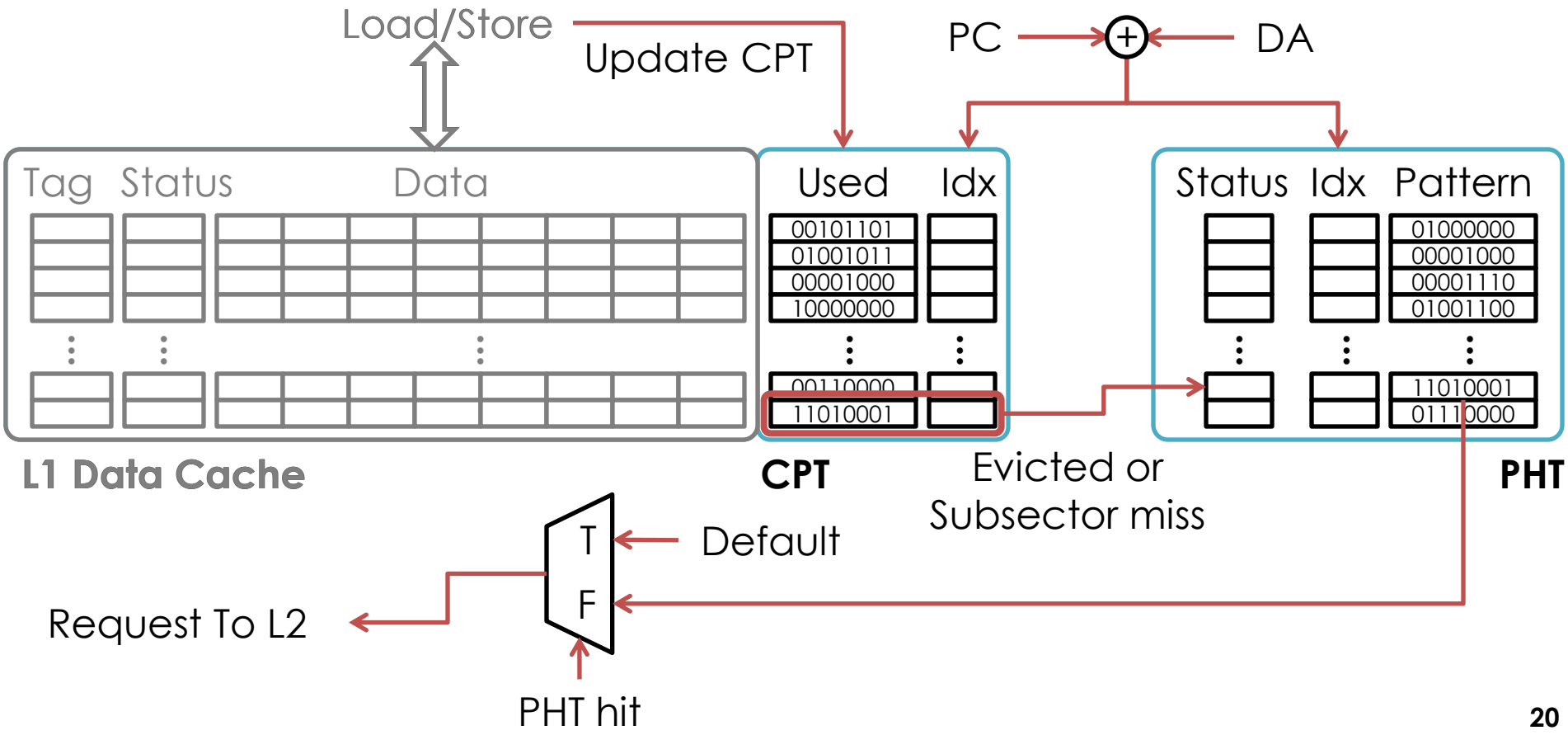
GRANULARITY PREDICTION

Spatial Pattern Predictor
[Chen; HPCA'04]

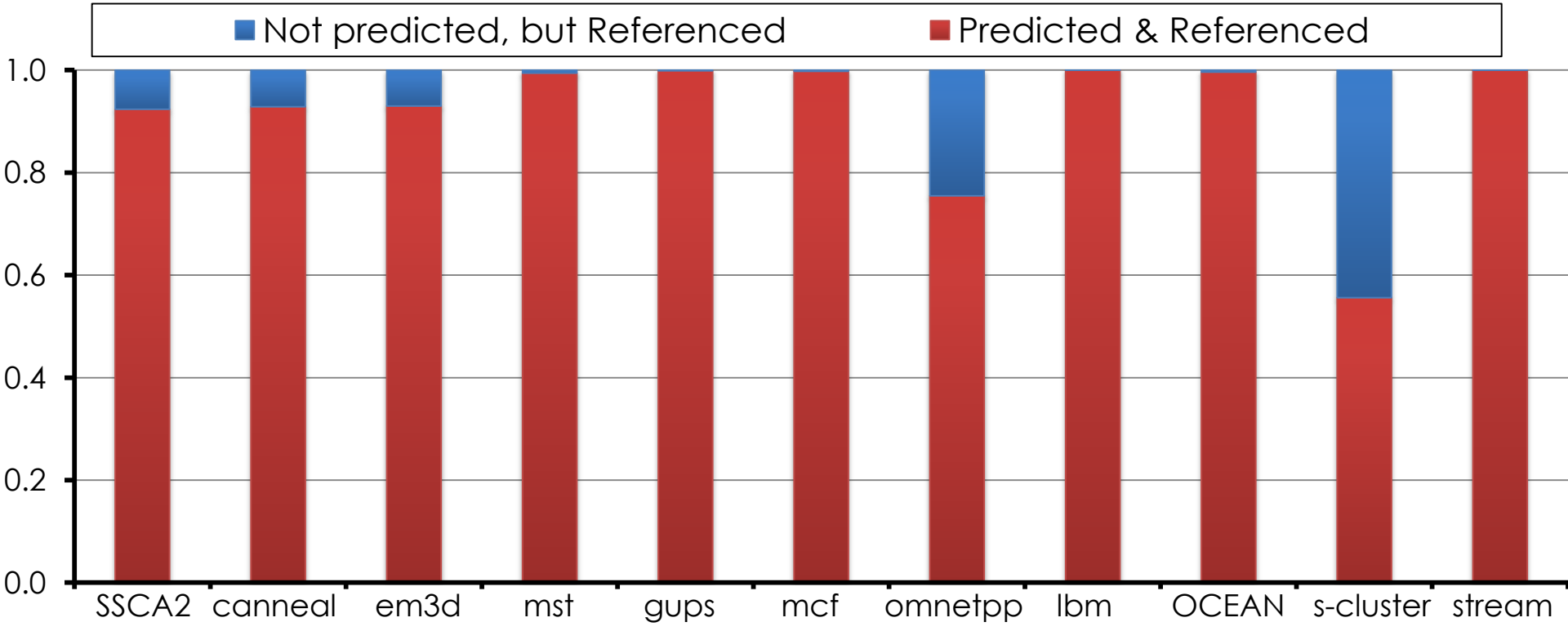


Which words within a cache line will be used?

SPATIAL PATTERN PREDICTOR [CHEN; HPCA'04]



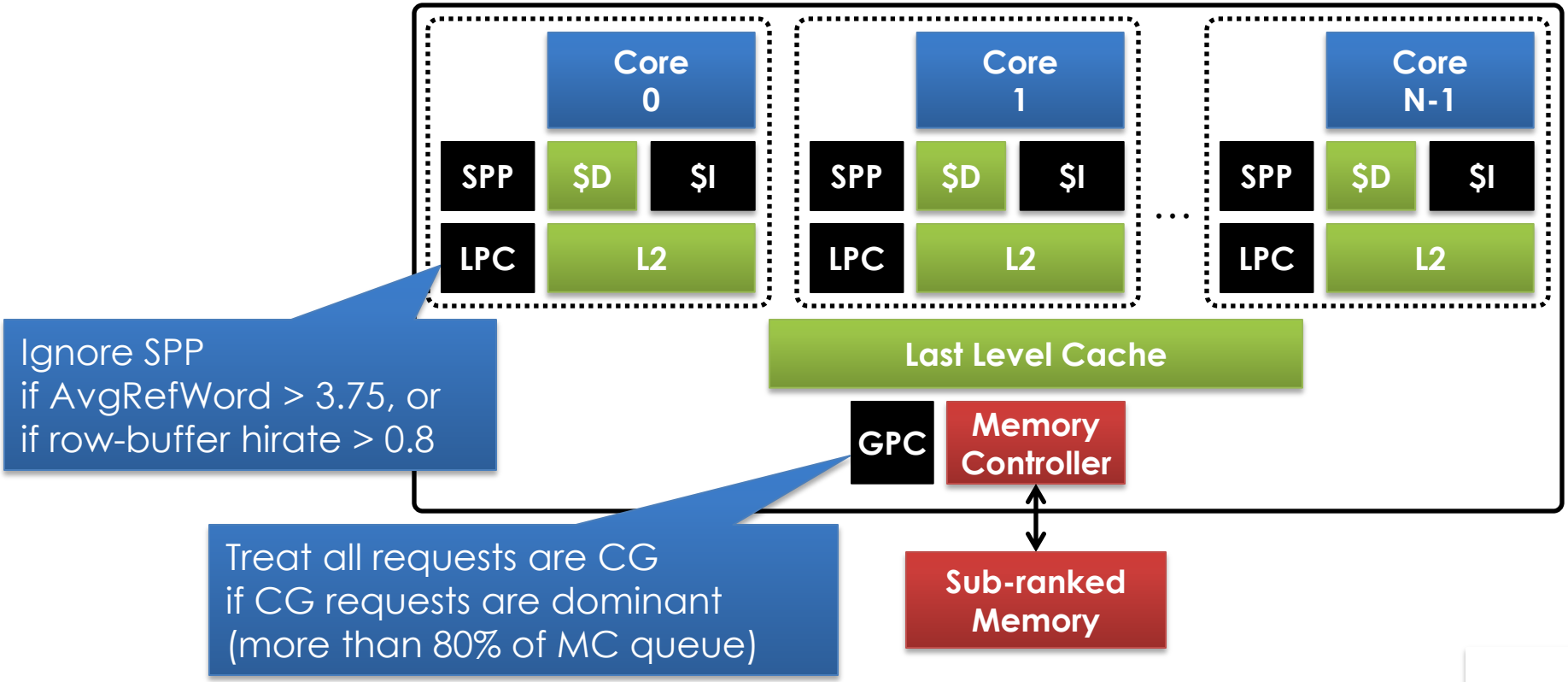
SPP ACCURACY



SPP LIMITATIONS

- Case 1)
  - Application accesses 5~7 words per cache line
- Case 2)
  - App1 has low spatial locality, MPKI is 1
  - App2 has high spatial locality, MPKI is 20
- Minimizing traffic doesn't always improve performance

PREDICTION CONTROLLER



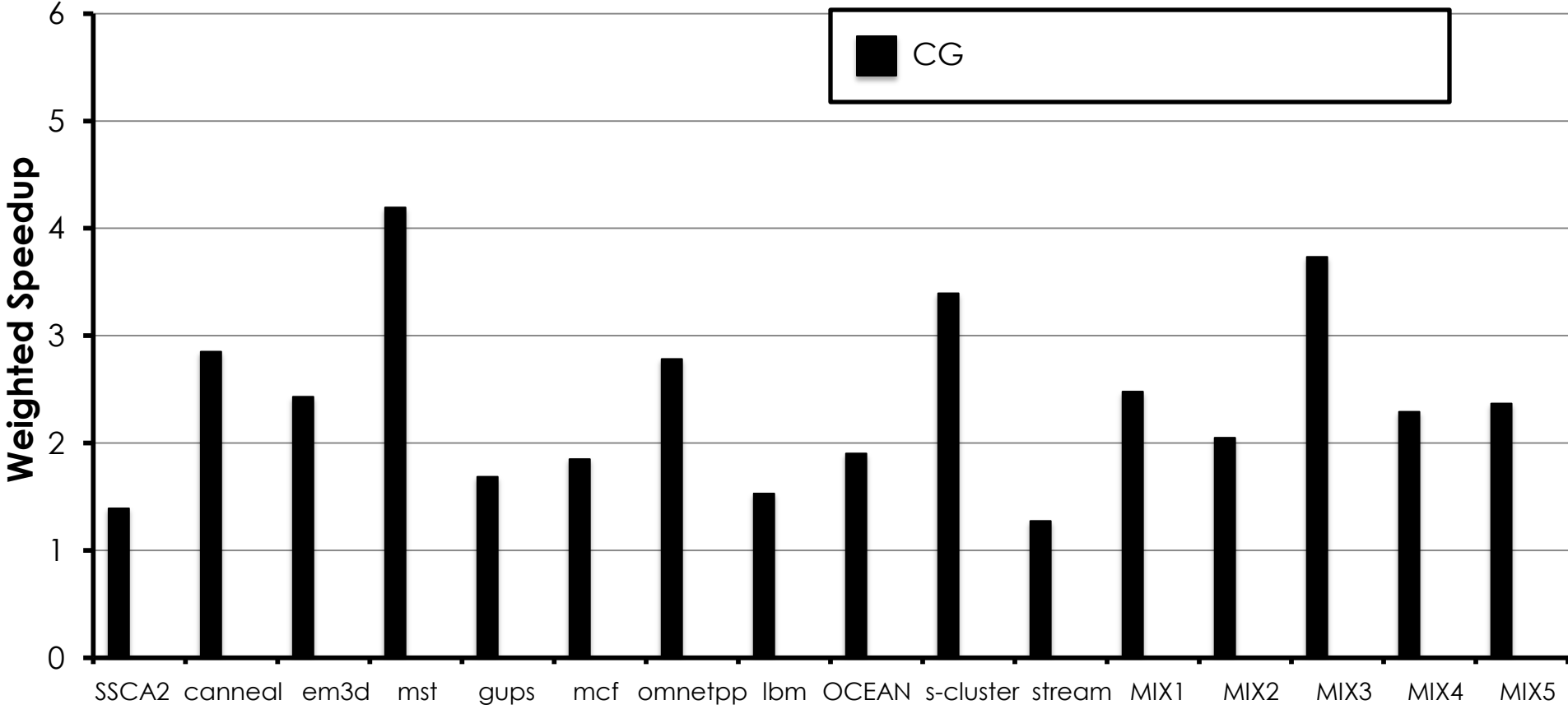
LPC & GPC prevent performance degradation in some CG-friendly apps

EVALUATION

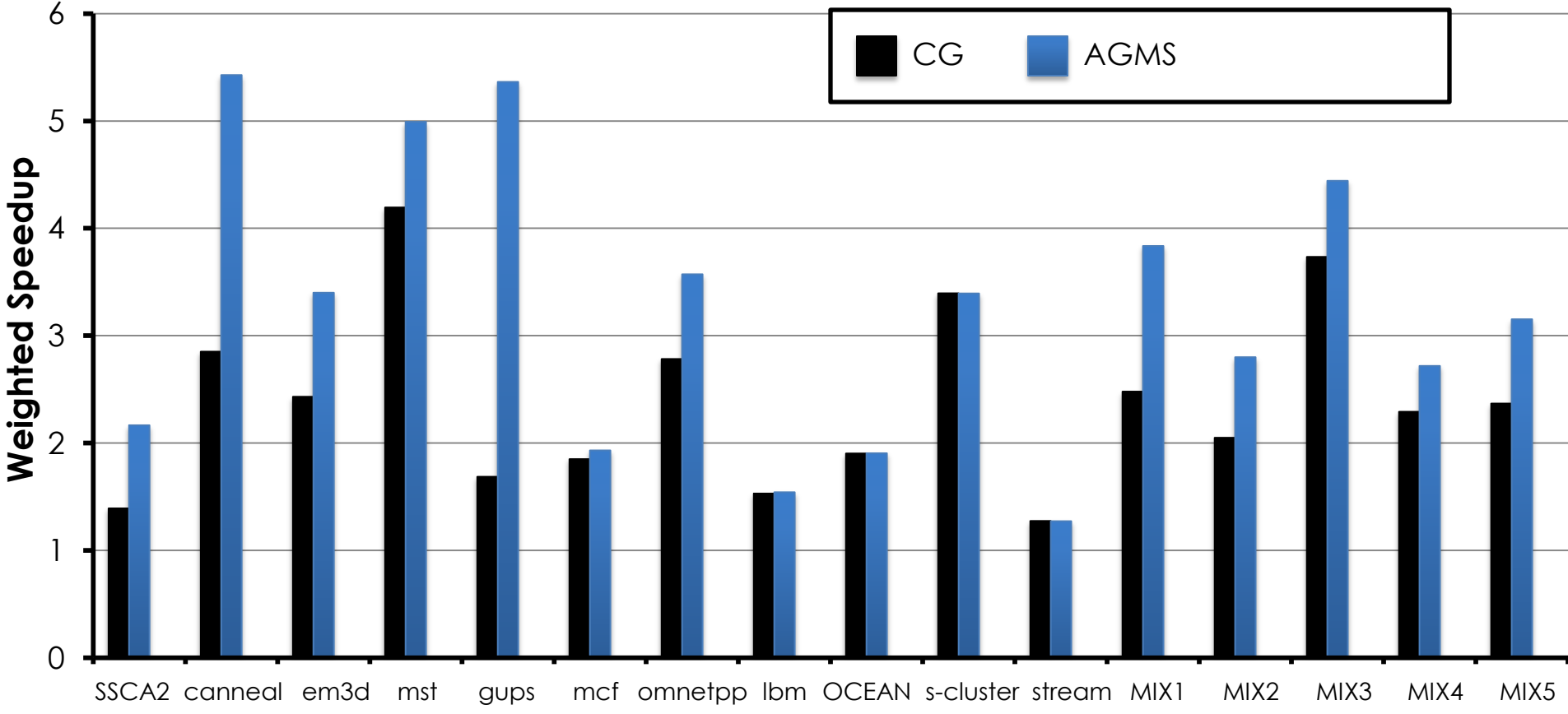
EVALUATION

- Zesto simulator
  - 8 out-of-order x86 cores
  - Private caches: 32kB I/D L1, 256kB unified L2
  - Shared last-level cache: 8MB
- DrSim: detailed DDR3 DRAM model
- Memory intensive multiprogrammed workloads

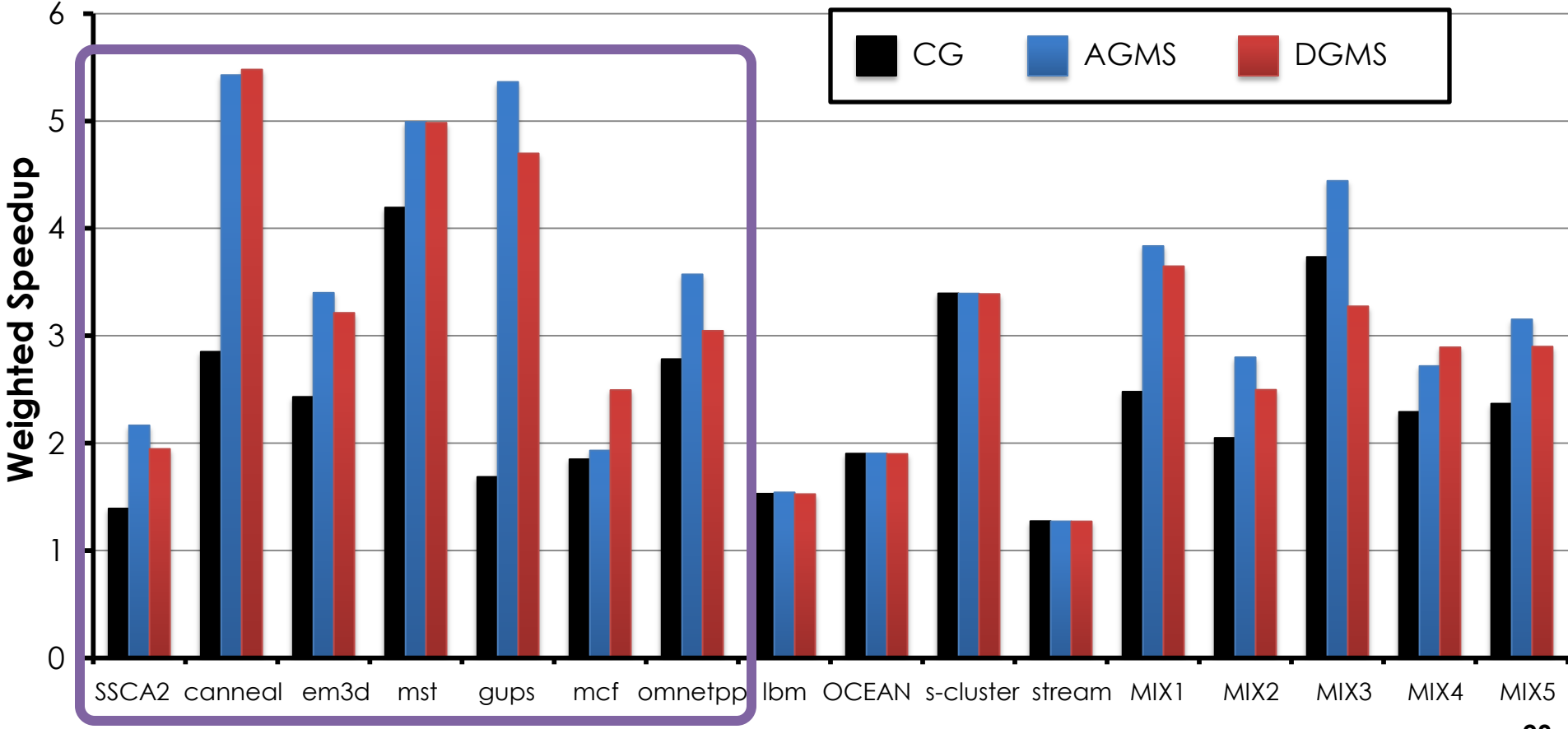
SYSTEM THROUGHPUT



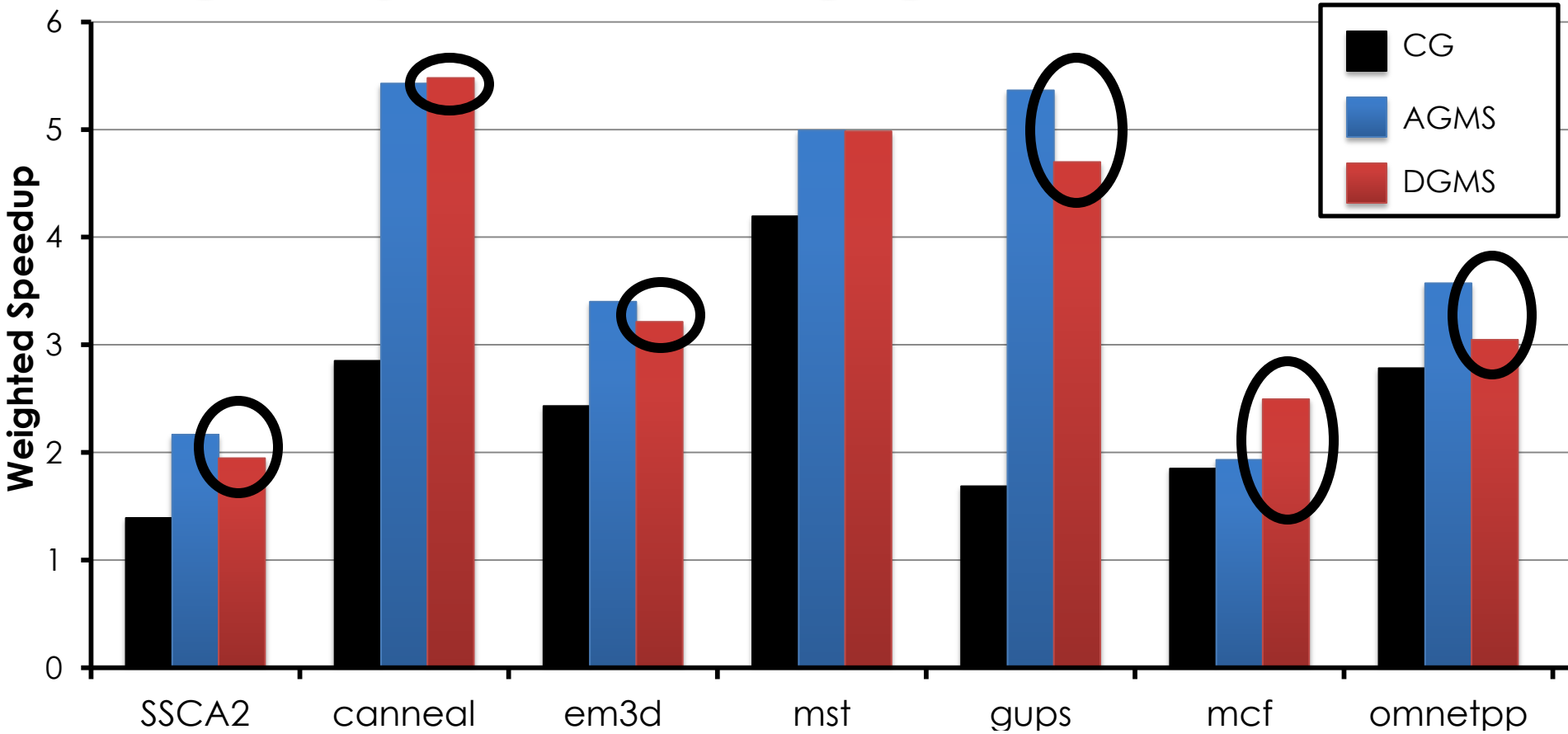
SYSTEM THROUGHPUT



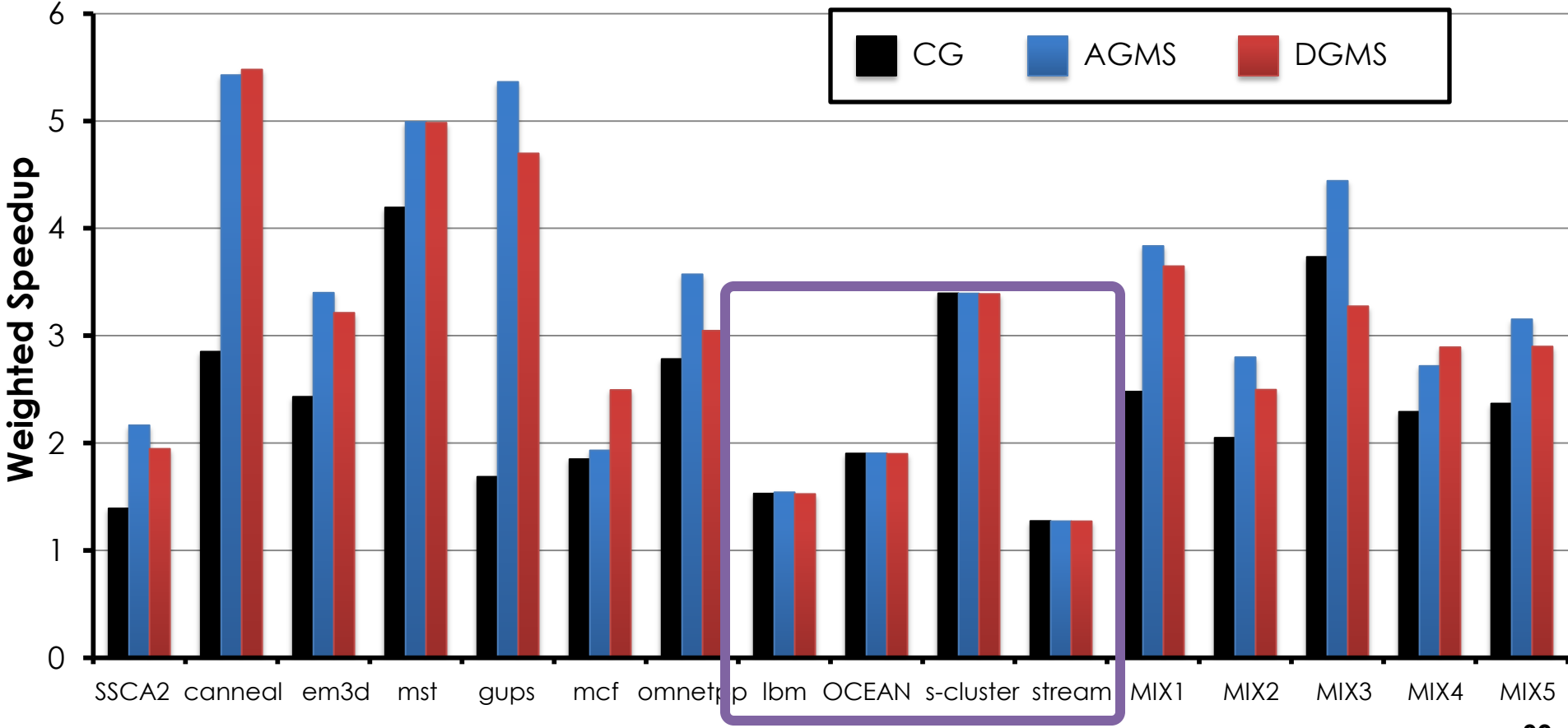
SYSTEM THROUGHPUT



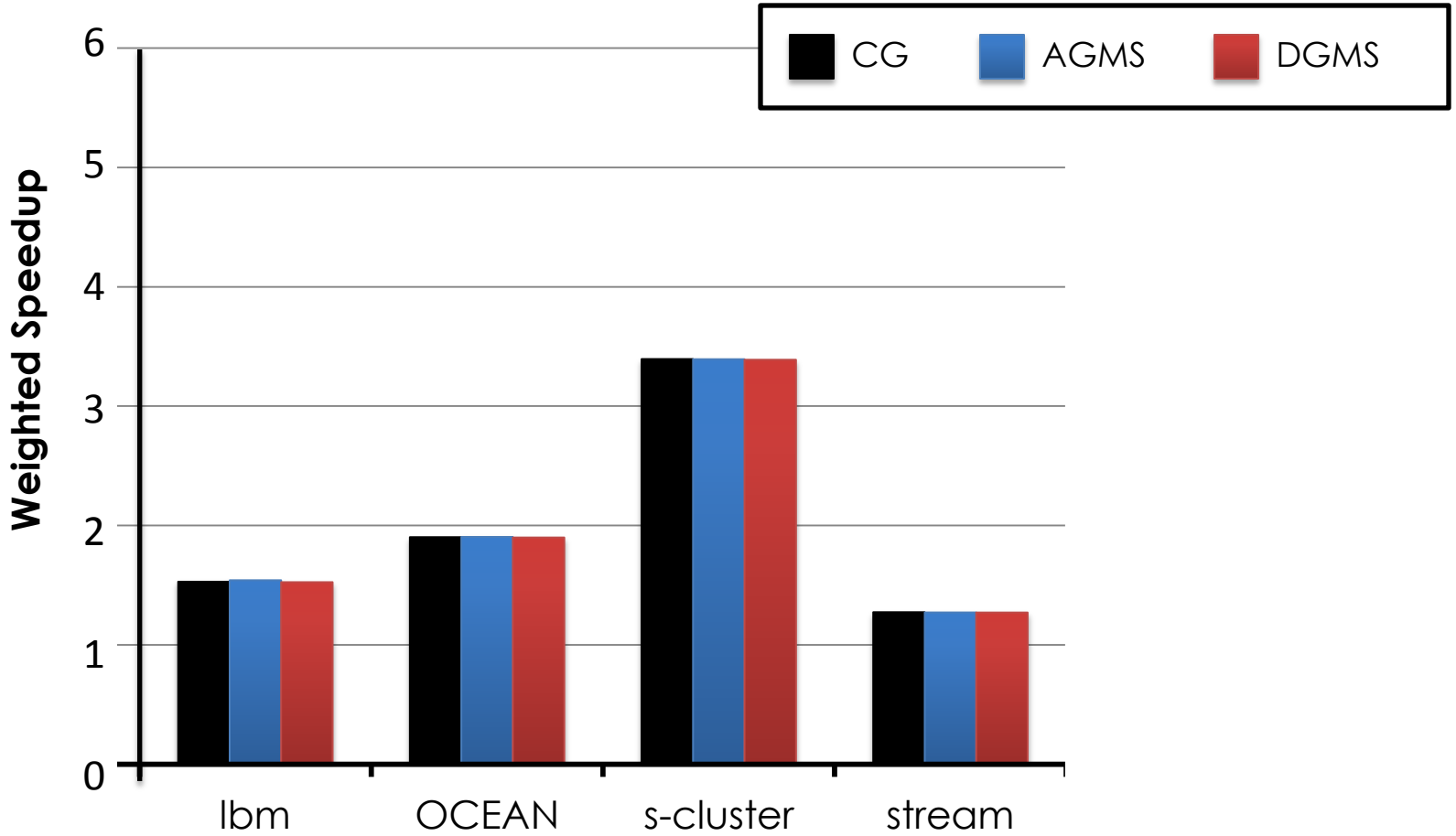
LOW SPATIAL LOCALITY APPS



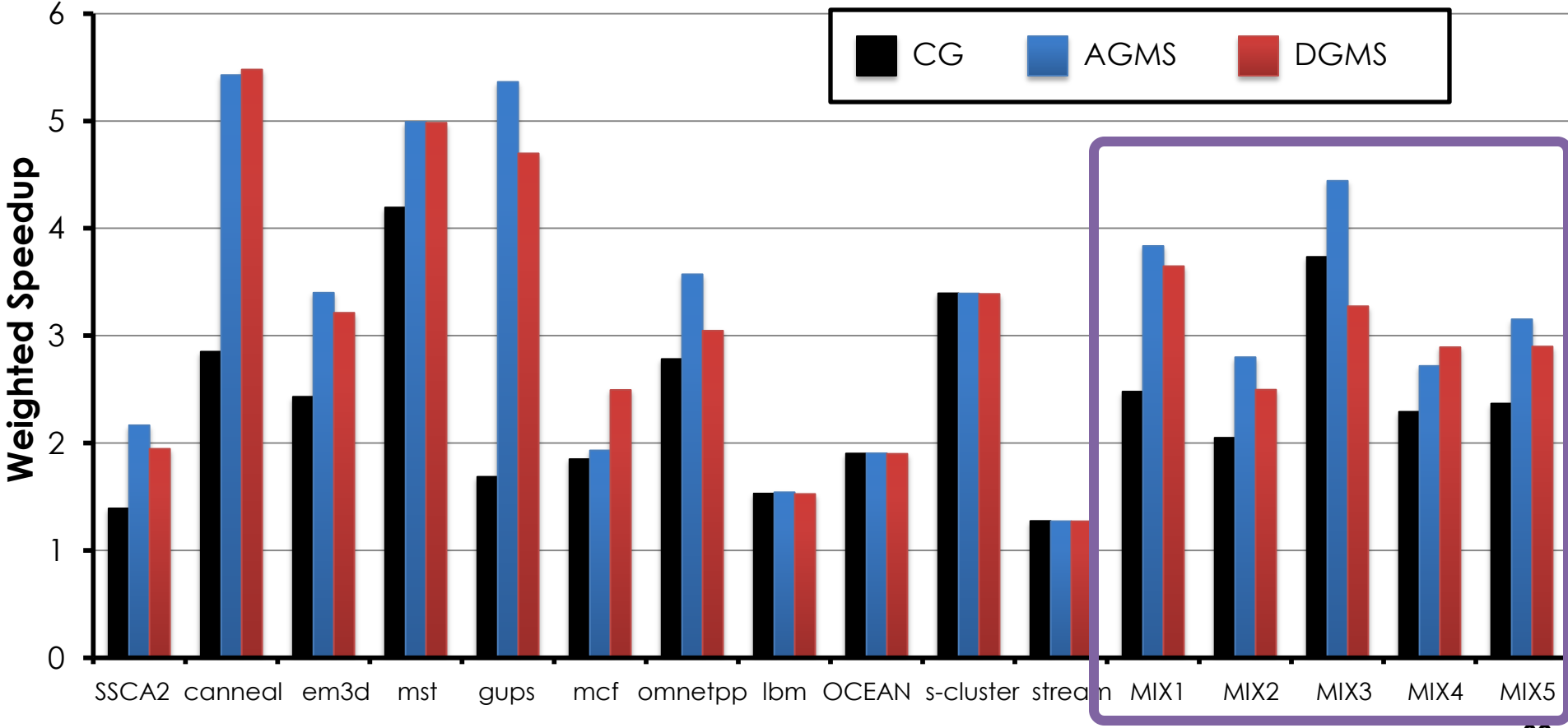
SYSTEM THROUGHPUT



HIGH SPATIAL LOCALITY APPS

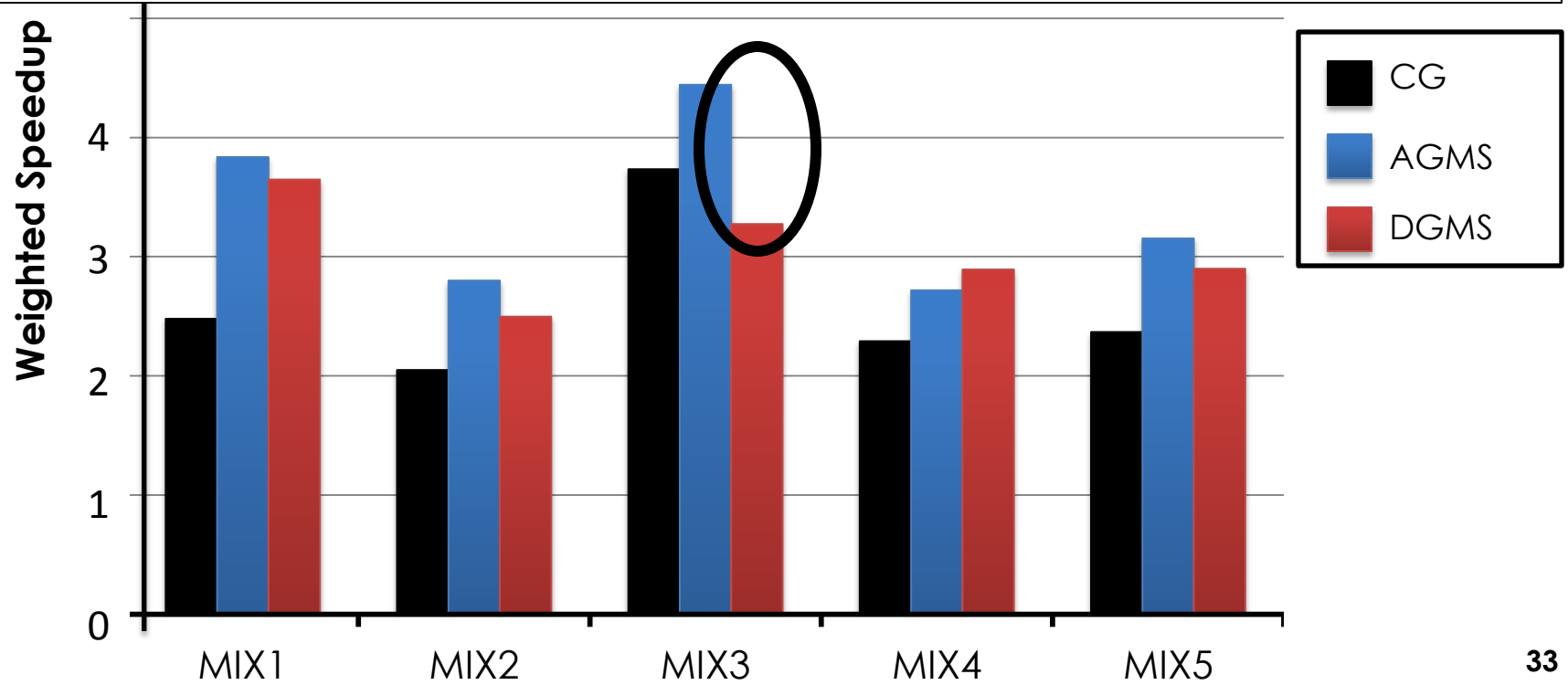


SYSTEM THROUGHPUT

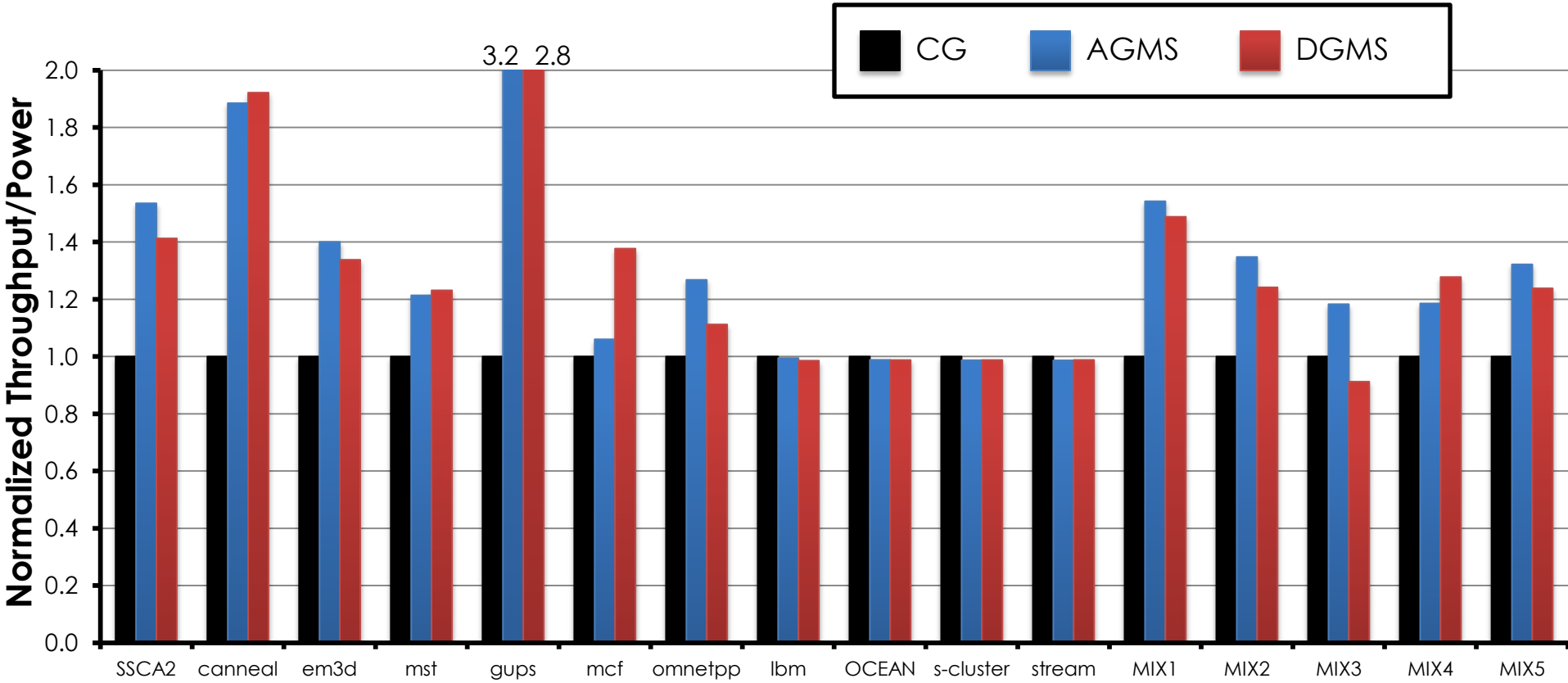


MIXED CASES

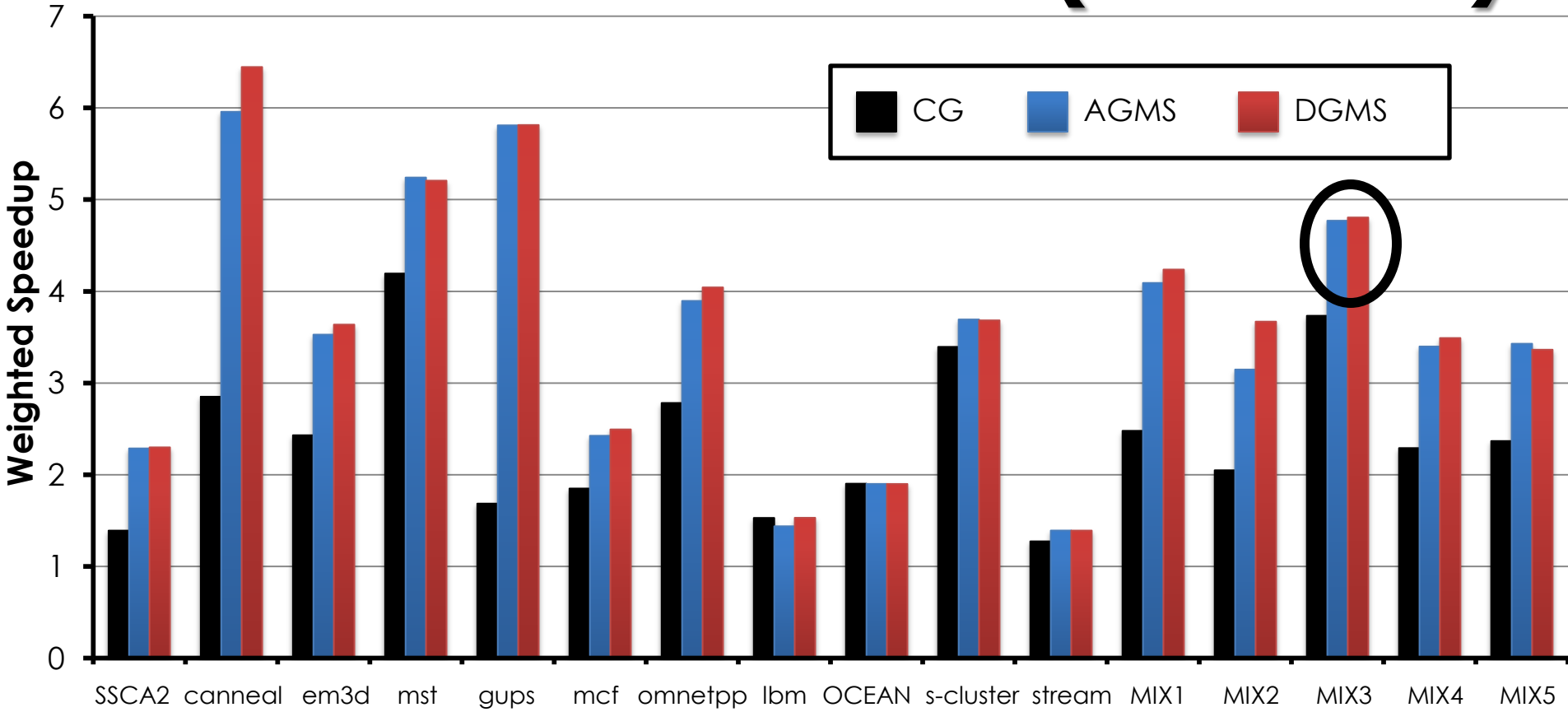
MIX1: SSCA2 x2, mst x2, em3d x2, canneal x2
MIX2: SSCA2 x2, canneal x2, mcf x2, OCEAN x2
MIX3: canneal x2, mcf x2, bzip2 x2, hmmer x2
MIX4: mcf x4, omnetpp x4
MIX5: SSCA2 x2, canneal x2, mcf x2, streamcluster x2



POWER EFFICIENCY



SYSTEM THROUGHPUT (NO ECC)



CONCLUSIONS

- Dynamic Granularity Memory System
  - HW-only variant of AGMS
  - Truly dynamic granularity adaptation
  - Higher performance [31% vs. CG]
  - Lower DRAM power [13% vs. CG]
- More in the paper
  - Reg/demux and address/command bus bandwidth
  - LPC&GPC details
  - DGMS with chipkill-correct support

THE DYNAMIC GRANULARITY MEMORY SYSTEM

Doe Hyun Yoon

IIL, HP Labs

Michael Sullivan
Min Kyu Jeong
Mattan Erez

ECE, UT Austin