

Spatial Memory Streaming

(with rotated patterns)

Michael Ferdman,

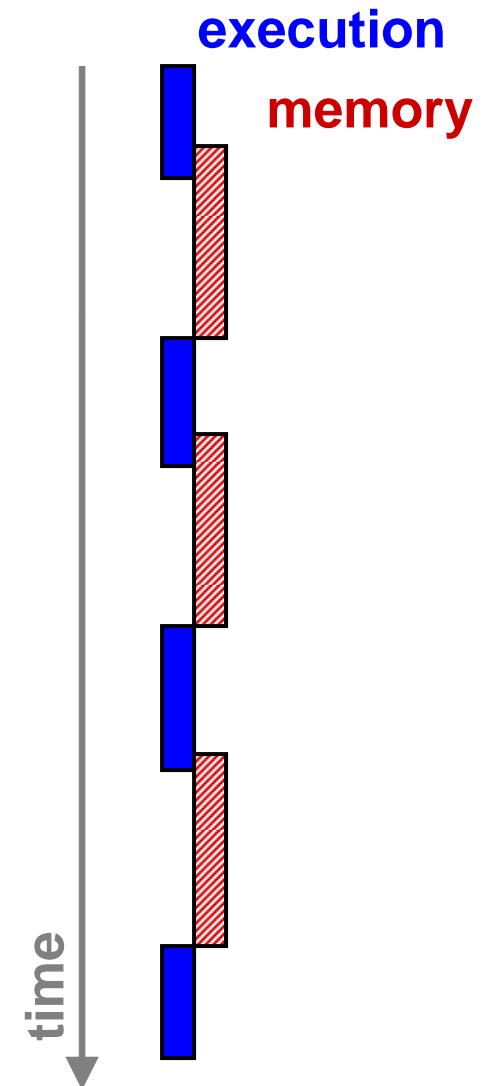
Stephen Somogyi, and Babak Falsafi

Computer Architecture Lab at
Carnegie Mellon

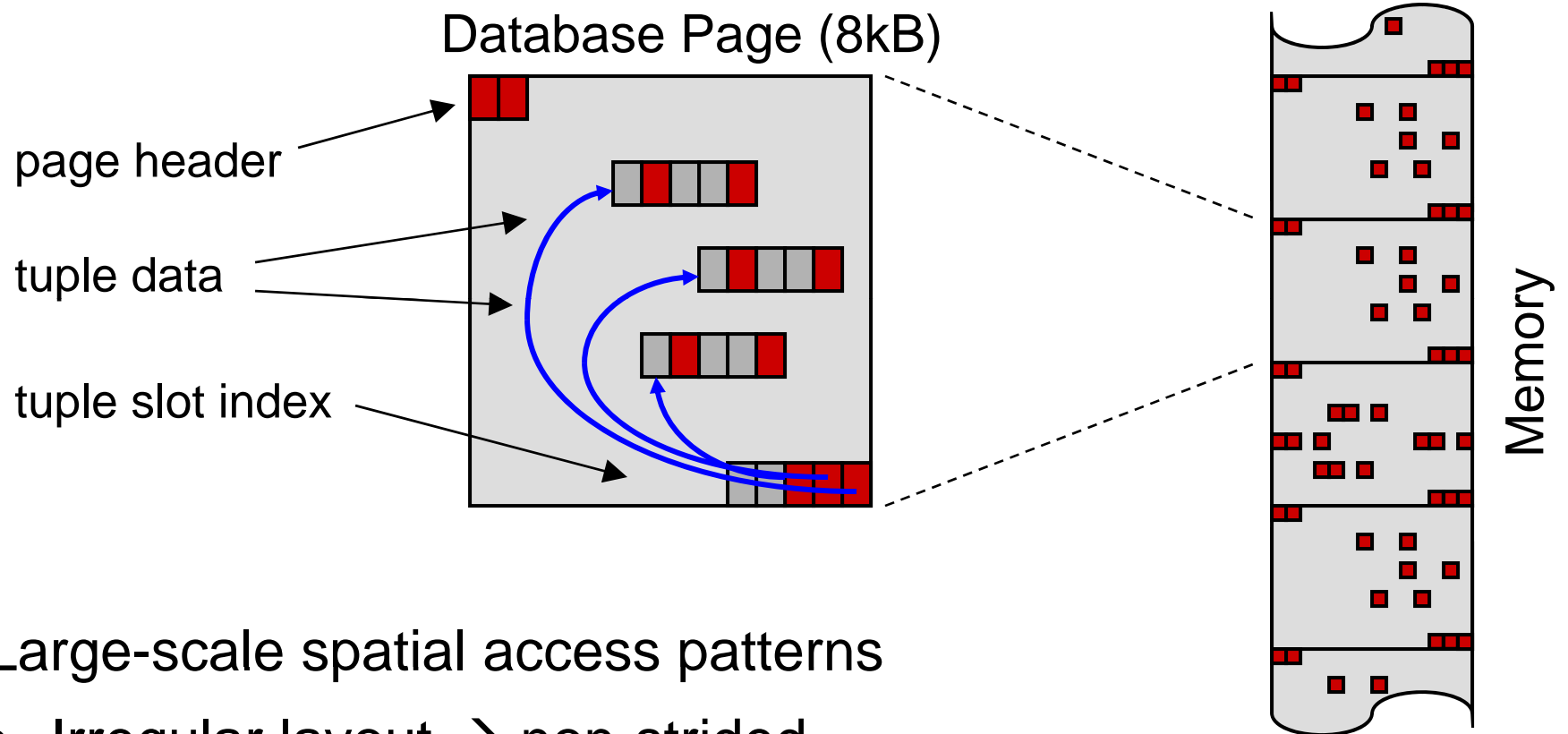


The Memory Wall

- Memory latency
 - 100's clock cycles; improving slowly
- Reduce time stalled on memory
 - Raise memory-level parallelism
- Capture all access patterns
 - Strides
 - Pointers (linked lists, trees)
 - Complex layouts (sparse structs)



Our Observation: Spatial Correlation



Large-scale spatial access patterns

- Irregular layout → non-strided
- Sparse → can't capture with cache blocks
- But, repetitive → **predict to improve MLP**

DPC Submission

- Code-correlated spatial patterns
 - Pattern storage independent of dataset size
 - Compulsory misses predictable
- Spatial Memory Streaming
 - Observes and records spatial patterns
 - Upon first access, stream remaining blocks
 - Fetch in parallel → increase MLP
 - Sparse patterns → fetch directly into L1

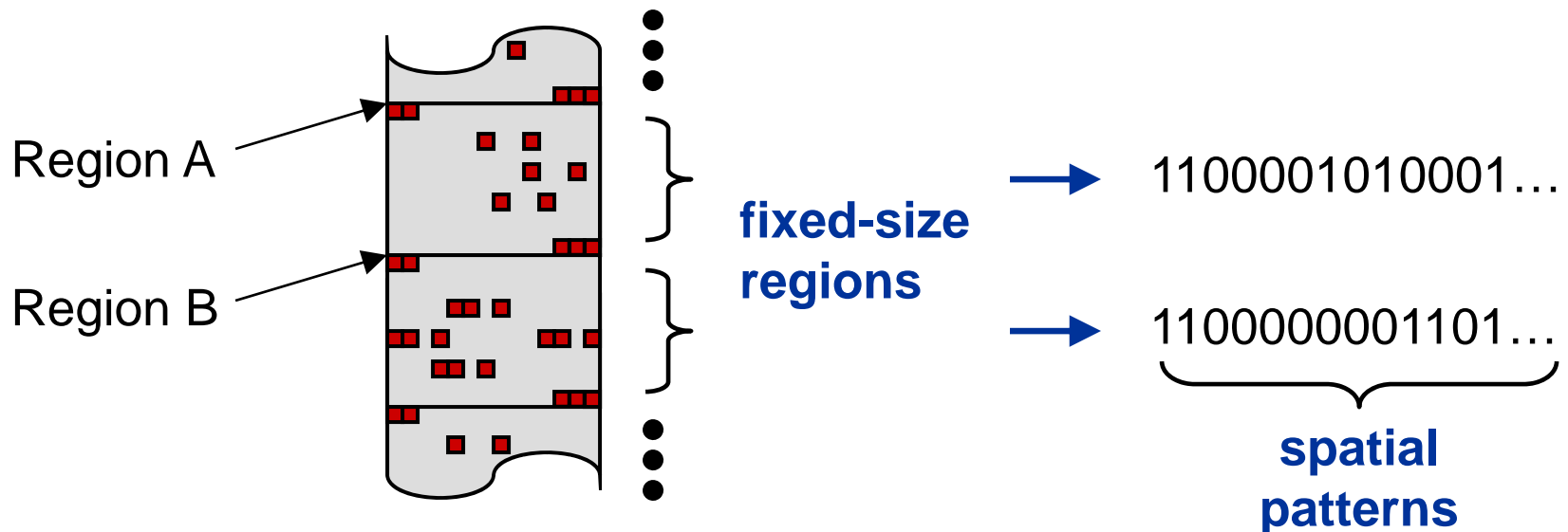
Outline

- Introduction
- Spatial Correlation
- Spatial Memory Streaming
- Pattern Rotation

Spatial Regions

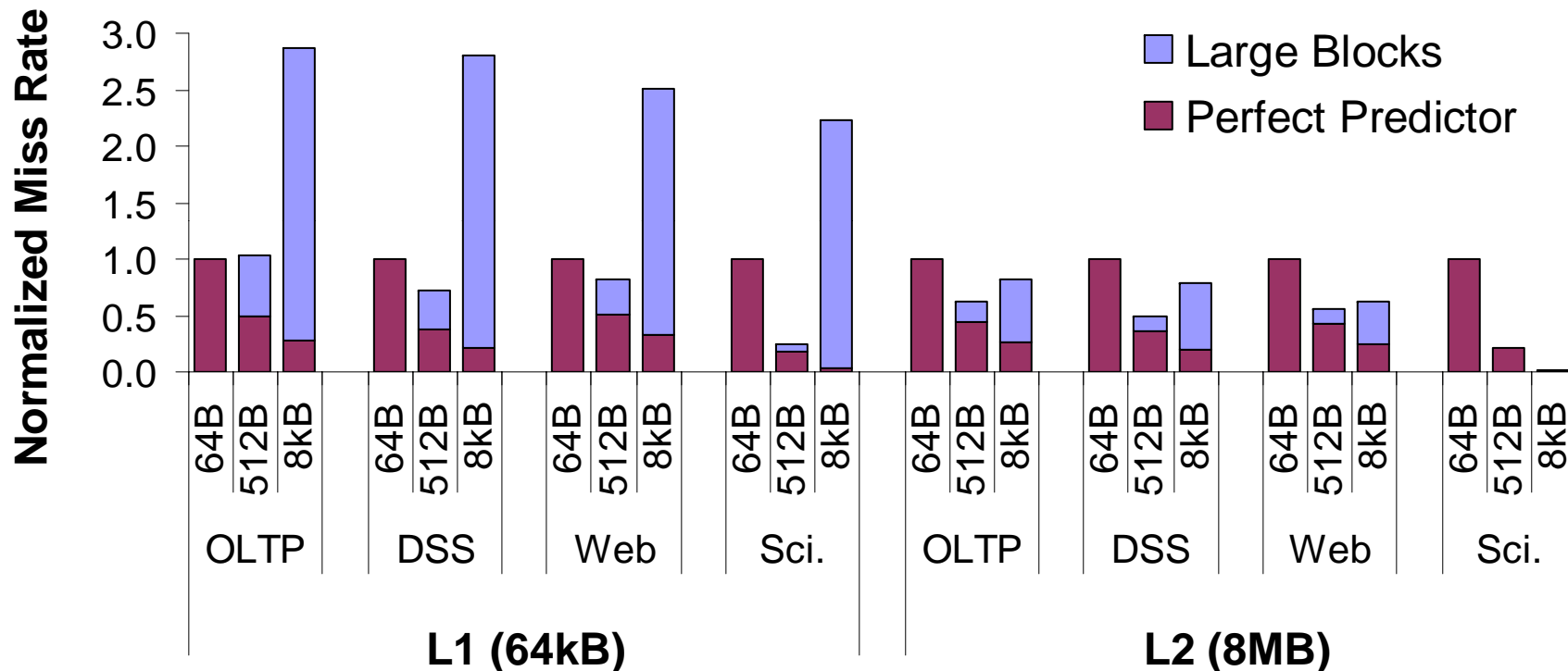
Logically divide memory into regions

- Identify region by base address
- Fixed-size
 - Simplifies hardware
 - Can represent spatial patterns as bit vectors



Why Exploit Spatial Correlation?

Perfect predictor = one miss per spatial pattern



- Large blocks → prohibitive miss rate at L1
→ bandwidth inefficient
- Spatial correlation → opportunity to eliminate misses

How to Exploit Spatial Correlation?

- Patterns are code-correlated
- Use PC to predict patterns
 - Storage independent of dataset size
 - Can predict compulsory misses

But, data layout may not be aligned to region

- PC is not enough [Kumar 98] [Chen 04]
- Offset within region identifies alignment

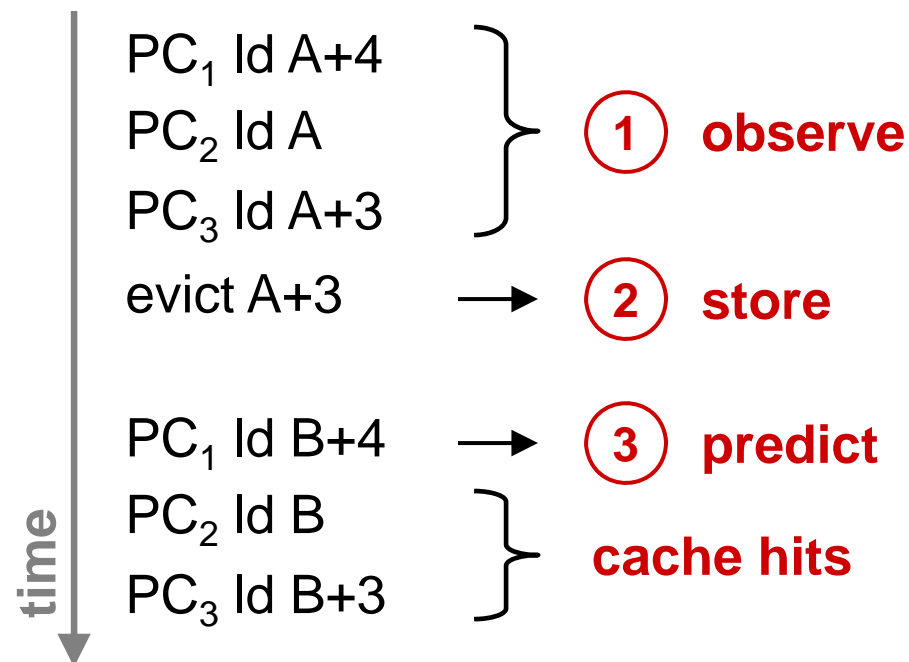
Practical hardware can predict spatial correlation

Outline

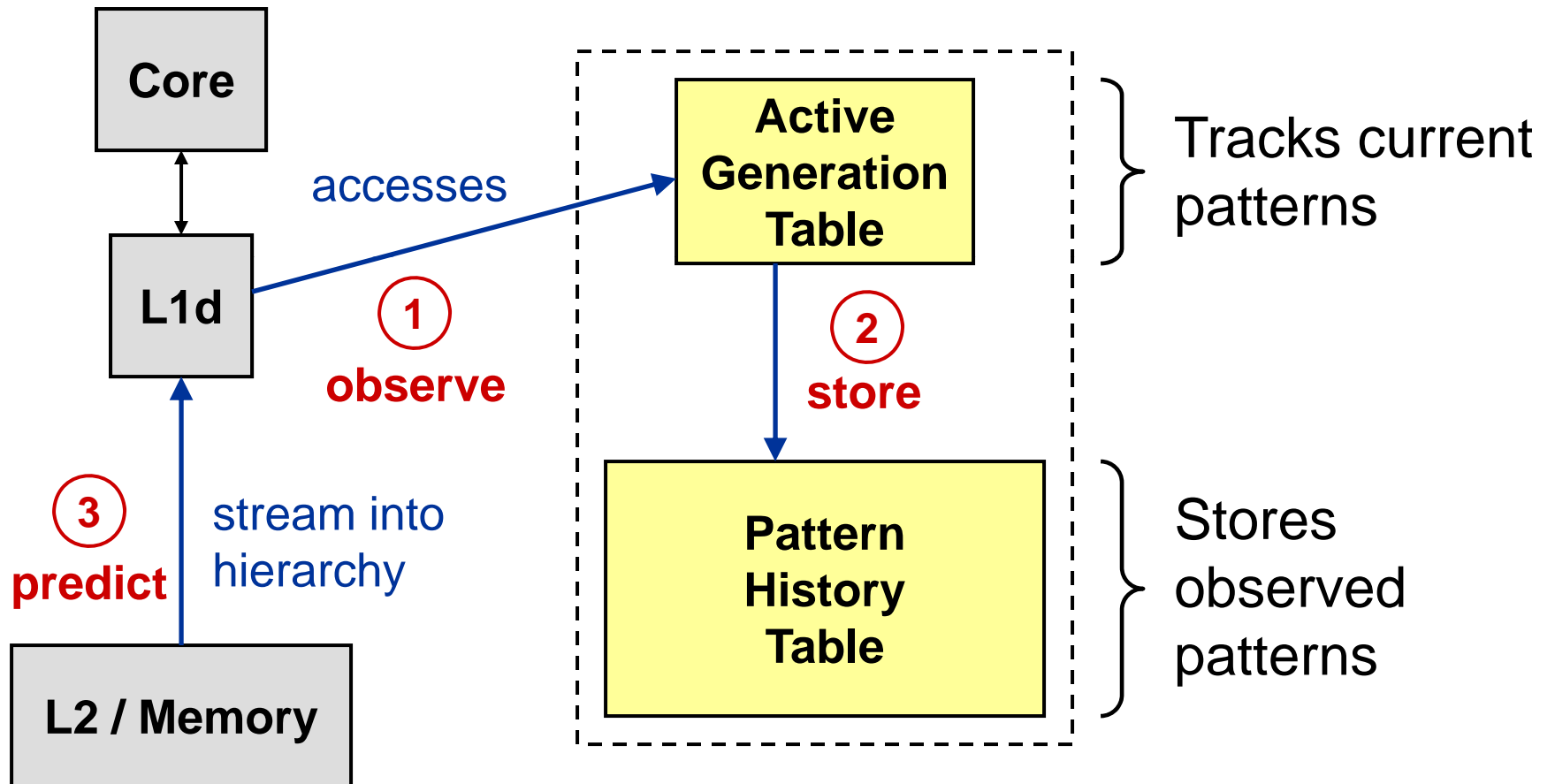
- Introduction
- Spatial Correlation
- Spatial Memory Streaming
- Rotated Patterns

Spatial Memory Streaming (SMS)

1. **Observe** pattern during generation
2. **Store** pattern at end of generation
3. **Predict** pattern at subsequent generation



SMS Hardware Overview



Learning Patterns

PC₁ Id A+4

PC₂ Id A

PC₃ Id A+3

evict A+3

PC₁ Id B+4

PC₂ Id B

PC₃ Id B+3

Active Generation Table

Region	PC / off	Pattern

- Active Generation Table
 - Accumulates patterns
 - 32 ~ 64 entries sufficient

Learning Patterns

PC₁ Id A+4

PC₂ Id A

PC₃ Id A+3

evict A+3

PC₁ Id B+4

PC₂ Id B

PC₃ Id B+3

Active Generation Table

Region	PC / off	Pattern
A	PC₁ / 4	00001000

- First access creates new entry

Learning Patterns

PC₁ ld A+4

PC₂ ld A

PC₃ ld A+3

evict A+3

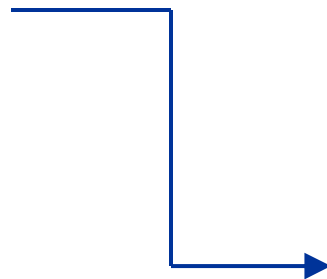
PC₁ ld B+4

PC₂ ld B

PC₃ ld B+3

Active Generation Table

Region	PC / off	Pattern
A	PC₁ / 4	10001000



- Further accesses accumulate bits in pattern

Learning Patterns

PC₁ ld A+4

PC₂ ld A

PC₃ ld A+3

evict A+3

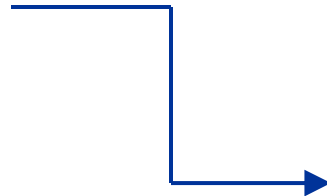
PC₁ ld B+4

PC₂ ld B

PC₃ ld B+3

Active Generation Table

Region	PC / off	Pattern
A	PC₁ / 4	10011000



- Further accesses accumulate bits in pattern

Learning Patterns

PC₁ Id A+4

PC₂ Id A

PC₃ Id A+3

evict A+3

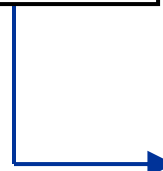
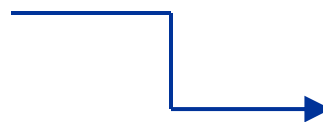
PC₁ Id B+4

PC₂ Id B

PC₃ Id B+3

Active Generation Table

Region	PC / off	Pattern



to Pattern History Table

- Eviction ends pattern

10011000
@ PC₁/4

Learning Patterns

PC₁ ld A+4

PC₂ ld A

PC₃ ld A+3

evict A+3



Pattern History Table

PC / off	Pattern
PC₁ / 4	10011000

PC₁ ld B+4

PC₂ ld B

PC₃ ld B+3

- Pattern History Table
 - Stores previously-observed patterns
 - Set-associative: 8-way 2k-entries

Predicting Patterns

PC₁ ld A+4
 PC₂ ld A
 PC₃ ld A+3
 evict A+3

PC₁ ld B+4
 PC₂ ld B
 PC₃ ld B+3

Pattern History Table

PC / off	Pattern
PC₁ / 4	10011000

10011000 → **stream B, B+3 into cache**

- First access looks in Pattern History Table
- Stream predicted blocks into L1 cache

Predicting Patterns

PC₁ ld A+4

PC₂ ld A

PC₃ ld A+3

evict A+3

PC₁ ld B+4

PC₂ ld B **cache hit**

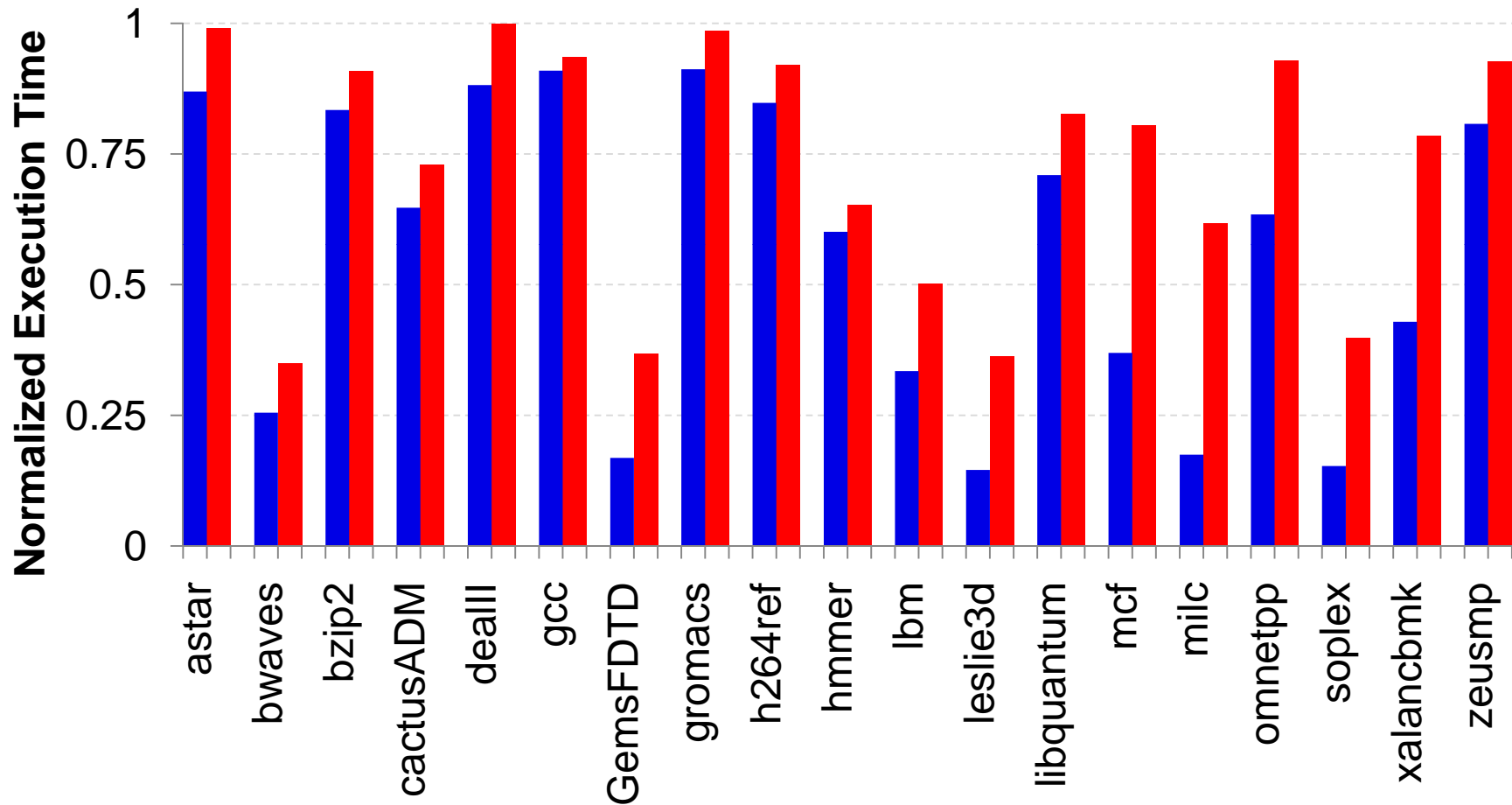
PC₃ ld B+3 **cache hit**

Pattern History Table

PC / off	Pattern
PC₁ / 4	10011000

- Subsequent accesses hit in L1 cache

SMS Results (SPEC CPU 2006)



Outline

- Introduction
- Spatial Correlation
- Spatial Memory Streaming
- Rotated Patterns

Our Observation: Rotated Patterns

- PC is insufficient to predict pattern
 - Offset of first access highly variable
 - *But:* Access pattern almost always the same
- Can store “rotated” patterns in PHT
 - Rotate as needed before prediction

Learning Patterns

PC₁ ld A+4

PC₂ ld A+8

PC₃ ld A+7

evict A+7

PC₁ ld B+2

PC₂ ld B+6

PC₃ ld B+5

Active Generation Table

Region	PC / off	Pattern

- Active Generation Table
 - Accumulates patterns
 - 32 ~ 64 entries sufficient

Learning Patterns

PC₁ Id A+4

PC₂ Id A+8

PC₃ Id A+7

evict A+7

PC₁ Id B+2

PC₂ Id B+6

PC₃ Id B+5

Active Generation Table

Region	PC / off	Pattern
A	PC₁ / 4	100000000

- First access creates new entry
- Bits are recorded *rotated left* by initial offset

Learning Patterns

PC₁ ld A+4

PC₂ ld A+8

PC₃ ld A+7

evict A+7

PC₁ ld B+2

PC₂ ld B+6

PC₃ ld B+5

Active Generation Table

Region	PC / off	Pattern
A	PC₁ / 4	100010000

- Further accesses accumulate bits in pattern
- Bits are recorded *rotated left* by initial offset

Learning Patterns

PC₁ ld A+4

PC₂ ld A+8

PC₃ ld A+7

evict A+7

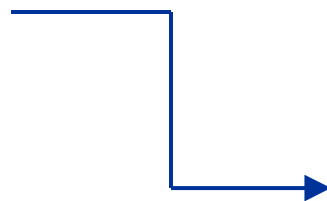
PC₁ ld B+2

PC₂ ld B+6

PC₃ ld B+5

Active Generation Table

Region	PC / off	Pattern
A	PC₁ / 4	100110000



- Further accesses accumulate bits in pattern
- Bits are recorded *rotated left* by initial offset

Learning Patterns

PC₁ ld A+4

PC₂ ld A+8

PC₃ ld A+7

evict A+7

PC₁ ld B+2

PC₂ ld B+6

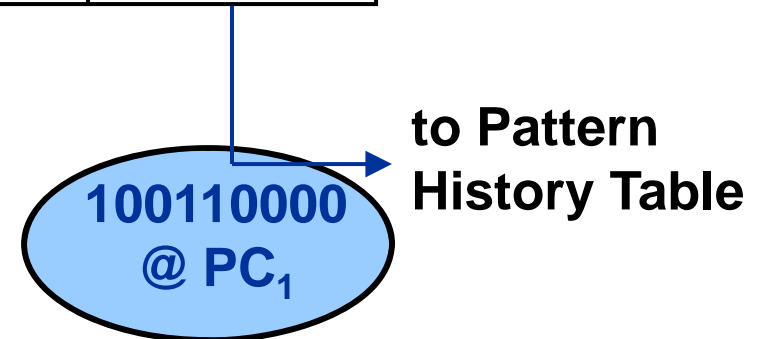
PC₃ ld B+5

Active Generation Table

Region	PC / off	Pattern



- Eviction ends pattern



PC only – no offset

Learning Patterns

PC₁ ld A+4

PC₂ ld A+8

PC₃ ld A+7

evict A+7

PC₁ ld B+2

PC₂ ld B+6

PC₃ ld B+5

PC only – no offset

Pattern History Table

PC	Pattern
PC ₁	100110000



- Pattern History Table
 - Stores previously-observed patterns
 - Set-associative: 8-way 2k-entries

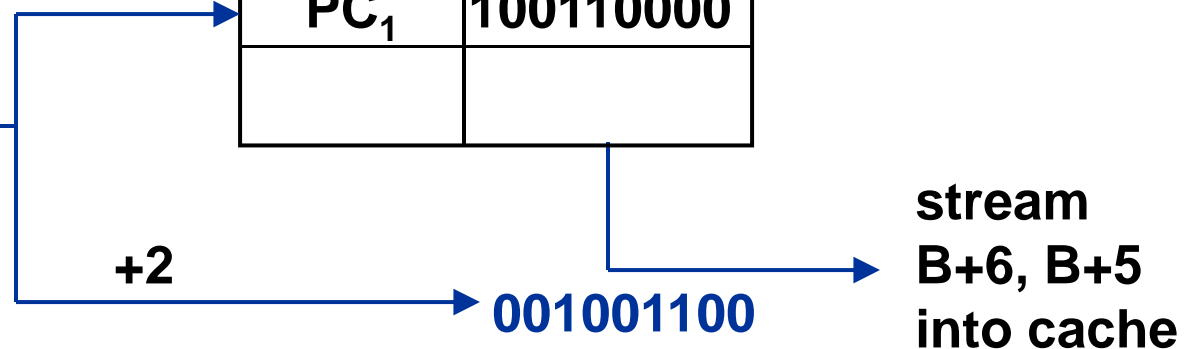
Predicting Patterns

PC₁ ld A+4
 PC₂ ld A+8
 PC₃ ld A+7
 evict A+7

PC₁ ld B+2
 PC₂ ld B+6
 PC₃ ld B+5

Pattern History Table

PC	Pattern
PC₁	100110000



- First access looks in Pattern History Table
- Stream predicted *rotated* blocks into L1 cache

Predicting Patterns

PC₁ ld A+4

PC₂ ld A+8

PC₃ ld A+7

evict A+7

PC₁ ld B+2

PC₂ ld B+6 **cache hit**

PC₃ ld B+5 **cache hit**

Pattern History Table

PC	Pattern
PC₁	100110000

- Subsequent accesses hit in L1 cache

Rotation: Theoretical Benefit

Before

Pattern History Table

PC / off	Pattern
PC ₁ / 4	000010011
PC ₁ / 2	001001100
PC ₁ / 5	100001001
PC ₁ / 1	100110000

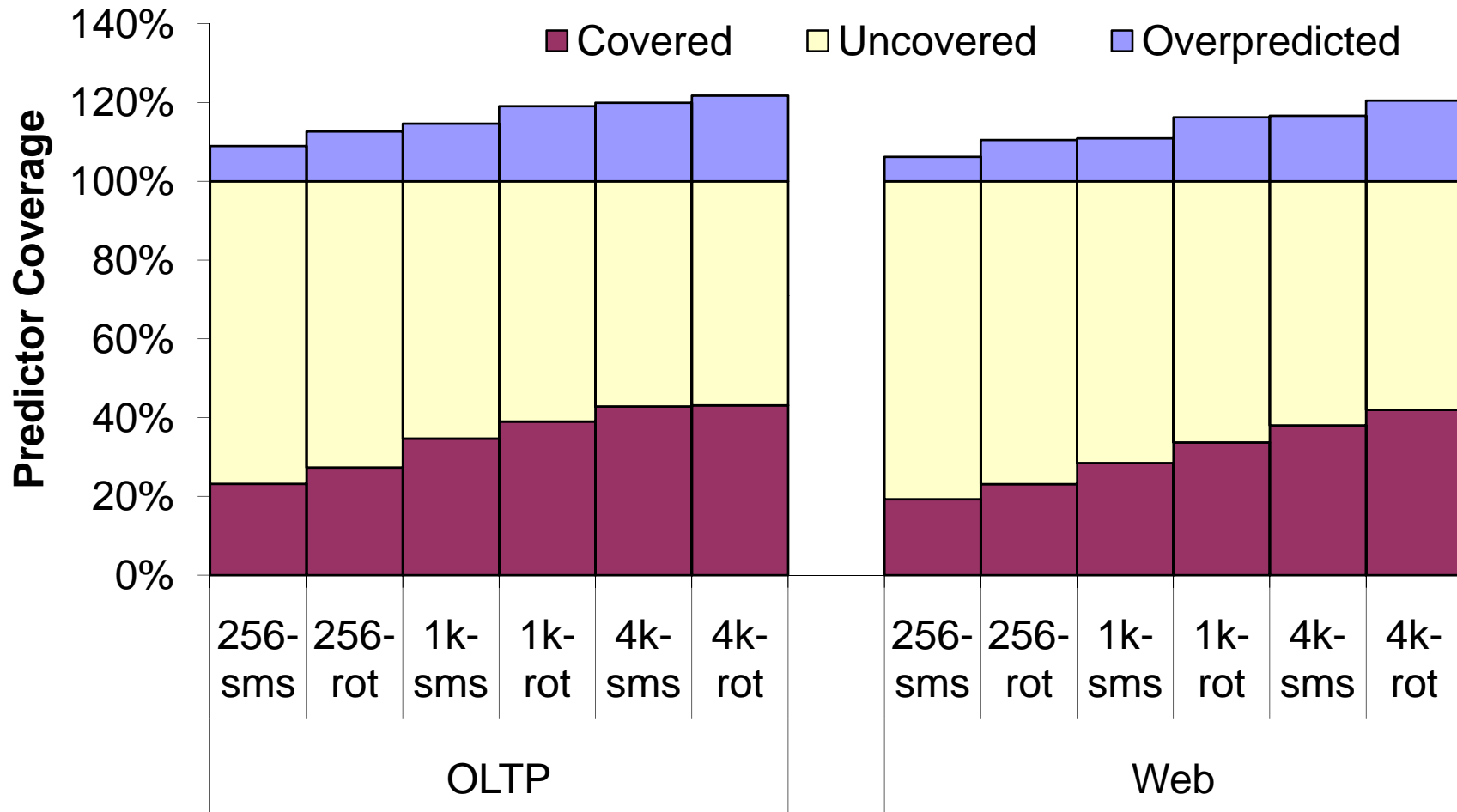
After

Pattern History Table

PC	Pattern
PC ₁	100110000

***Rotated patterns* ⇒ saves PHT storage**

Rotation: Practical Benefit



***Rotated patterns* ⇒ saves 2x PHT storage**

Rotation: Applicability

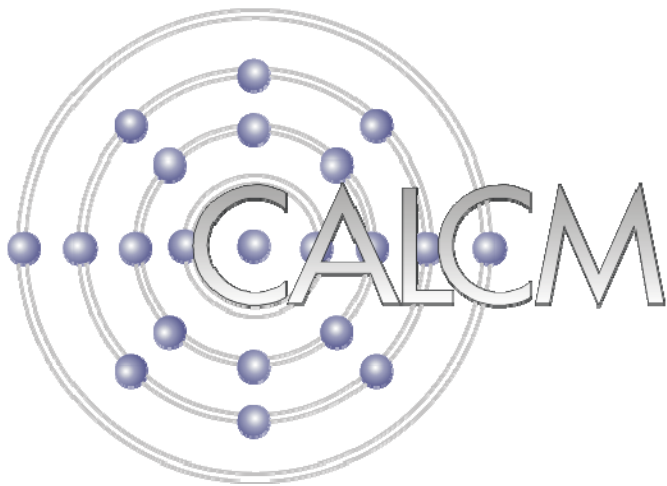
- Commercial workloads (e.g., OLTP, web, DSS)
 - Large instruction footprints (>1MB [cidr 07])
 - Benefits from rotation
- Desktop/engineering (e.g., SPEC CPU 2000)
 - Small instruction footprints (fit in L1-I)
 - Unlikely to benefit from rotation [hpca 04]
 - SPEC CPU 2006 very similar to CPU 2000

Need broad range of workloads to observe benefit of rotated patterns

Conclusion

- **Spatial Memory Streaming**
 - Learns large-scale spatial access patterns
 - Streams remaining blocks upon first access in pattern
 - Accurate predictor with small hardware cost
- **Rotated Patterns**
 - Stores one rotated version of spatial pattern per PC
 - Significant reduction in number of patterns
 - Needed in PHT-capacity constrained environment

Questions ?



STeMS Project

Spatio-Temporal Memory Streaming

www.ece.cmu.edu/~stems

Computer Architecture Laboratory

Carnegie Mellon University

www.ece.cmu.edu/~calcm