Prefetching

Prof. Mikko H. Lipasti
University of Wisconsin-Madison

Lecture notes based on notes by John P. Shen and Mark Hill
Updated by Mikko Lipasti

Prefetching

- Even “demand fetching” prefetches other words in block
  - Spatial locality
- Prefetching is useless
  - Unless a prefetch costs less than demand miss
- Ideally, prefetches should
  - Always get data before it is referenced
  - Never get data not used
  - Never prematurely replace data
  - Never interfere with other cache activity

Software Prefetching

- Use compiler to try to
  - Prefetch early
  - Prefetch accurately
- Prefetch into
  - Register (binding)
    - Use normal loads? Stall-on-use (Alpha 21164)
    - What about page faults? Exceptions?
  - Caches (non-binding) – preferred
    - Needs ISA support

Software Prefetching

- For example:

```
do j= 1, cols
  do ii = 1 to rows by BLOCK
    prefetch (&(x[i,j])+BLOCK)    # prefetch one block ahead
    do i = ii to ii + BLOCK-1
      sum = sum + x[i,j]
```

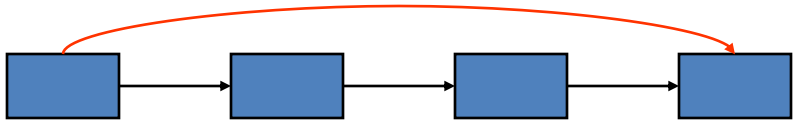
- How many blocks ahead should we prefetch?
  - Affects timeliness of prefetches
  - Must be scaled based on miss latency

Hardware Prefetching

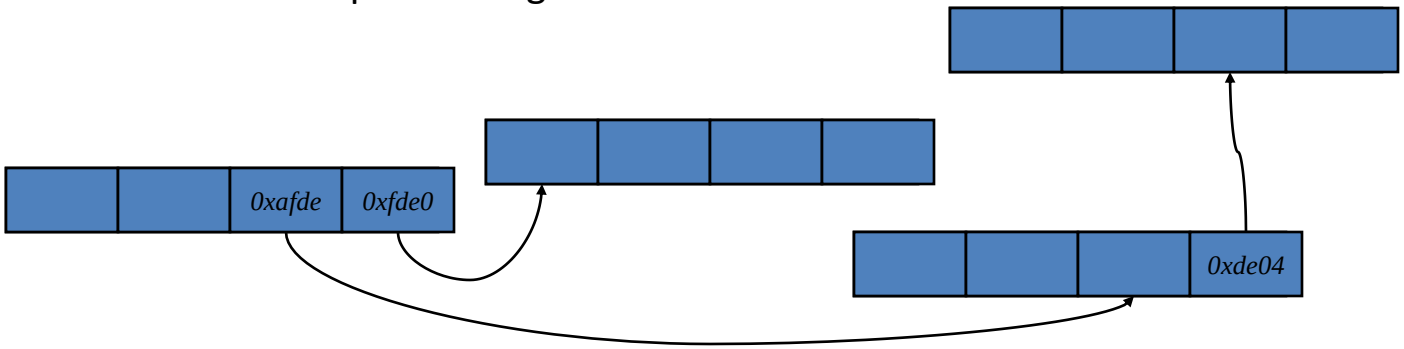
- What to prefetch
  - One block spatially ahead
  - N blocks spatially ahead
  - Based on observed stride, track/prefetch multiple strides
- Training hardware prefetcher
  - On every reference (expensive)
  - On every miss (information loss)
  - Misses at what level of cache?
  - Prefetchers at every level of cache?
- Pressure for nonblocking miss support (MSHRs)

Prefetching for Pointer-based Data Structures

- What to prefetch
  - Next level of tree: n+1, n+2, n+?
    - Entire tree? Or just one path
  - Next node in linked list: n+1, n+2, n+?
  - Jump-pointer prefetching



- How to prefetch
  - Software places jump pointers in data structure
  - Hardware scans blocks for pointers
    - Content-driven data prefetching



Stream or Prefetch Buffers

- Prefetching causes capacity and conflict misses (pollution)
  - Can displace useful blocks
- Aimed at compulsory and capacity misses
- Prefetch into buffers, NOT into cache
  - On miss start filling stream buffer with successive lines
  - Check both cache and stream buffer
    - Hit in stream buffer => move line into cache (promote)
    - Miss in both => clear and refill stream buffer
- Performance
  - Very effective for I-caches, less for D-caches
  - Multiple buffers to capture multiple streams (better for D-caches)
- Can use with any prefetching scheme to avoid pollution

Example: Global History Buffer

- K. Nesbit, J. Smith, “Prefetching using a global history buffer”, HPCA 2004.
  - [slides from conference talk follow]
- Hardware prefetching scheme
- Monitors miss stream
- Learns correlations
- Issues prefetches for likely next address

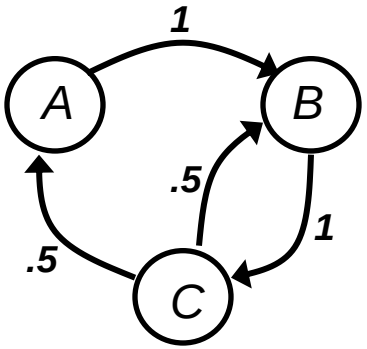
Markov Prefetching

- Markov prefetching forms address correlations
  - Joseph and Grunwald (ISCA '97)
- Uses global memory addresses as states in the Markov graph
- Correlation Table *approximates* Markov graph

Miss Address Stream

A B C A B C B C . . .

Markov Graph



Correlation Table

miss address		***1st predict.***	***2nd predict.***
→	***A***	***B***	
	B	***C***	
	C	***B***	***A***

Correlation Prefetching

- Distance Prefetching forms *delta* correlations
  - Kandiraju and Sivasubramaniam (ISCA '02)
- Delta-based prefetching leads to much smaller table than "classical" Markov Prefetching
- Delta-based prefetching can remove compulsory misses

Markov Prefetching

Miss Address Stream

27 28 29 27 28 29 28 29

Distance Prefetching

Global Delta Stream

1 1 -2 1 1 -1 1

miss address

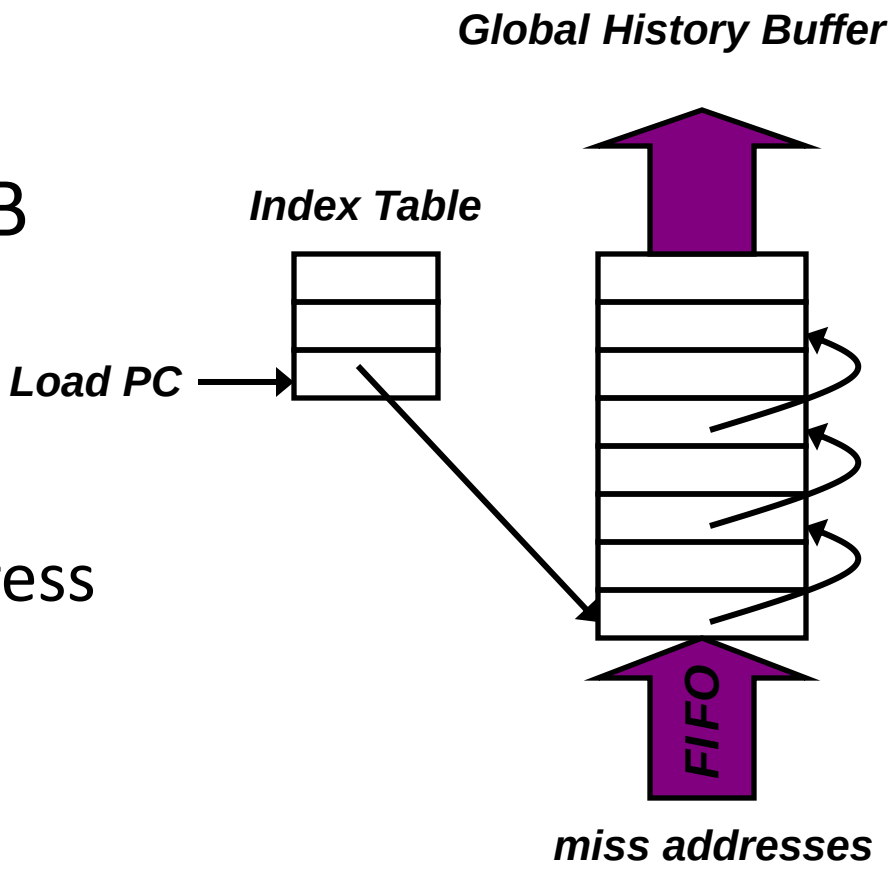
	1st predict.	***2nd predict.***
27	***28***	
28	***29***	
29	***28***	***29***

global delta

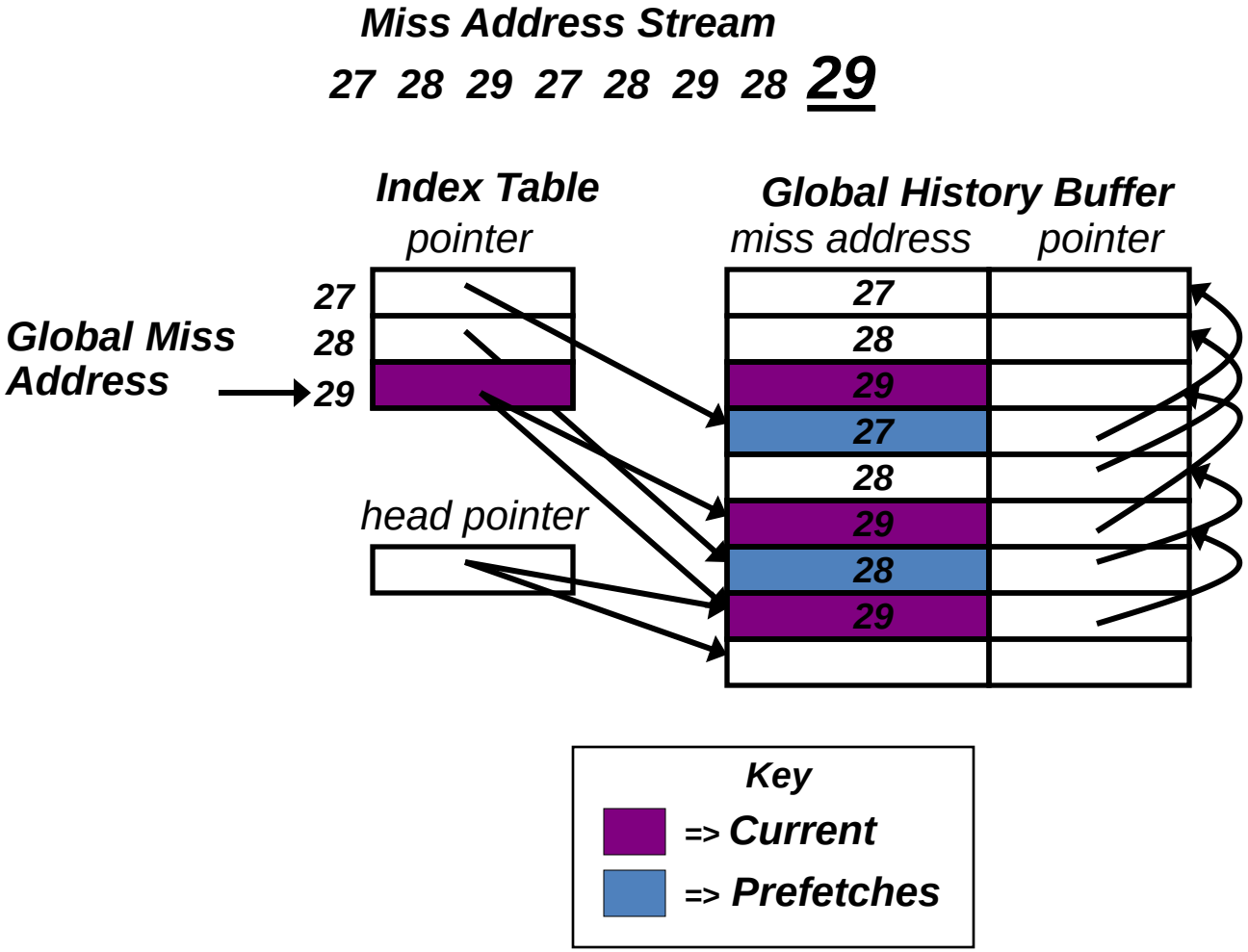
	1st predict.	***2nd predict.***
-2	***1***	
-1	***1***	
1	***-1***	***-2***

Global History Buffer (GHB)

- Holds miss address history in FIFO order
- Linked lists within GHB connect related addresses
  - Same static load
  - Same global miss address
  - Same global delta
- *Linked list walk is short compared with L2 miss latency*



GHB - Example


Miss Address Stream
27 28 29 27 28 29 28 29
Index Table
pointer
Global History Buffer
miss address
pointer
Global Miss Address
27
28
29
head pointer
27
28
29
27
28
29
28
29
Key
=> Current
=> Prefetches

GHB – Deltas

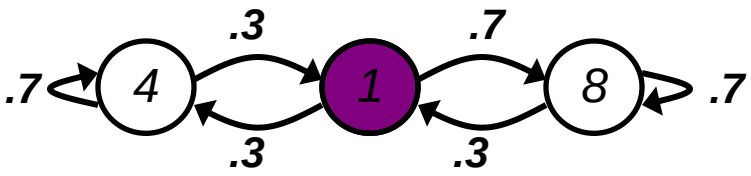
Miss Address Stream

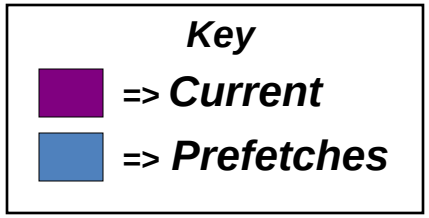
27 28 36 44 45 49 53 54 62 70 71

Global Delta Stream

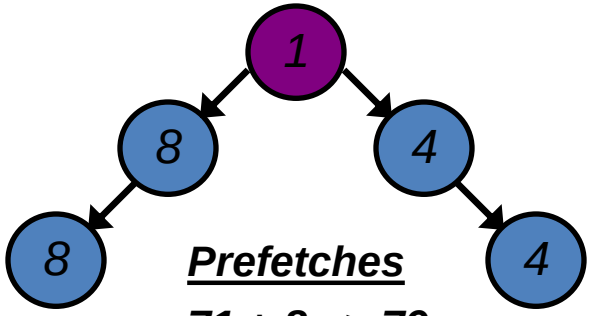
1 8 8 1 4 4 1 8 8 1

Markov Graph





Hybrid



Prefetches

71 + 8 => 79

71 + 4 => 75

Prefetches

71 + 8 => 79

79 + 8 => 87

Prefetches

71 + 4 => 75

79 + 4 => 79

GHB – Hybrid Delta

- Width prefetching suffers from poor accuracy and short look-ahead
- Depth prefetching has good look-ahead, but may miss prefetch opportunities when a number of “next” addresses have similar probability
- The hybrid method combines depth and width

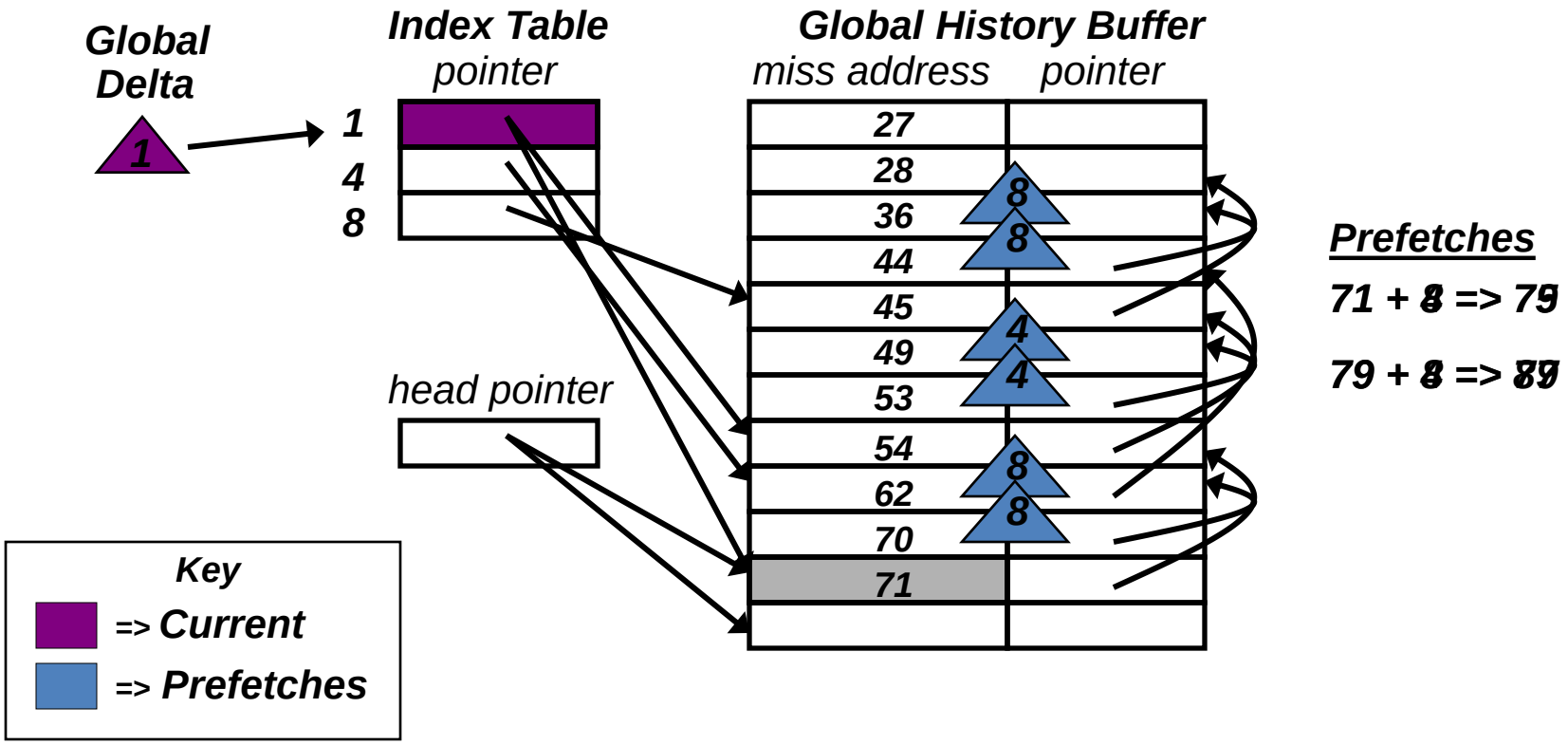
GHB - Hybrid Example

Miss Address Stream

27 28 36 44 45 49 53 54 62 70 71

Global Delta Stream

1 8 8 1 4 4 1 8 8 1



Summary

- Prefetching anticipates future memory references
  - Software prefetching
  - Next-block, stride prefetching
  - Global history buffer prefetching
- Issues/challenges
  - Accuracy
  - Timeliness
  - Overhead (bandwidth)
  - Conflicts (displace useful data)