# Stream Chaining: Exploiting Multiple Levels of Correlation in Data Prefetching

Pedro Díaz and Marcelo Cintra

University of Edinburgh

http://www.homepages.inf.ed.ac.uk/mc/Projects/CELLULAR

# Outline

- **Motivation**
- Correlation and Localization
- Stream Chaining and Miss Graph Prefetching
- Experimental Setup and Results
- Related Work
- Conclusions

# The "Memory Wall" and Prefetching

- The Memory Wall is still a problem
  - After decades of logic and DRAM technology disparity, memory access costs hundreds of processor cycles
  - On-chip cache quotas per processor unlikely to increase
  - Off-chip memory bandwidth quota per processor likely to decrease (unless some fancy memory technology succeeds)
- (Hardware) Prefetching is a viable solution
  - Time-tested approach used in most commercial processors
  - Trades-off memory bandwidth for latency (especially good if some fancy memory technology succeeds)

# Prefetching

- Prefetchers work by uncovering patterns in the miss address stream: **correlation** (e.g., address deltas)
- Prefetchers often separate misses into multiple streams: **localization** (e.g., by instruction)
- To eliminate more misses and hide longer latencies prefetchers often use prefetch degree greater than one
- Prefetchers often measured against three metrics:
  - Accuracy: ratio of used prefetches over all prefetches
  - Coverage: ratio of used prefetches over original misses
  - Timeliness: data arrives too early, too late, or just in time

# The Problem with Prefetching

- Correlation on global miss stream often suffers from poor accuracy
- Prefetching along localized streams often suffers from poor coverage and timeliness
  - Streams lose time ordering information of misses
  - "Cold" misses across stream boundaries
- Deep prefetching suffers from diminishing accuracy
- Applications access patterns exhibit different correlation patterns

**Ideally what we want is to combine multiple localized streams to improve coverage and timeliness while keeping accuracy high**

# Outline

- Motivation
- **Correlation and Localization**
- Stream Chaining and Miss Graph Prefetching
- Experimental Setup and Results
- Related Work
- Conclusions

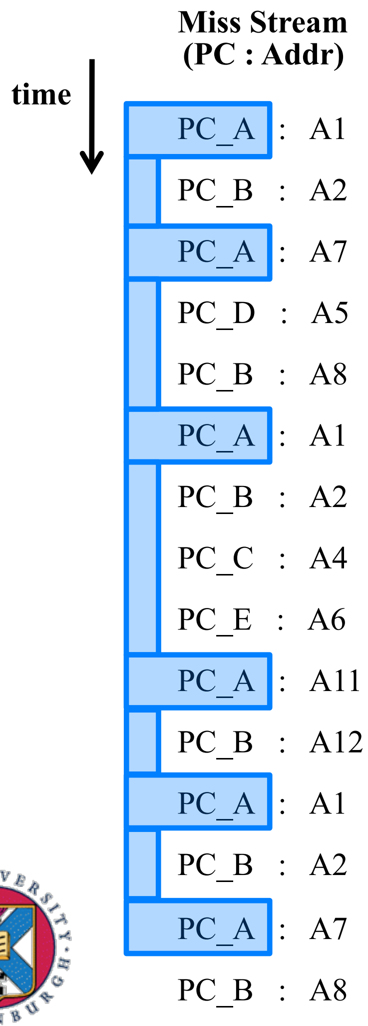
# Correlation

- Establishing "relationship" among addresses of misses. For instance:
  - Sequential: miss to line L is followed by miss to line L+1
  - Time : miss to address A is followed by miss to address B
  - Delta: miss to address A is followed by miss to address A +d
  - Markov: e.g., miss to address A is followed by miss to address B with probability p and miss to address C with probability (1-p)
- Correlations are found by inspecting miss history and are used to predict next miss

# Localization

- Complete global history is undesirable in most cases
  - Misses from unrelated sources (e.g., from pointer chasing followed by data object manipulation)
  - "Wild" interleaving of misses (e.g., OOO execution, infrequent control flow)
  - Correlations over long traces
- Localization: group misses according to some common property. For instance:
  - PC: misses from same static instruction
  - Temporal: misses that occur at about the same time
  - Spatial: misses to similar regions in memory address space
- Attempts to exploit some high-level behaviour

# Localization

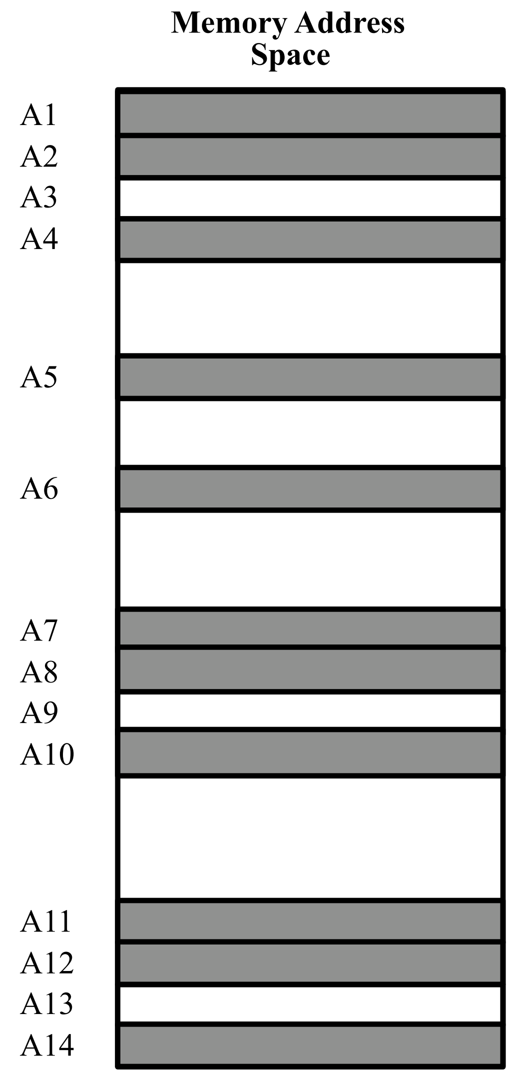



**PC Correlation**

**PC Localized Streams:**

**A1 → A7 → A1 → A11 → A1 → A7**

**A2 → A8 → A2 → A12 → A2 → A8**



# Localization

**Miss Stream (PC : Addr)**

time

| | |
|---|---|
| PC_A | A1 |
| PC_B | A2 |
| PC_A | A7 |
| PC_D | A5 |
| PC_B | A8 |
| PC_A | A1 |
| PC_B | A2 |
| PC_C | A4 |
| PC_E | A6 |
| PC_A | A11 |
| PC_B | A12 |
| PC_A | A1 |
| PC_B | A2 |
| PC_A | A7 |
| PC_B | A8 |

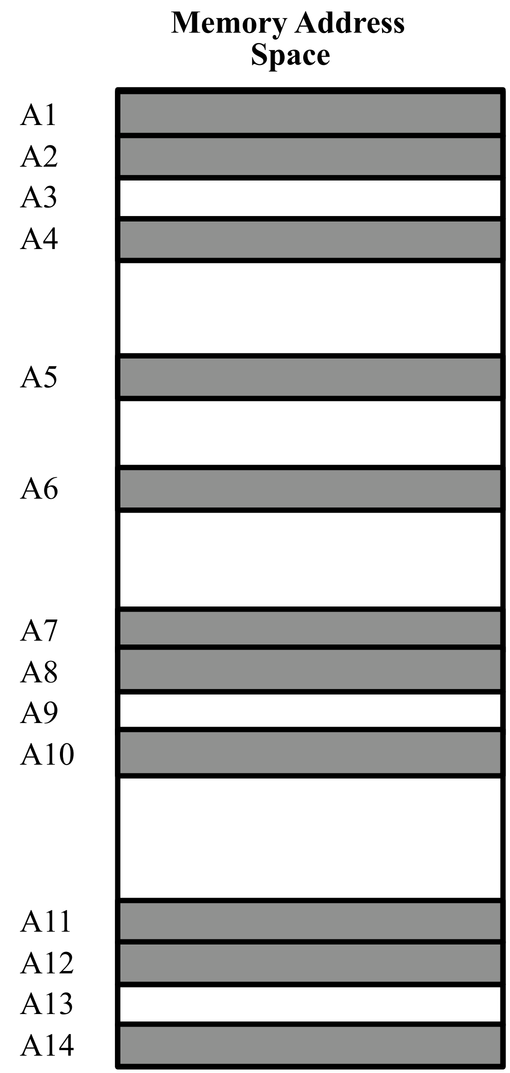
**Temporal Correlation**

**Time Localized Streams:**

**A1 → A2 → A7 → A5 → A8**

**A1 → A2 → A4 → A6 → A11 → A12**

**A1 → A2 → A7 → A8**



# Localization

time

**Miss Stream**
**(PC : Addr)**

PC_A : A1
PC_B : A2
PC_A : A7
PC_D : A5
PC_B : A8
PC_A : A1
PC_B : A2
PC_C : A4
PC_E : A6
PC_A : A11
PC_B : A12
PC_A : A1
PC_B : A2
PC_A : A7
PC_B : A8

**Spatial Correlation**

**Space Localized Streams:**

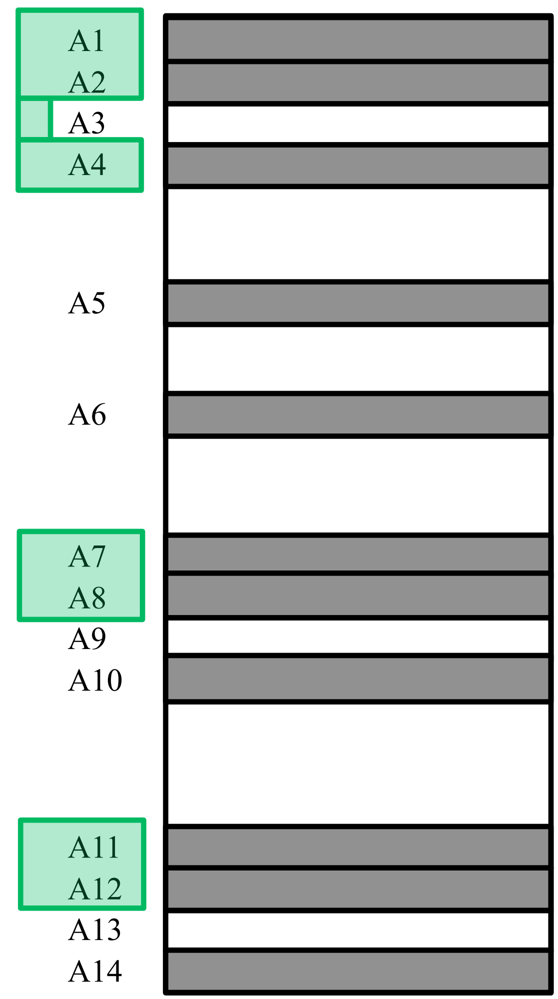
**A1 → A2**

**A1 → A2 → A4**

**A7 → A8**

**A11 → A12**

**Memory Address Space**



# Outline

- Motivation
- Correlation and Localization
- **Stream Chaining and Miss Graph Prefetching**
- Experimental Setup and Results
- Related Work
- Conclusions

# Stream Chaining: Idea and Operation

- Chain streams:
  - Start from global, ordered, miss stream
  - Perform localization and build localized streams
  - Order and link streams according to program execution to *partially* reconstruct order of misses
- Prefetch
  - On a miss to stream A follow chain and identify streams that *commonly* follow A
  - Perform correlation on each stream individually
  - Prefetch data for streams that follow A and, possibly, also for A itself

# Benefits and Limitations

+ Recover chronological information following program's *stable* memory access pattern
+ Still eliminate "spurious" misses
+ Still benefit from better predictability of localized streams
+ Prefetch across stream boundaries
+ Better use of large prefetch degrees
- Stream chain patterns must be stable
- Stream chains must be relatively small as to be manageable
- Longer run time of algorithm as must correlate on multiple streams

# Miss Graph Prefetcher

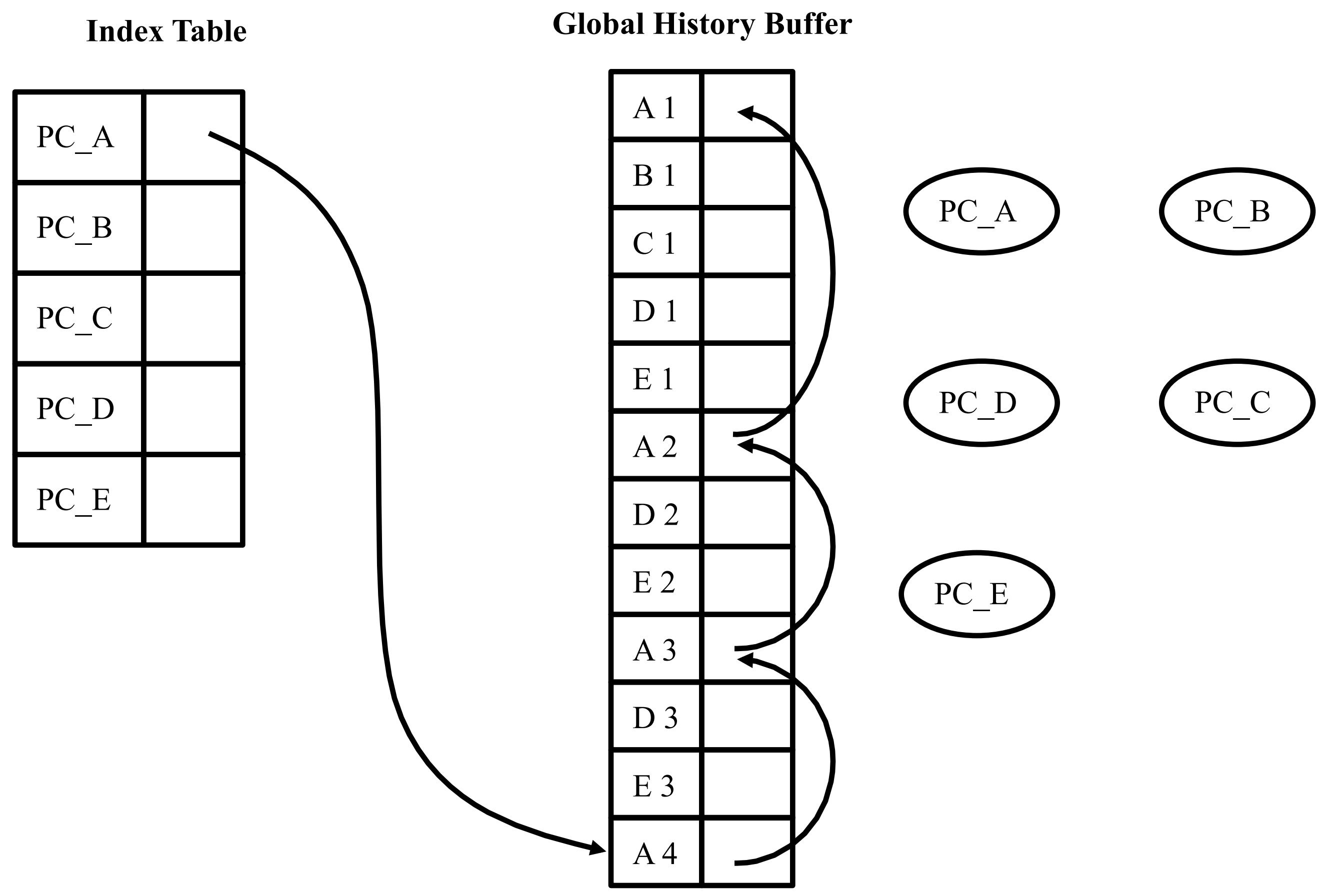
- Based on Nesbitt and Smith's GHB structure (HPCA'04)
- Uses PC localization with delta correlation (PC/DC)
- Represents stream chains as simple directed graphs
  - Nodes represent streams and edges represent time ordering (i.e., miss to stream A is followed by miss to stream B ➡ A→B)
  - Only 1 outgoing edge per node but multiple incoming edges possible
  - Edges only added to recurring sequences by using a threshold
  - Cycles allowed
- Named PC/DC/MG

# Miss Graph Prefetcher

**Miss Stream**
**(PC : Addr)**

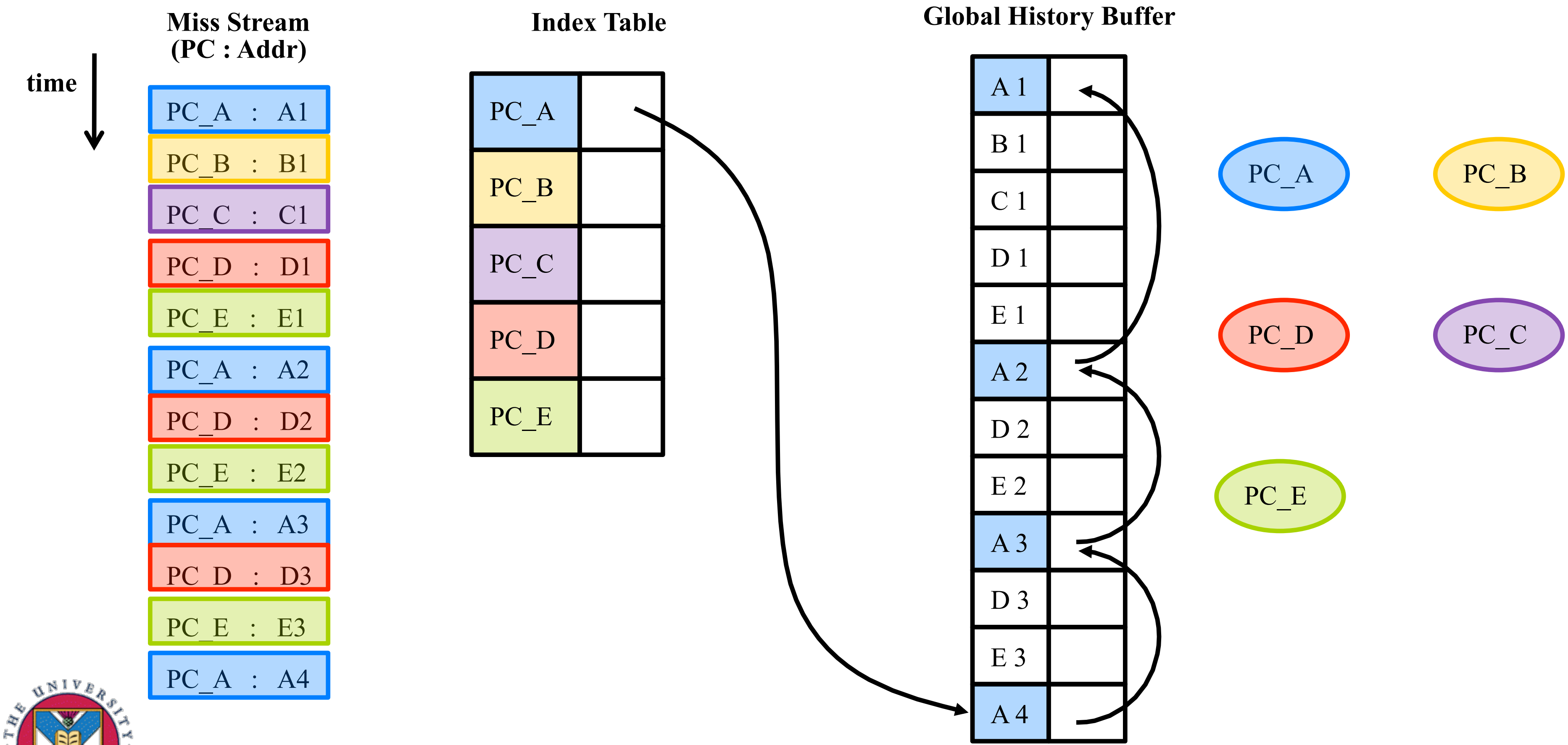
time ↓

PC_A : A1
PC_B : B1
PC_C : C1
PC_D : D1
PC_E : E1
PC_A : A2
PC_D : D2
PC_E : E2
PC_A : A3
PC_D : D3
PC_E : E3
PC_A : A4



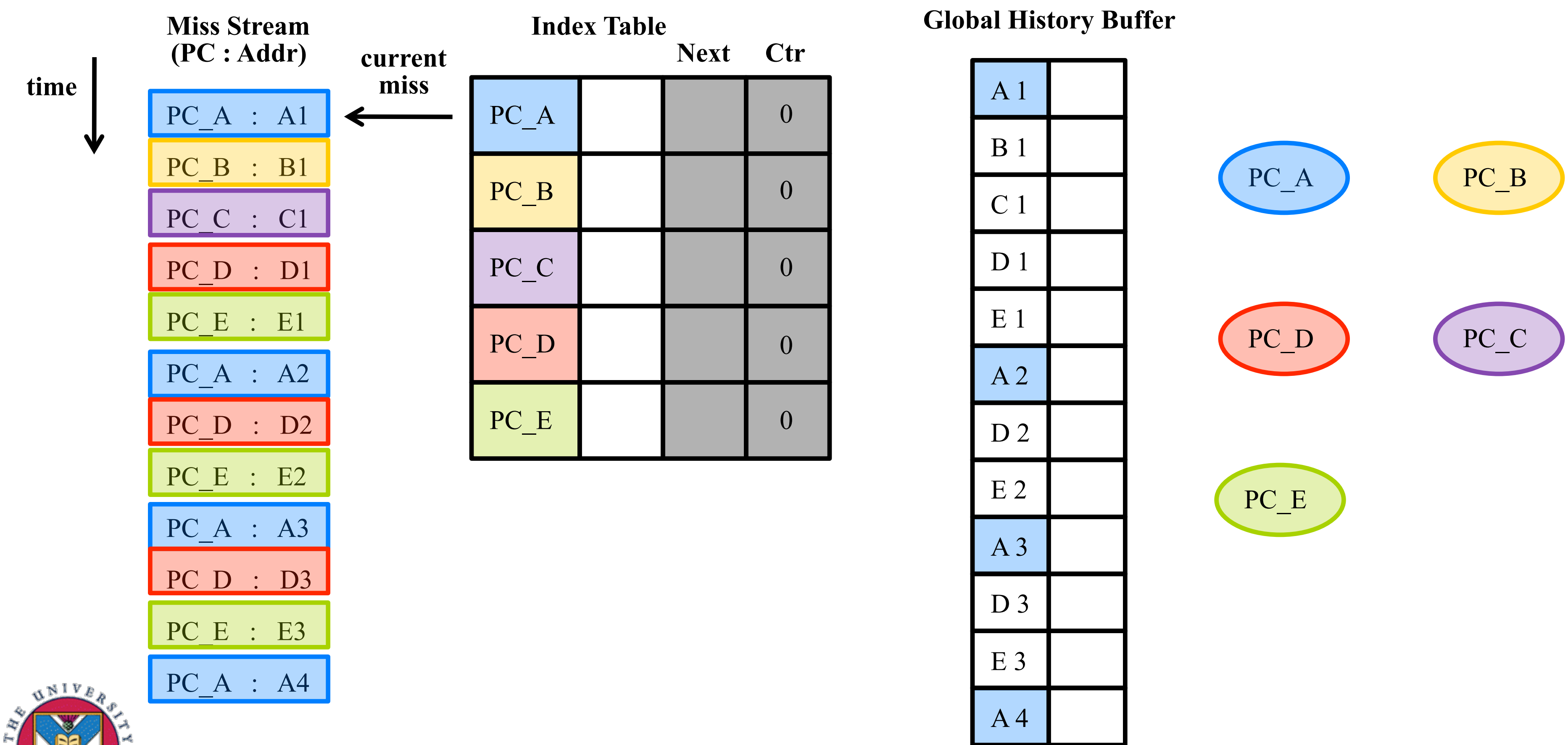
# Miss Graph Prefetcher

- **Step 1: perform localization → already part of GHB funct.**



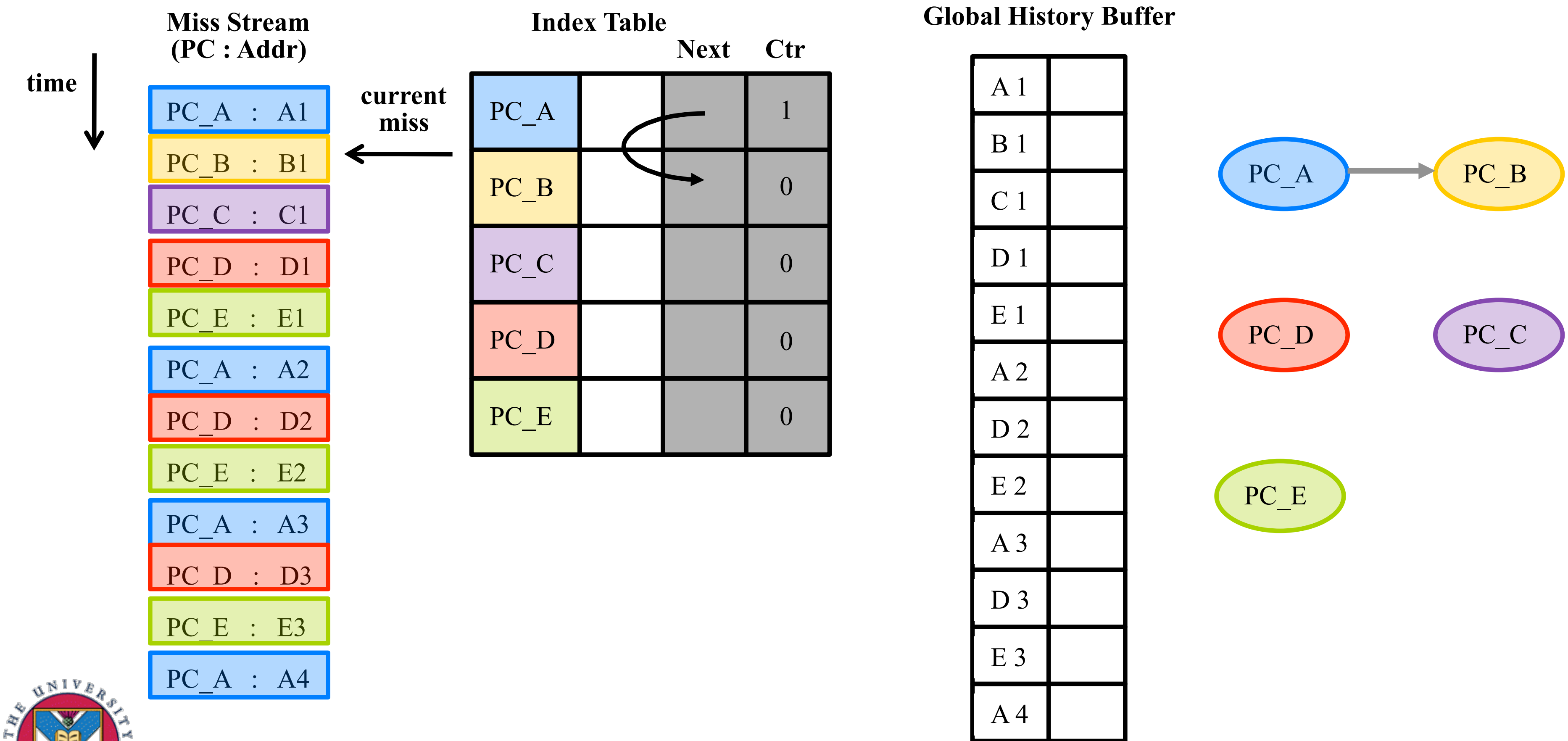
# Miss Graph Prefetcher

- **Step 2: chain streams**



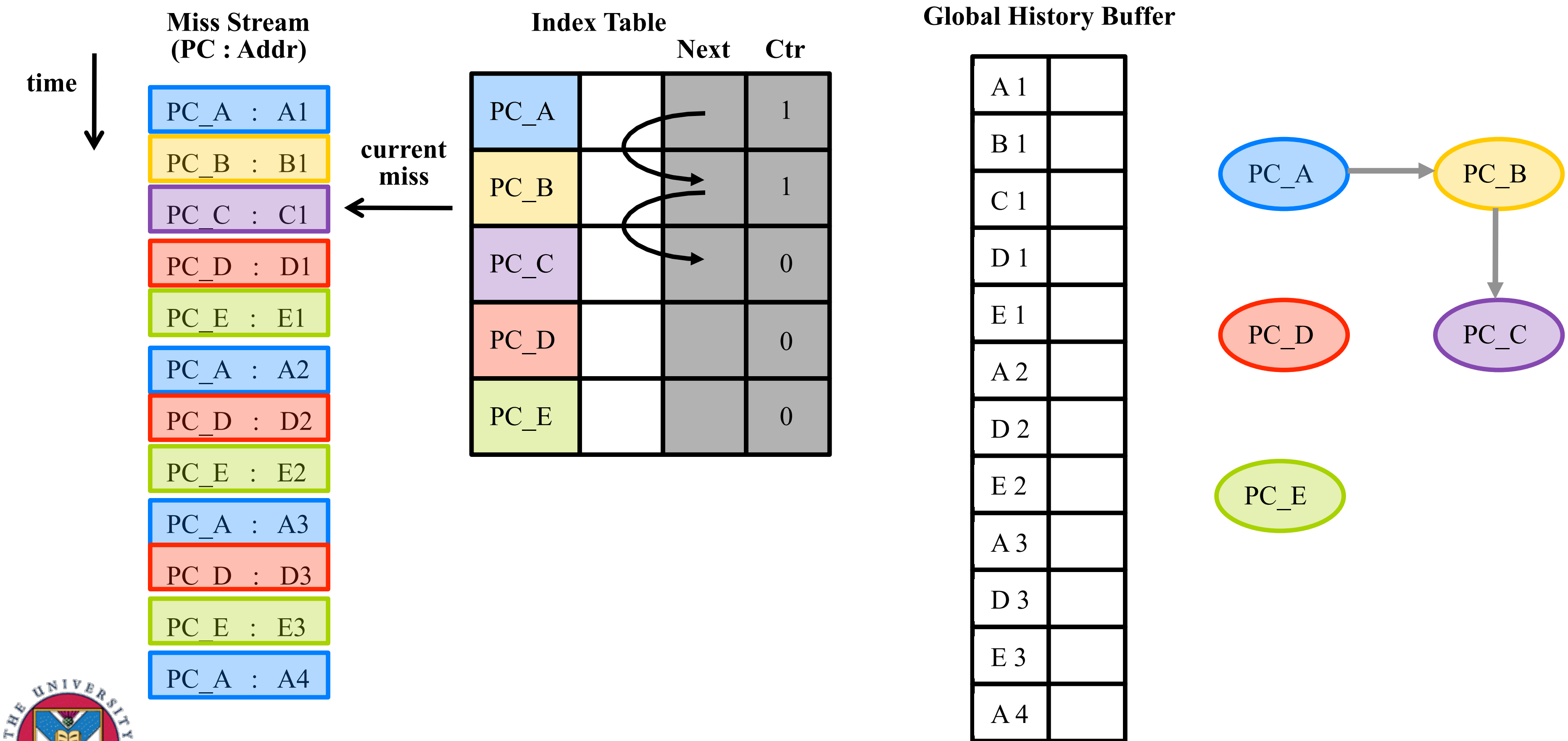
# Miss Graph Prefetcher

- **Step 2: chain streams**



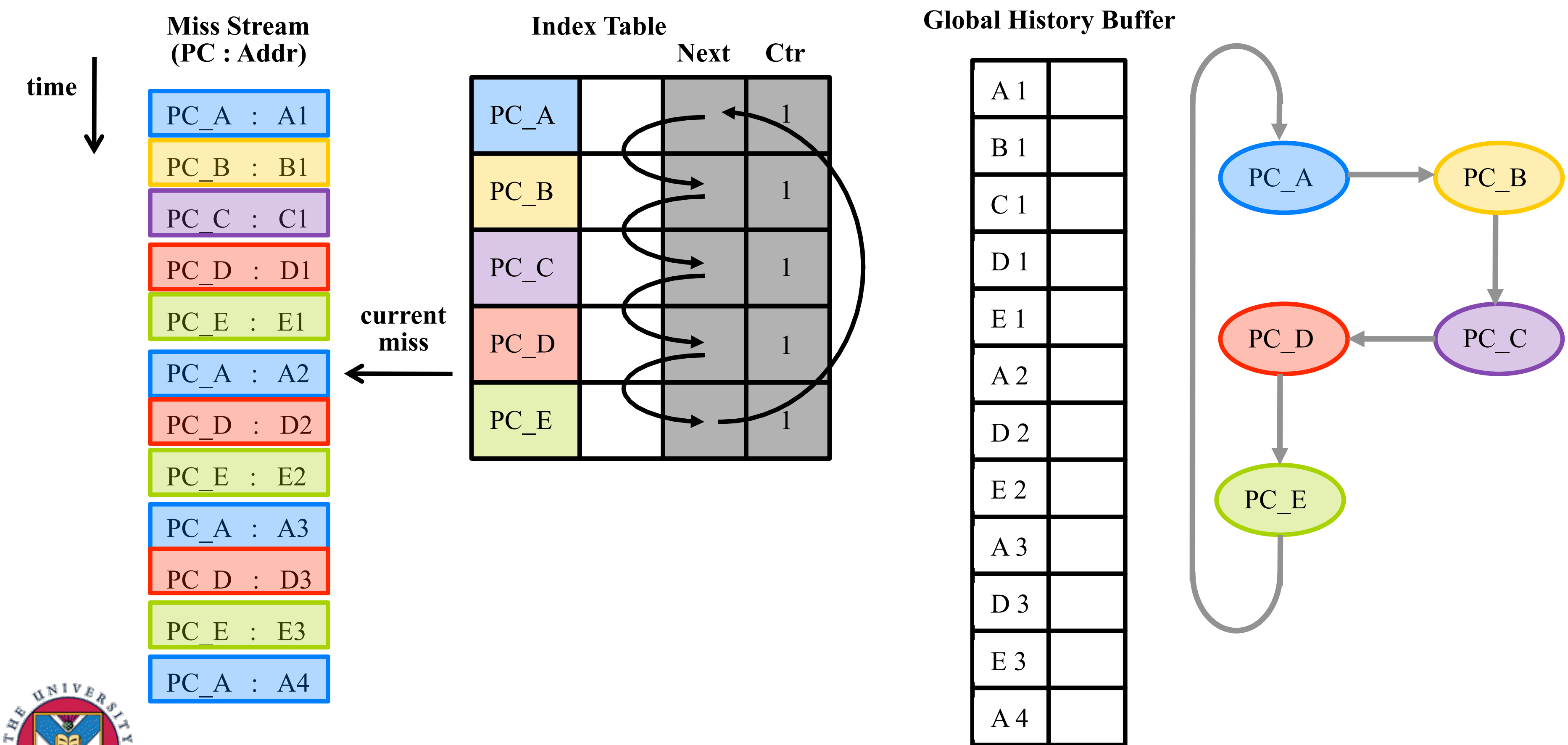
# Miss Graph Prefetcher

- **Step 2: chain streams**



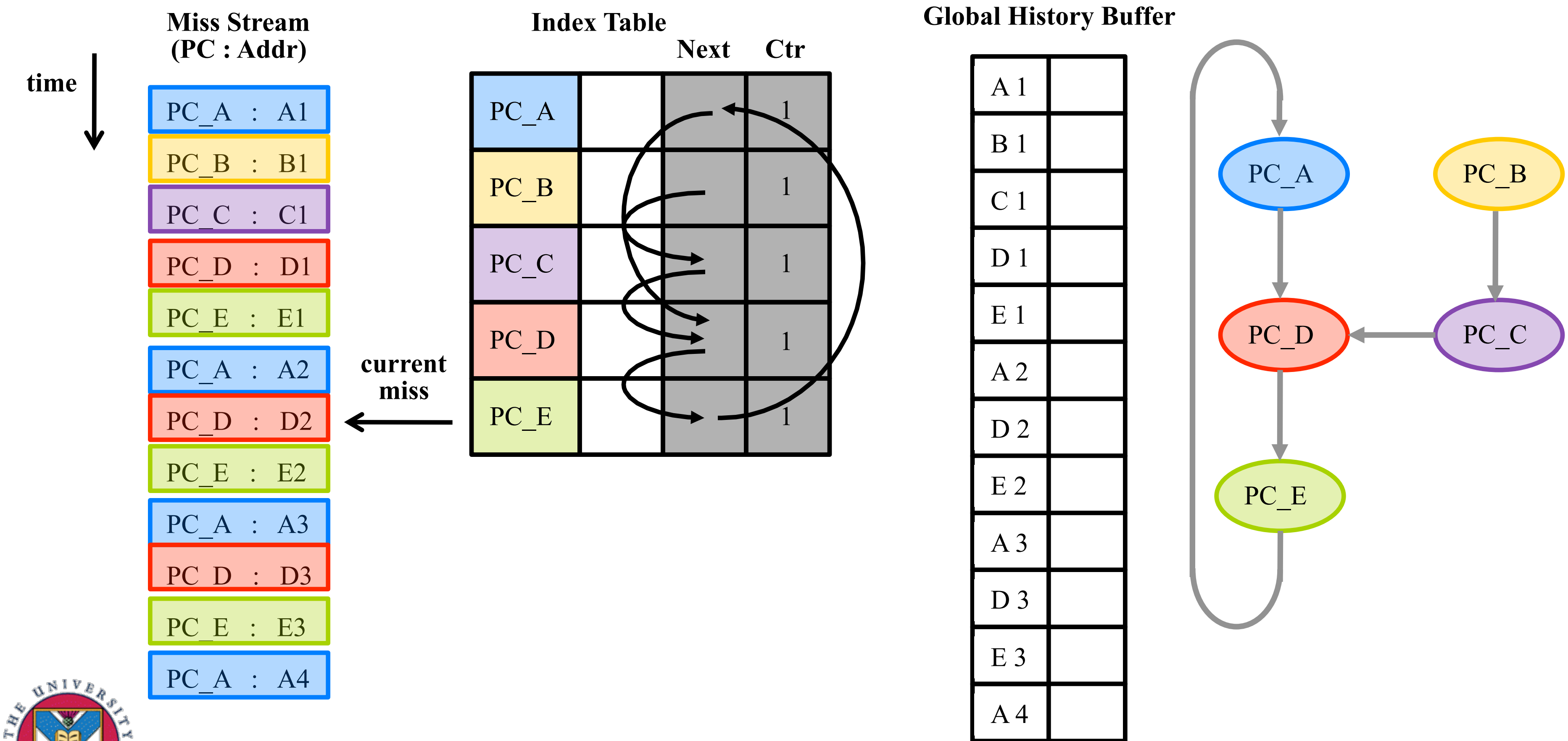
# Miss Graph Prefetcher

- **Step 2: chain streams**



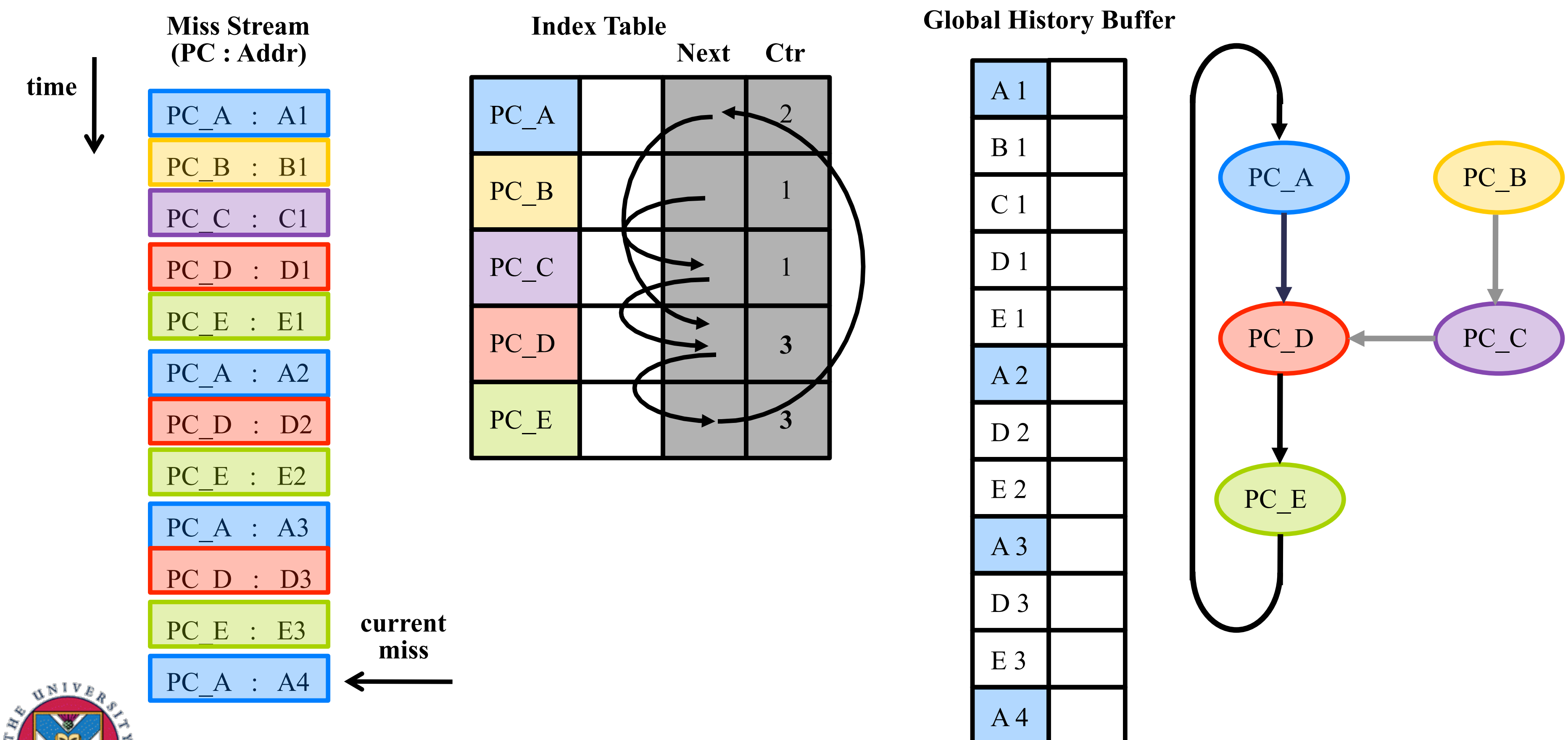
# Miss Graph Prefetcher

- **Step 2: chain streams**



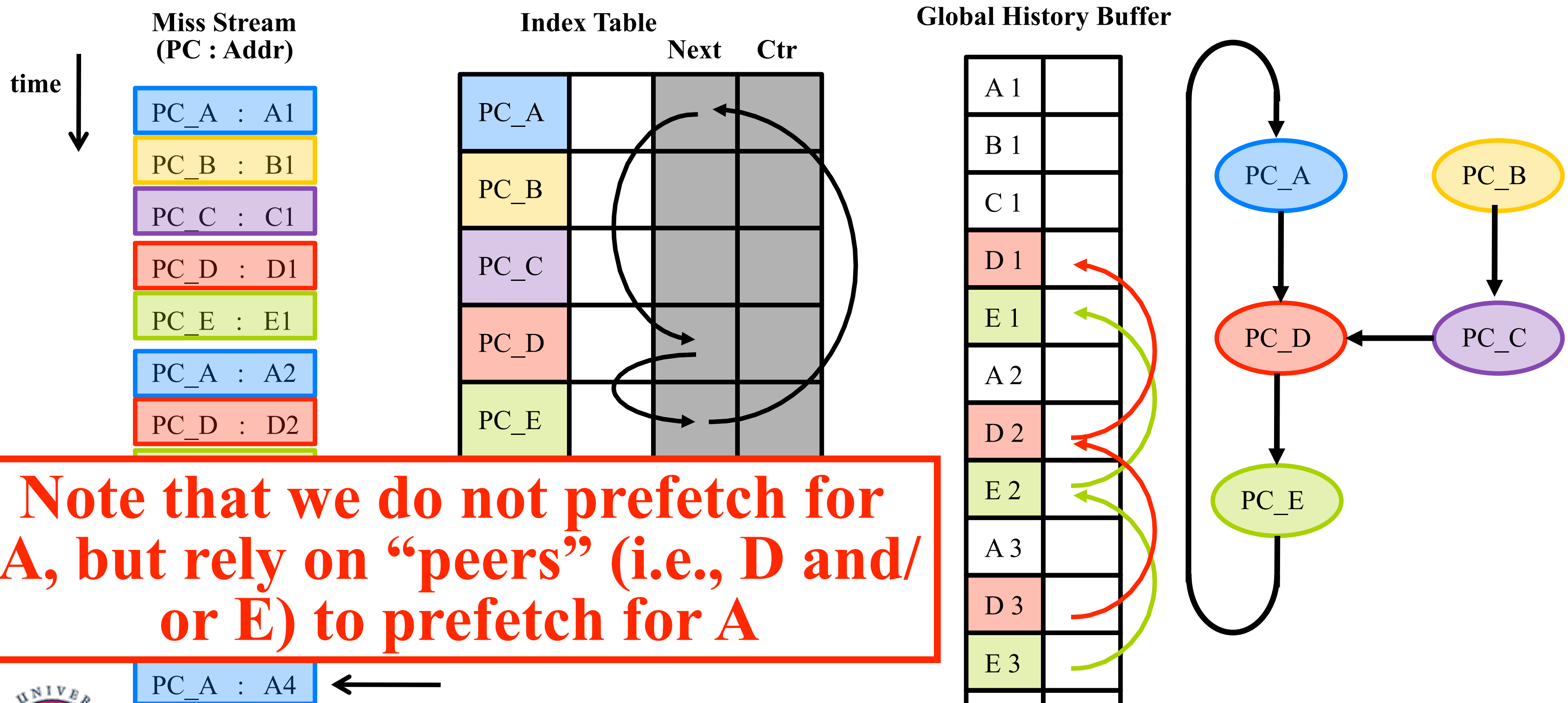
# Miss Graph Prefetcher

- **Step 2: chain streams**



# Miss Graph Prefetcher

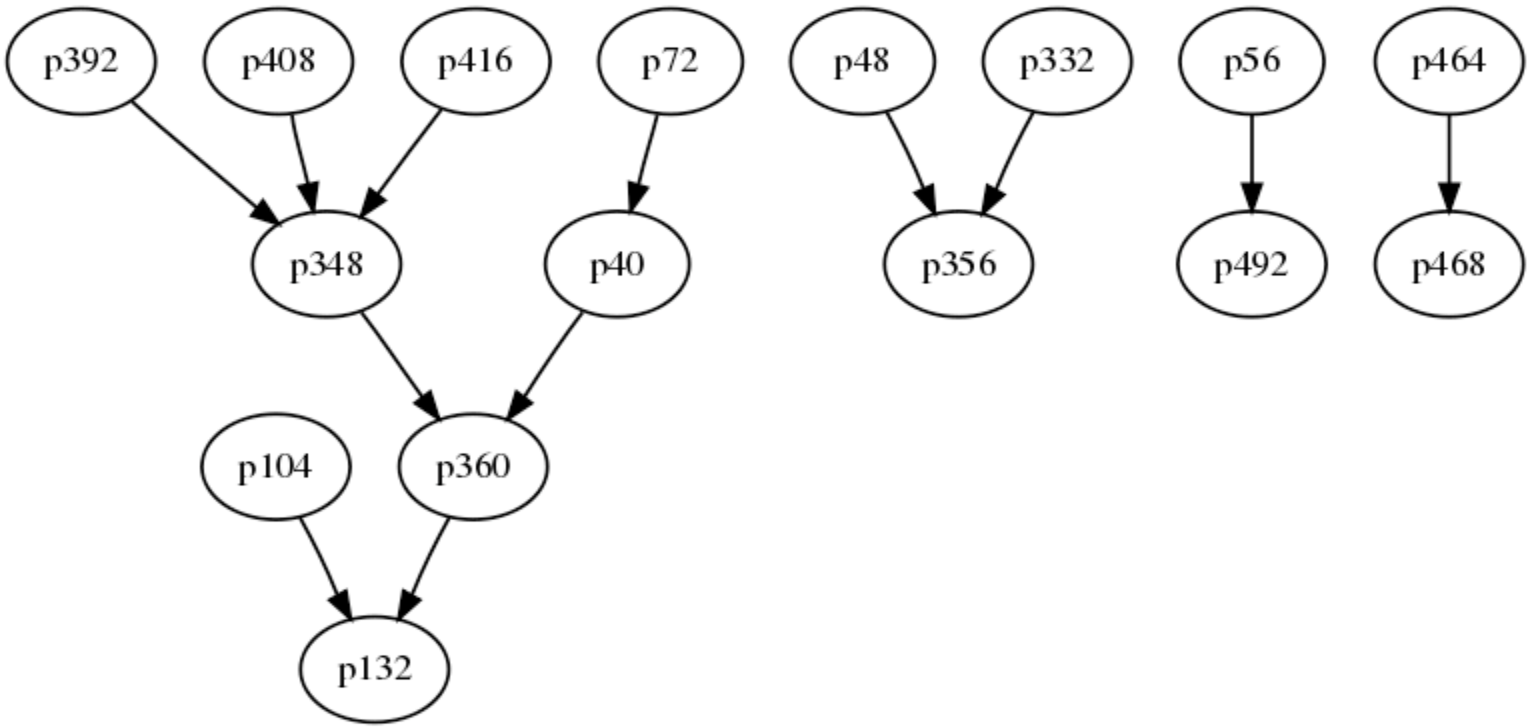
- **Step 3: perform correlations and prefetch along streams**




**Note that we do not prefetch for A, but rely on "peers" (i.e., D and/ or E) to prefetch for A**

# Miss Graph Example

- perlbench (512KB L2)



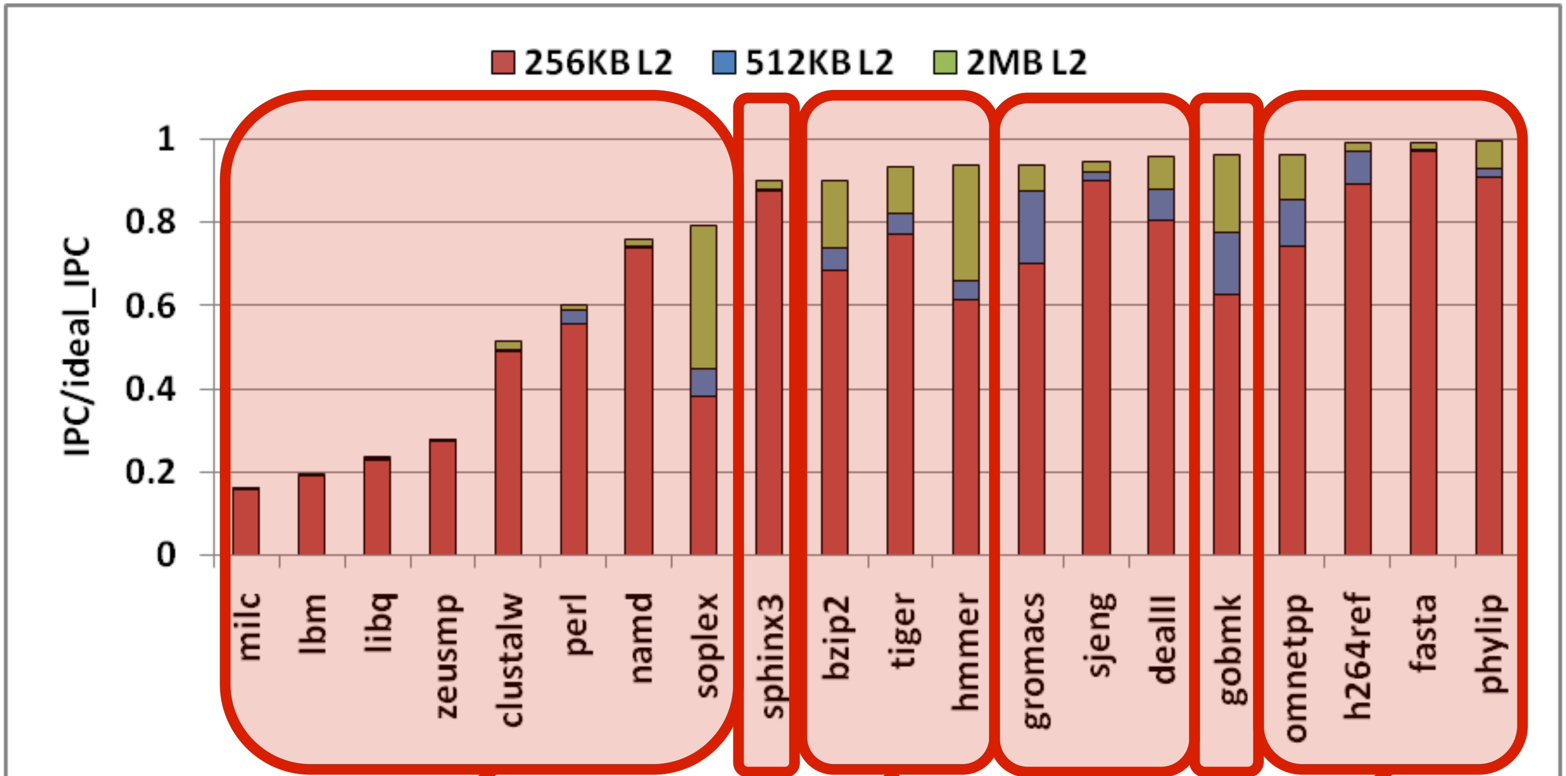
# Outline

- Motivation
- Correlation and Localization
- Stream Chaining and Miss Graph Prefetching
- **Experimental Setup and Results**
- Related Work
- Conclusions

# Experimental Setup

- Simulator:
  - SESC: cycle-accurate architectural simulator from UIUC
- Applications: SPEC2006 and BioBench
- Architecture:
  - 5GHz, 4-issue superscalar MIPS processor
  - 64KB, 2-way L1 I-Cache and 64KB, 2-way L1 D-Cache
  - 256KB/2MB, 8-way L2 cache
  - 64bit, 1.25GHz memory bus
  - Main memory: 400 cycle latency

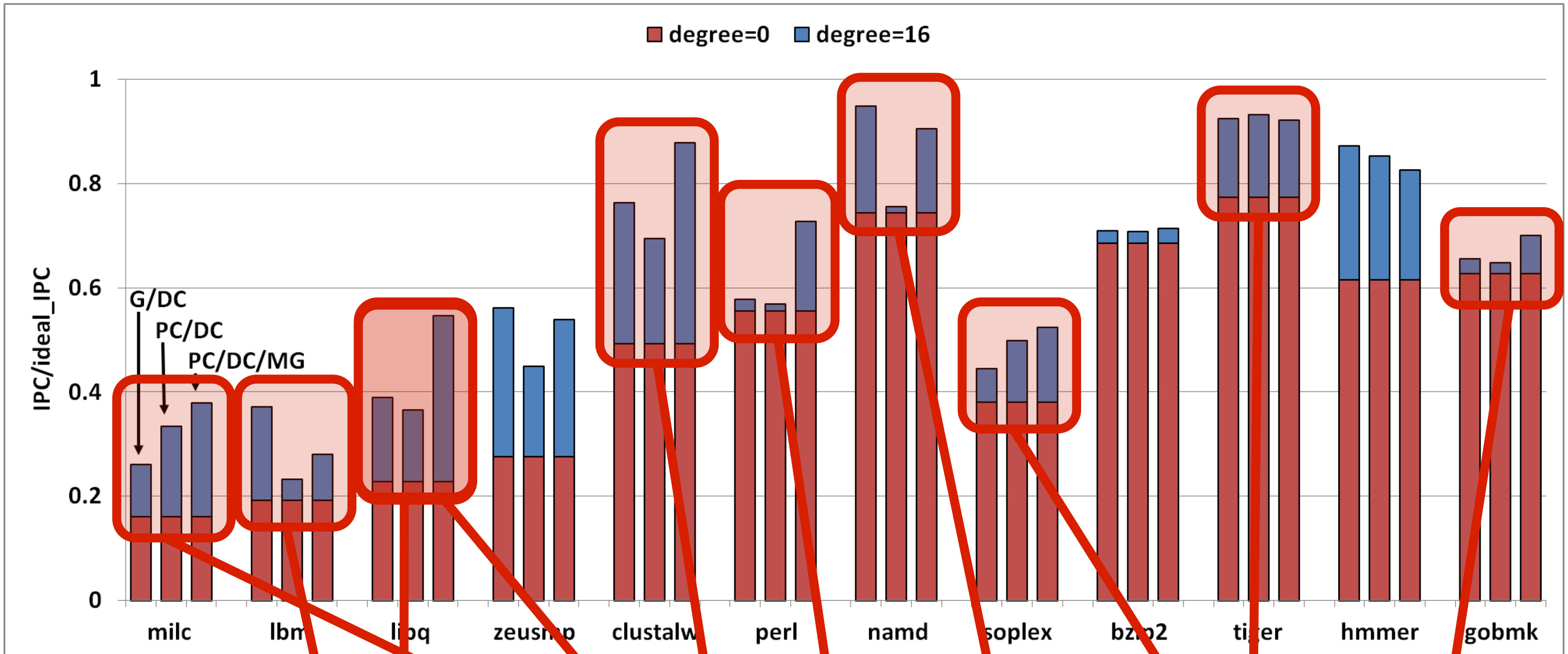
# Performance Without Prefetching




Many applications still perform poorly, some even with large 2MB L2

Some applications already perform well (within 15% of ideal) with 512KB L2

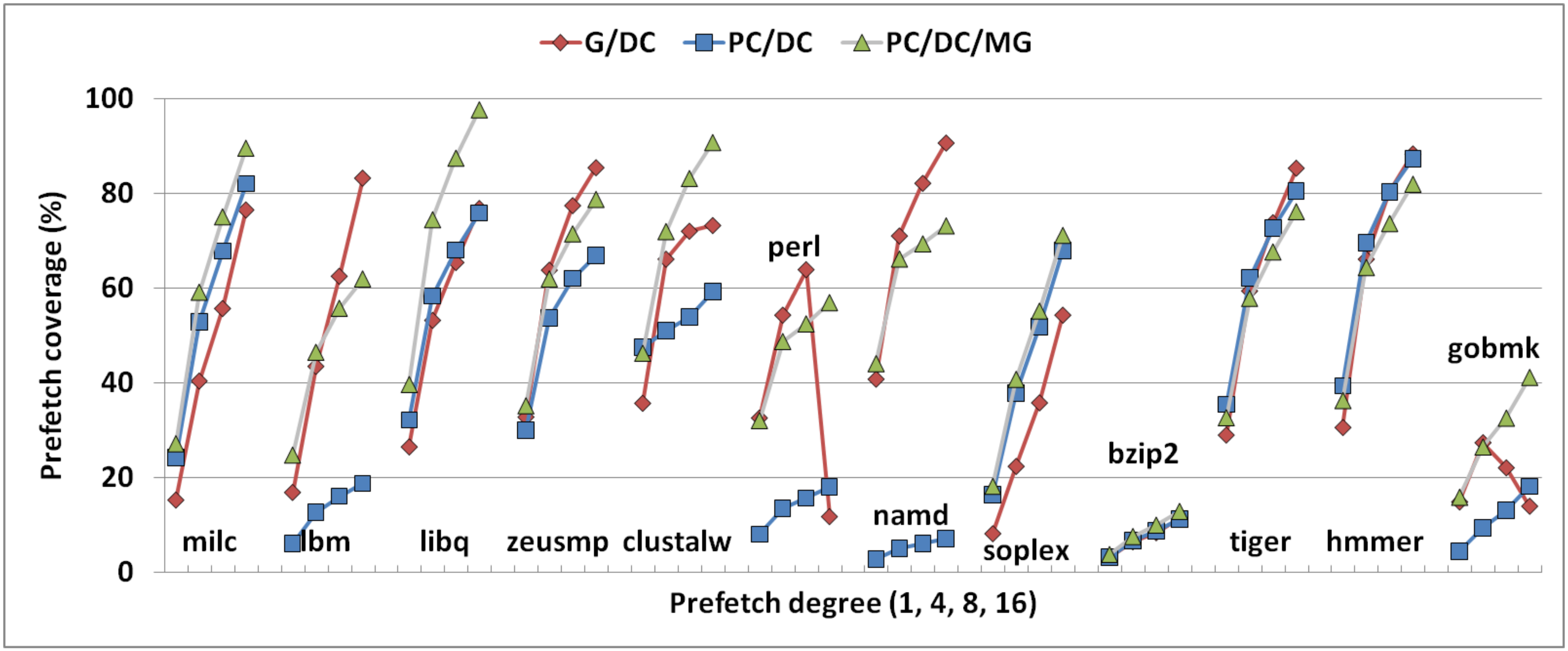
# Performance With Prefetching




Best performing prefetching scheme varies across applications

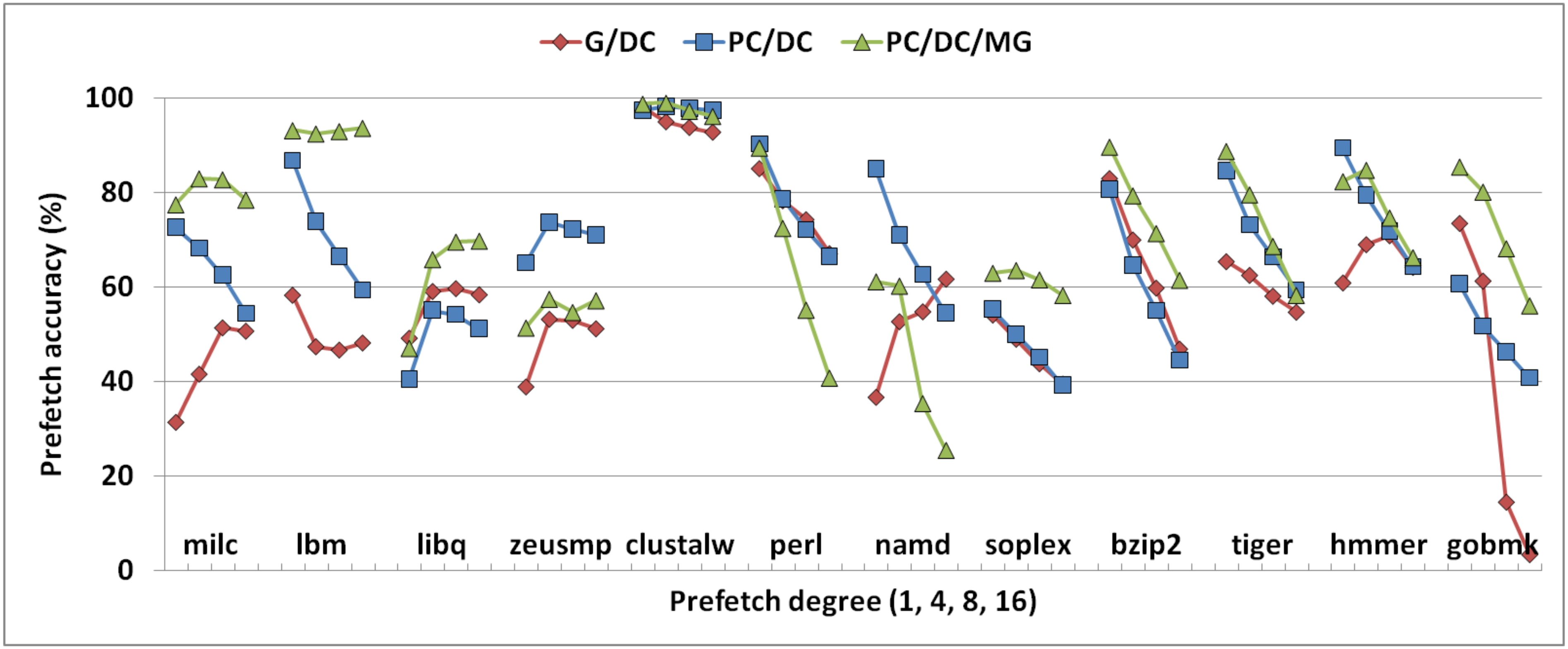
Overall, PC/DC/MG performs best or close to best in most applications

# Prefetch Coverage




**PC/DC often has lowest coverage, and PC/DC/ MG and G/DC vary across applications**

# … and Accuracy




**PC/DC/MG is often the most accurate, and PC/DC is often more accurate than G/DC**

# Miss Graphs Statistics

| Benchmark | Unique Subgraphs (%) | Nodes | | | |
|---|---|---|---|---|---|
| | | Snapshot | | CC | |
| | | max | avg. | max | avg. |
| milc | 4.7 | 15 | 7.7 | 7 | 3.6 |
| lbm | 22 | 20 | 7.9 | 18 | 3.7 |
| lbq | 0.8 | 23 | 19 | 18 | 7 |
| zeusmp | 11 | 18 | 11 | 9 | 4.4 |
| clustalw | 1.1 | 10 | 9.3 | 10 | 8.2 |
| perl | 11 | 16 | 8.6 | 9 | 3.3 |
| namd | 21 | 8 | 5.8 | 8 | 5 |
| soplex | 2.8 | 30 | 12 | 10 | 3.6 |
| bzip2 | 5.6 | 38 | 20 | 9 | 3.8 |
| tiger | 5.4 | 41 | 30 | 18 | 4.2 |
| hmmer | 12 | 50 | 38 | 33 | 5.4 |
| gobmk | 20 | 10 | 5.2 | 5 | 3.4 |

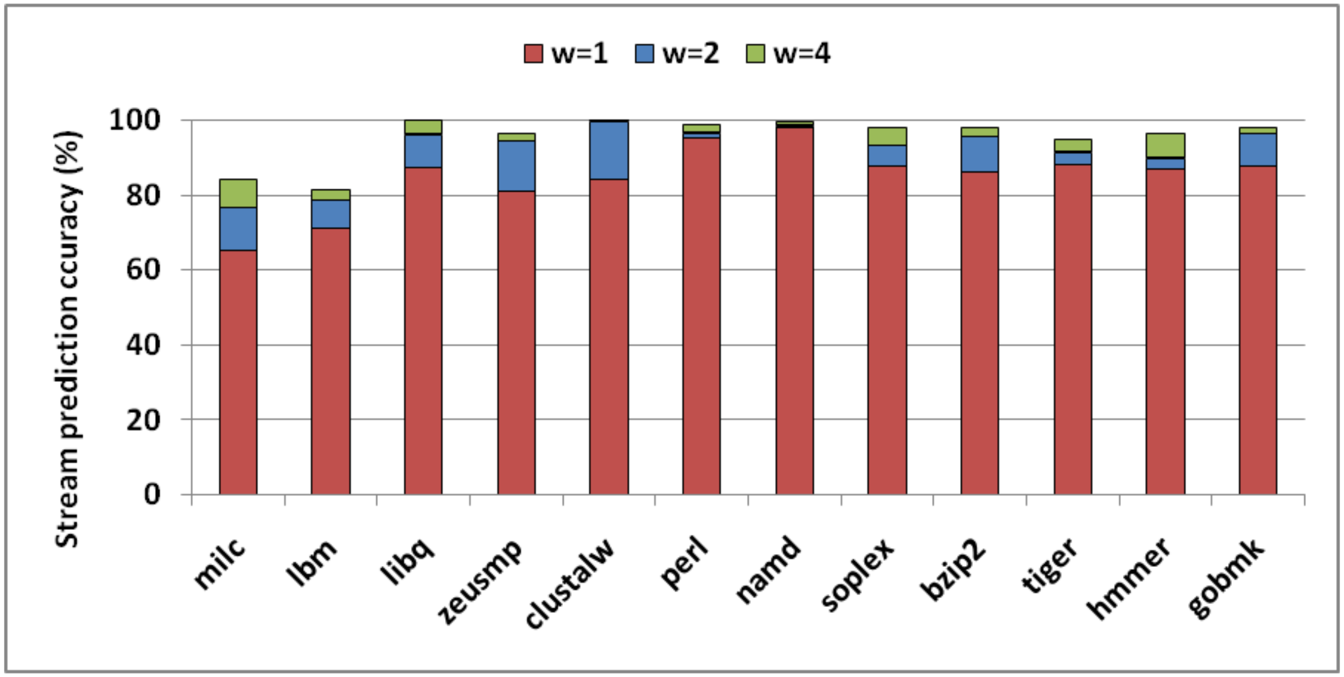
**Most graphs appear repeatedly during execution → potential for learning**

**Moreover (results not shown) graphs are stable for long periods of time → potential to exploit patterns**

**Number of nodes at any given time is small → manageable to keep track of**

**Number of nodes per stream groups is small → small protocol execution overheads**

# Next-Stream Prediction Accuracy




**Miss-graph's prediction accuracy is often very high**

# Outline

- Motivation
- Correlation and Localization
- Stream Chaining and Miss Graph Prefetching
- Experimental Setup and Results
- **Related Work**
- Conclusions

# (Closest) Related Work

- K. Nesbit and J. Smith – HPCA'04
  - Proposed GHB and introduced PC/DC
- S. Somogyi, T. Wenisch, A. Ailamaki, and B. Falsafi – ISCA'09
  - Combined spatial and temporal memory streaming
  - Can be seen as close to a PID/SMS/TMS prefetcher (except that PID is not used to index at prefetch time)

# Outline

- Motivation
- Correlation and Localization
- Stream Chaining and Miss Graph Prefetching
- Experimental Setup and Results
- Related Work
- **Conclusions**

# Conclusions

- New strategy for creating prefetchers by composing (chaining) localization and correlation schemes
- New prefetcher based on the Stream Chaining idea
  - Simple extension of GHB-based PC/DC of Nesbit and Smith (HPCA'04)
  - Captures most of the stable miss sequences in the programs tested
  - Overall better performance than PC/DC or G/DC
- Stream Chaining could be applied to other localization and correlation schemes (we are working on it)

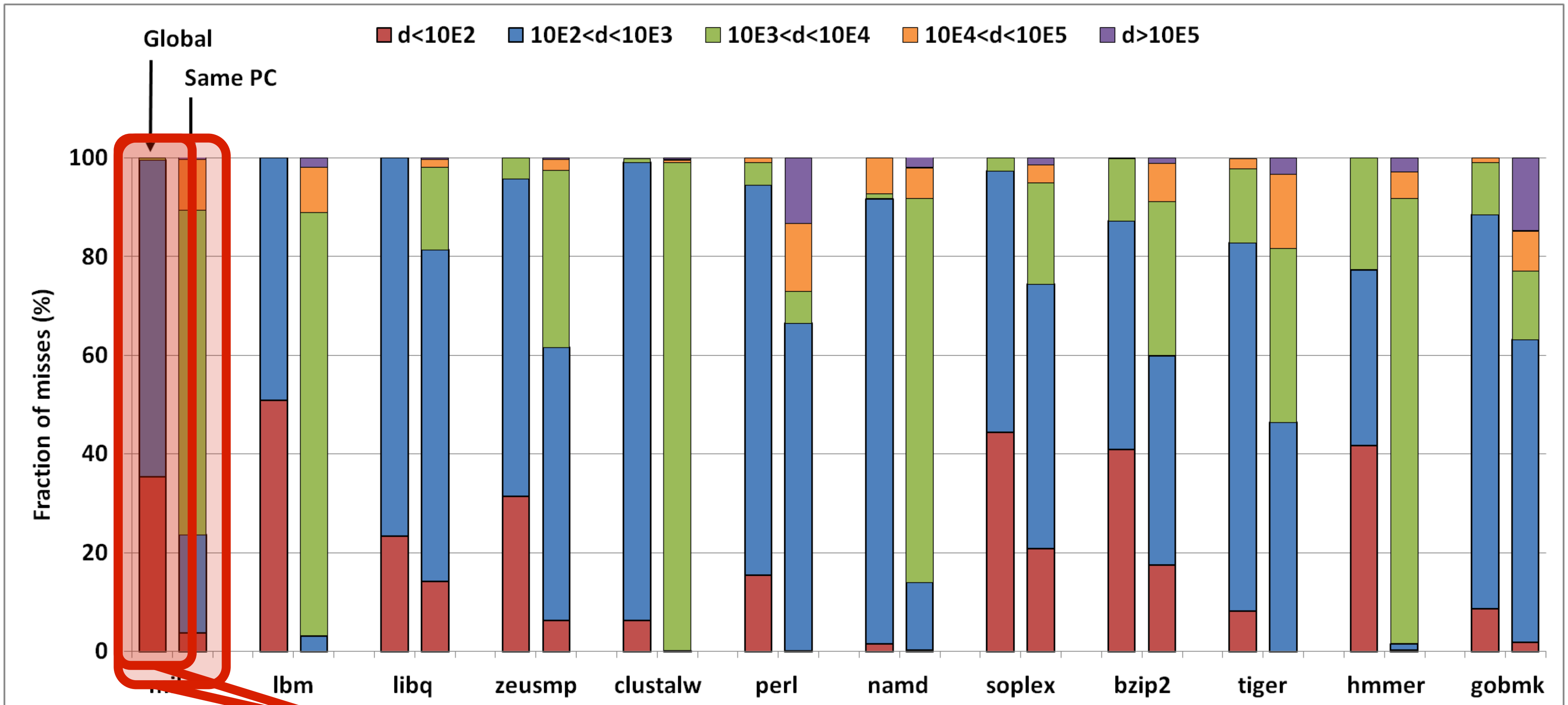
# Stream Chaining: Exploiting Multiple Levels of Correlation in Data Prefetching

Pedro Díaz and Marcelo Cintra

University of Edinburgh

http://www.homepages.inf.ed.ac.uk/mc/Projects/CELLULAR

# Miss Distances

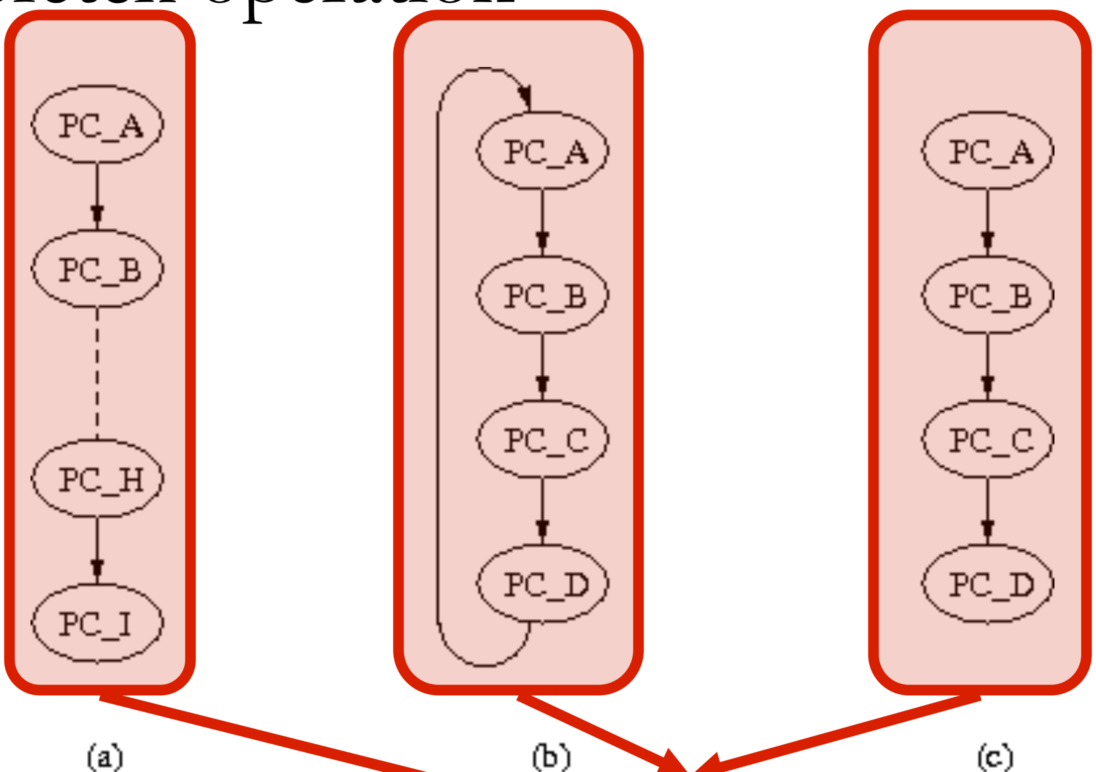



**Global miss distances are often in the order of tens or hundreds of cycles only**

**PC localized miss distances are always much larger, often in the order of thousands or tens of thousands**

# Miss graph prefetching

- Prefetch operation

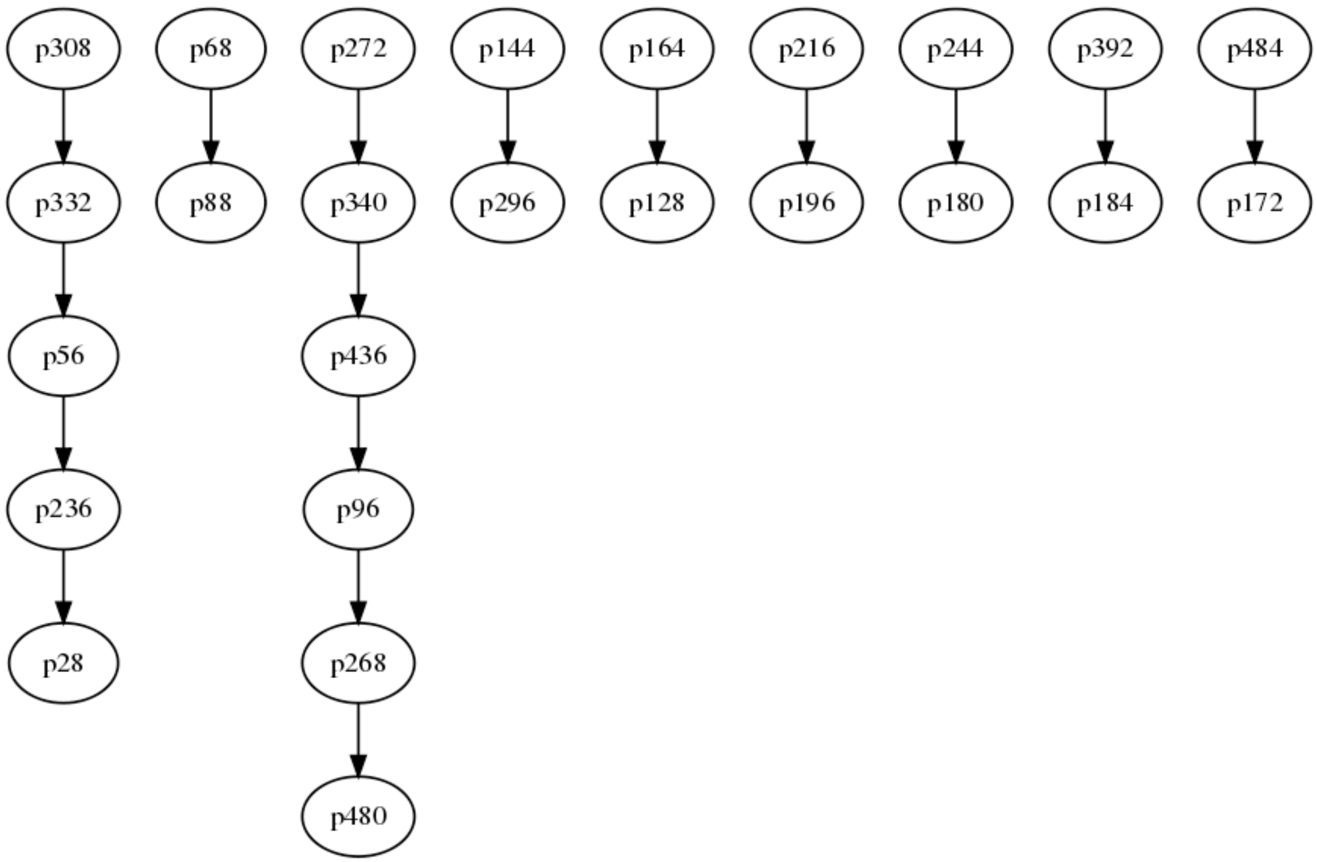



Long enough, linear stream: Prefetch 1 item from PC_B onwards

Not long enough or cyclic chains: Prefetch degree/length items per stream

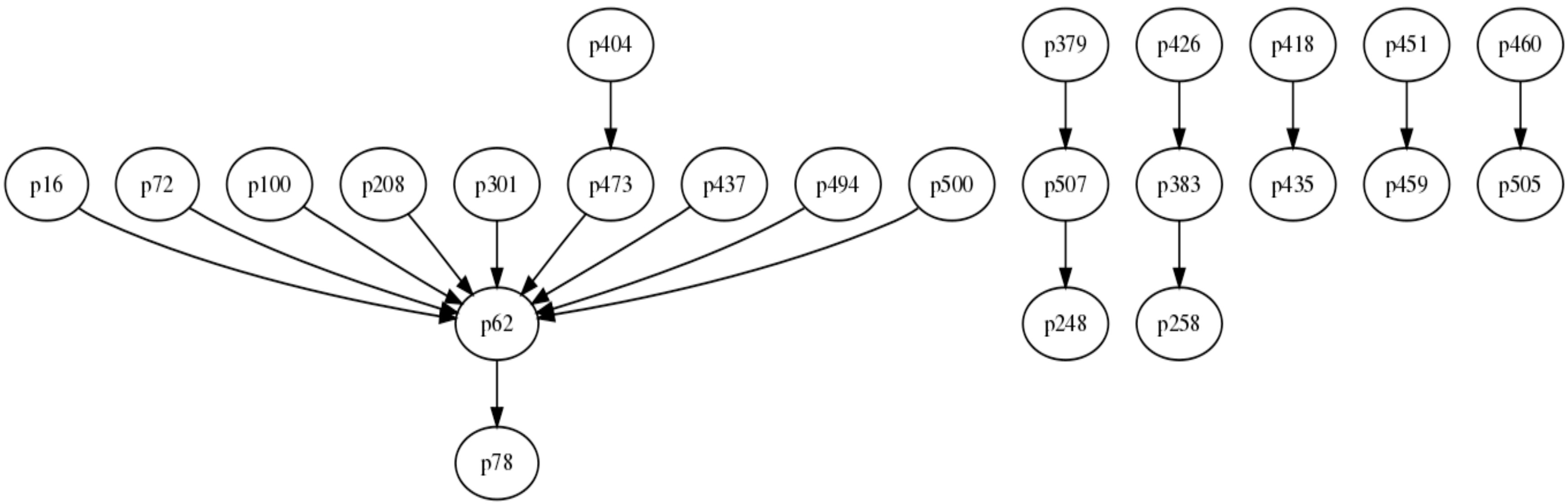
# Miss Graph examples

- bzip2 (2048KB L2)



# Miss Graph examples

- lbm (512KB L2)



# Miss Graph examples

- libquantum (256KB L2)