MANA: Microarchitecting an Instruction Prefetcher

Ali Ansari (Sharif)
Fatemeh Golshan (Sharif)
Pejman Lotfi-Kamran (IPM)
Hamid Sarbazi-Azad (Sharif, IPM)
Instruction Cache Misses

- Server applications
  - Multi-megabyte instruction footprint
  - 25% increase in size per year [Kanev, ISCA’15]

- Limited capacity L1 instruction cache
  - 512 blocks, 32 KB

Frequent L1i misses hurt performance!
Prior Work

Significant storage cost or uncovered potential!
Contributions

• Storage cost is important
  o Unlimited storage results in high speedup

• Prefetching records
  o A few distinct records
  o Low storage demand per record

• MANA
  o 4 K distinct prefetching records, on average
  o Each record ≈ 4 bytes
  o 24% and 26.6% speedup with 16.3 and 122 KB

MANA offers considerable speedup with a limited storage!
Outline

• Introduction
• Motivation
• Our Proposal, MANA Prefetcher
• Methodology
• Evaluation
• Conclusion
Motivation

• Spatial region
  ○ Trigger address + a footprint

• Advantages
  ○ Covering a large address space
    ▪ Few distinct prefetching records
  ○ Easily detectable
    ▪ Simple design

• Widely used in prior work
  ○ PIF [Ferdman, MICRO’11]
  ○ RDIP [Kolli, MICRO’13]
  ○ Shotgun [Kumar, ASPLOS’18]

Spatial region is a good prefetching record!
Motivation (cont.)

- Spatial region’s challenges:
  - Finding the successor, why?
    - Prefetching the trigger block
    - Timeliness
  - Storage cost
    - Trigger address = block address!

- Prior work cannot solve these challenges effectively
- MANA offers simple solutions for them

MANA microarchitects the use of spatial regions!
MANA

• Spatial region is the main prefetching record
  ○ No association with other events

• MANA_Table
  ○ A set-associative table to hold spatial regions
  ○ Looked up by trigger addresses

• Finding the successor
  ○ The sequence of spatial regions is repetitive (PIF)
  ○ Use a pointer to the successor spatial region
  ○ Chase the pointers to discover successor spatial regions

MANA: (Spatial region + a pointer) in a set-associative table!
MANA: High-Order Bit Patterns
MANA: High-Order Bit Patterns

- HOBP
- Partial Tag
- Set Number
- Block Offset

Instruction Address
MANA: High-Order Bit Patterns

HOBP

Partial Tag

Set Number

Block Offset

Instruction Address

HOBPs’ Table

100

0xffa358f12b

HOBP index

Partial Tag

100

b’01
MANA: Replaying

Sequence: A, A+1, B, A+2, B+1, C, D

L2 Cache

A, A+1, A+2
B, B+1

C

D, D+1

0

1

2

3

SAB

A,11  B,10

A,11  B,10  C,00

B,10  C,00  D,10

MANA_Table

... 

C 

B 

D 

A 

...
Methodology

- ChampSim Simulator
- Default parameters
- 32 KB, 8-way, L1 instruction cache
- 50 public traces
- Warmup: 50 M instructions
- Evaluation: 50 M instructions
- Competitors: RDIP, Shotgun, and PIF
Evaluation

Better performance in all given storage budgets!
MANA can effectively prefetch for small cache sizes!
Conclusion

- MANA uses spatial regions
- Spatial regions are chained with pointers to each other
- HOBP is used to reduce the storage cost
- 24% speedup with only 16.3 KB
  - Significant gap with prior work
  - More practical design
- 26.6% speedup with 122 KB
Thank You!

Any Questions?