Blasting Through The Front-End Bottleneck With Shotgun

Rakesh Kumar, Boris Grot, Vijay Nagarajan

NTNU

THE UNIVERSITY of EDINBURGH

ARM Research Summit 2018

NTNU

THE UNIVERSITY of EDINBURGH

The Front-End Problem

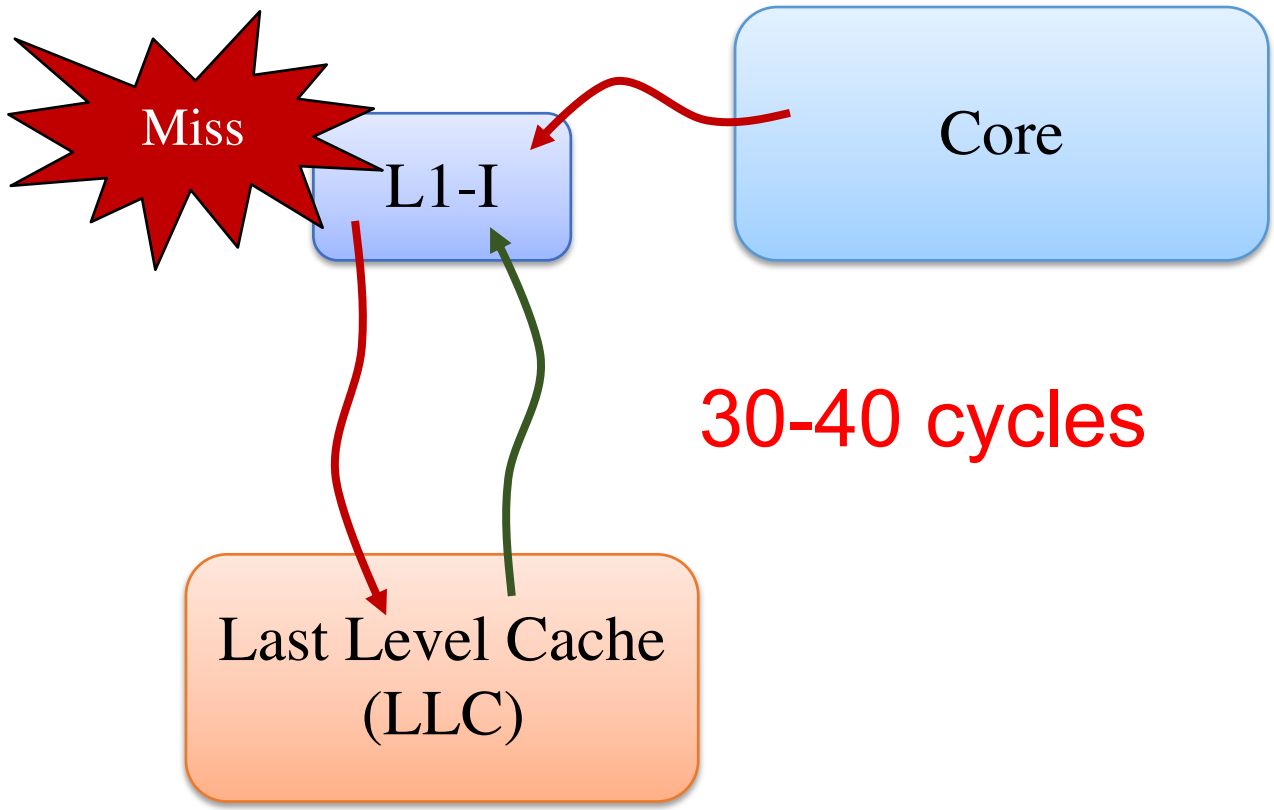
Traditional and emerging server applications

- Deep SW stacks, complex functionality
- Huge instruction working set size
  - Multiple megabytes
  - Growing 25% per year at Google [Kanev ISCA'15]

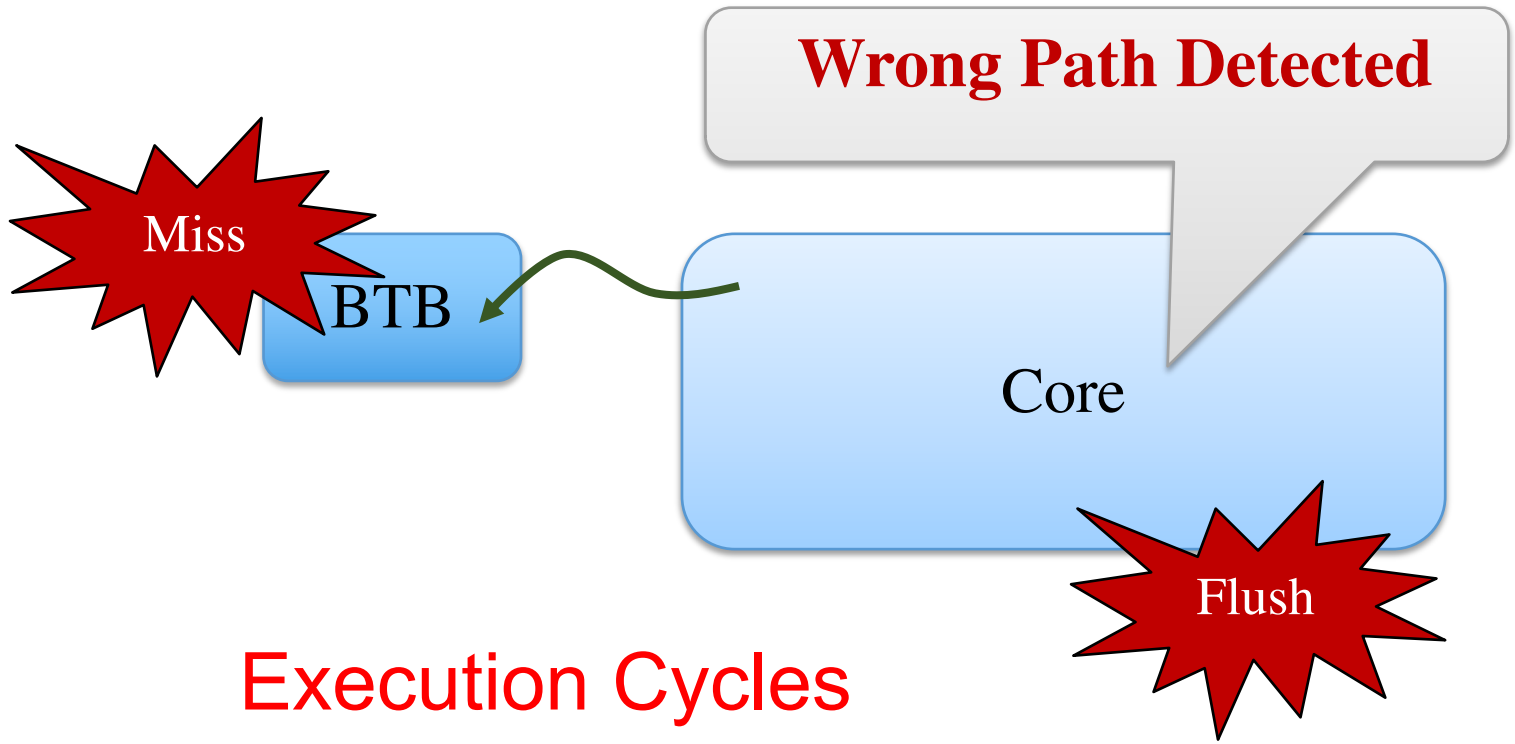
Frequent L1-I and BTB Misses

- Working sets don't fit in latency-critical components

Front-End Problem #1: L1-I Misses



Front-End Problem #2: BTB Misses



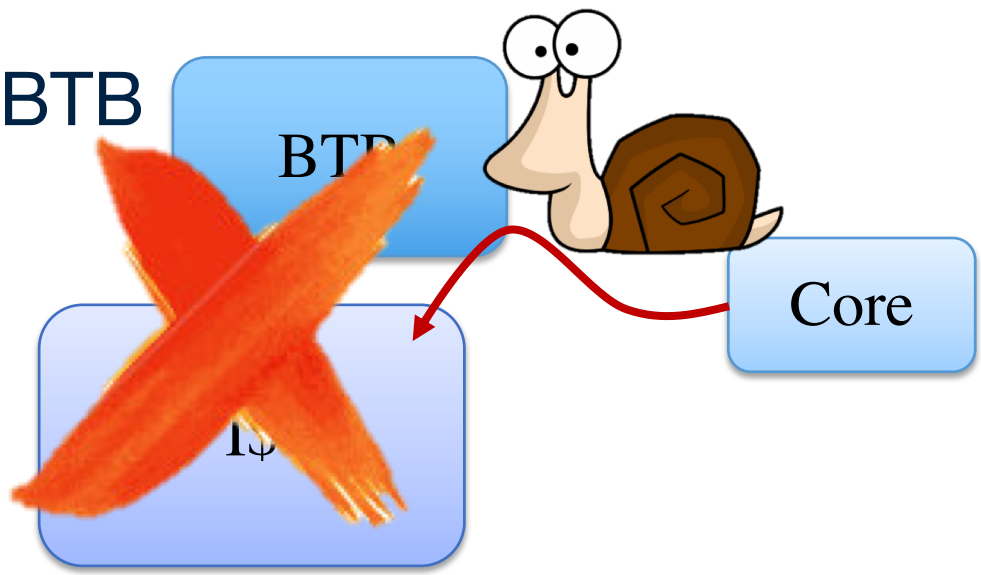
Execution Cycles

0+

Overcoming the Front-End Bottleneck: What are the Options?

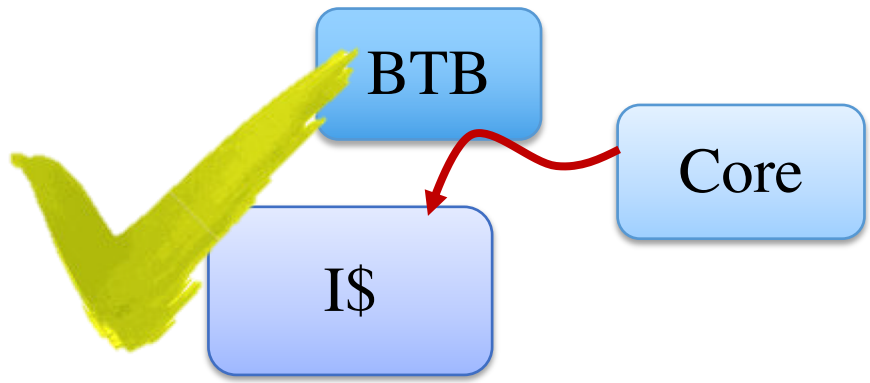
Bigger or multi-level caches, BTB

- High access latency hurts performance



Prefetching

- No impact on L1-I & BTB access latency and area



NTNU

THE UNIVERSITY of EDINBURGH

Outline

Motivation

Existing front-end prefetching approaches

- Temporal Stream: High storage ☹

- BTB-directed: Low Performance ☹

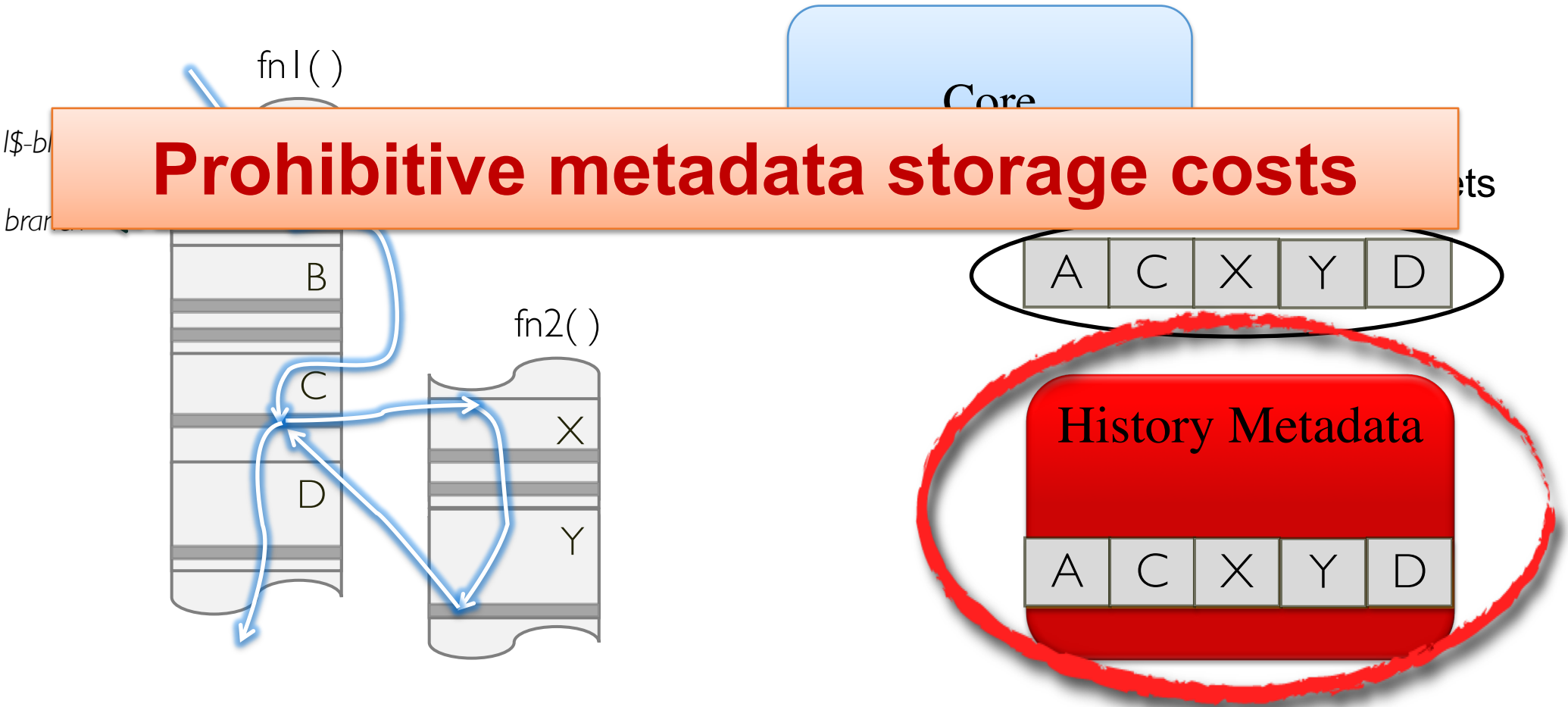
Shotgun: Low Storage and High Performance ☺

Summary

Temporal Stream Prefetching

Principle: Record and Replay

Confluence [MICRO'15]

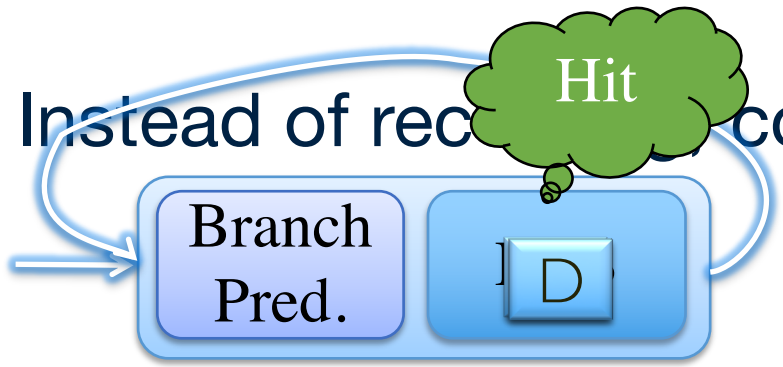


Prohibitive metadata storage costs

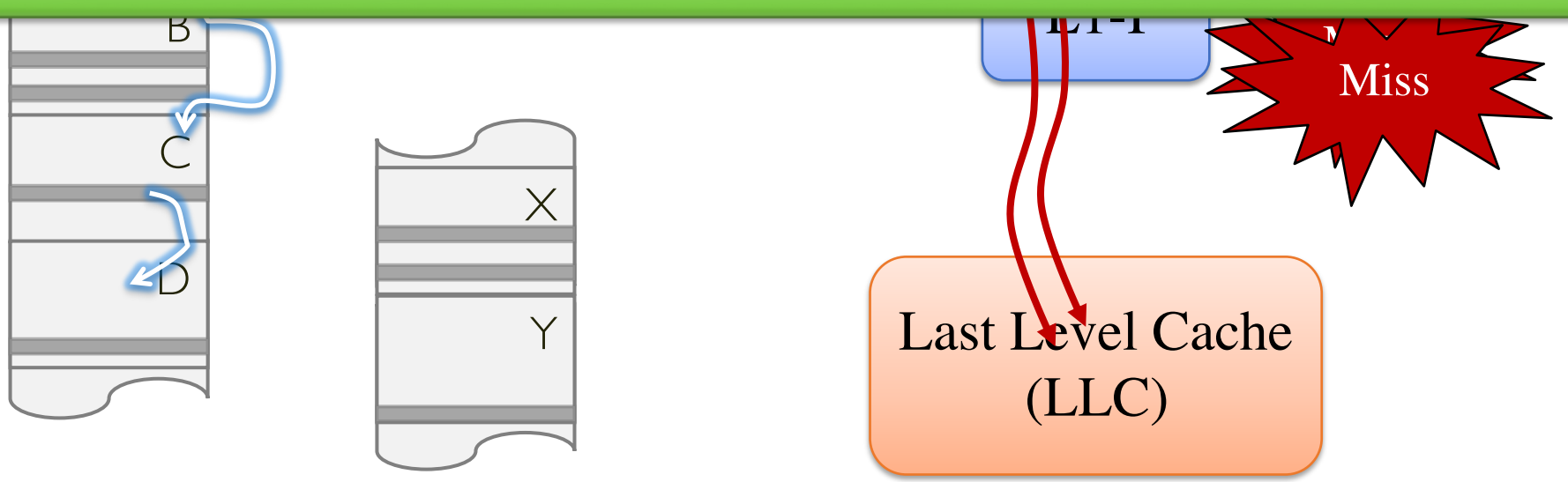
BTB-directed Prefetching

Boomerang [HPCA'17]

Idea: Instead of rec construct the control flow

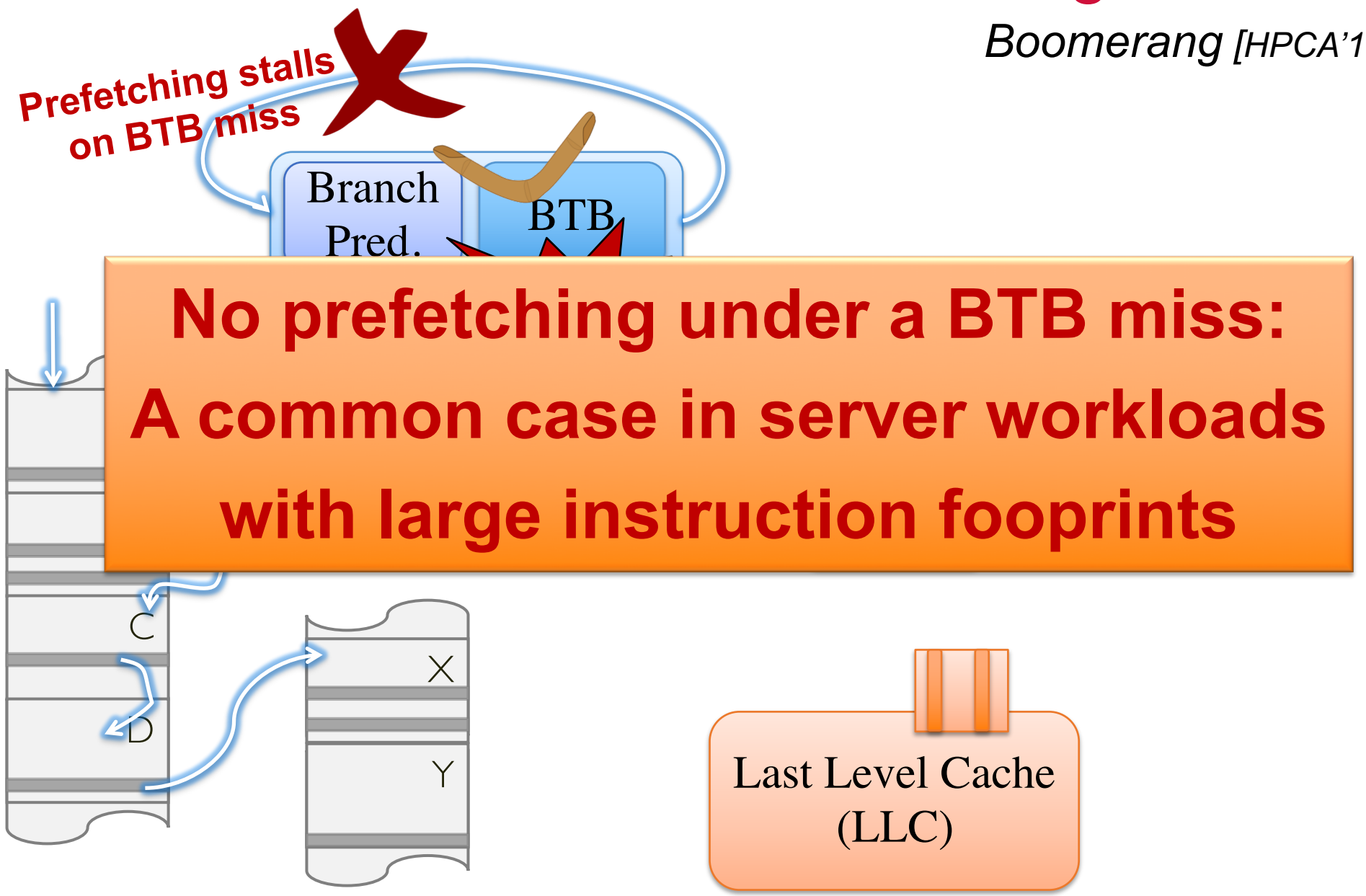


Control flow construction for prefetching without metadata cost



BTB-directed Prefetching

Boomerang [HPCA'17]



No prefetching under a BTB miss:
A common case in server workloads with large instruction fooprints

NTNU

THE UNIVERSITY of EDINBURGH

Our Goal

	Performance	Storage
Temporal Stream Prefetchers		
BTB-directed Prefetchers		

NTNU

THE UNIVERSITY of EDINBURGH

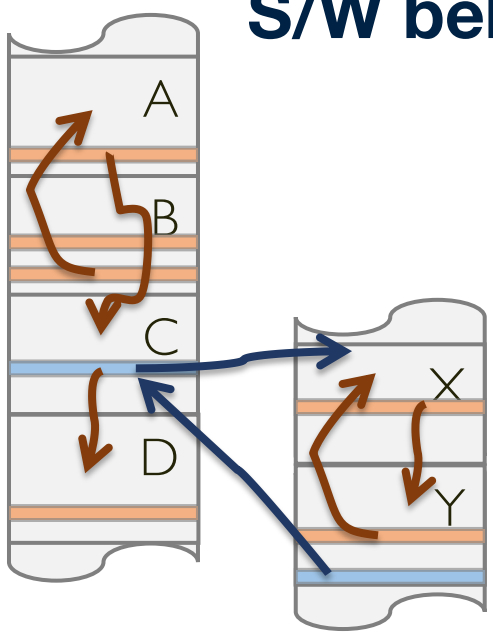
BTB-directed Prefetching: Another Look

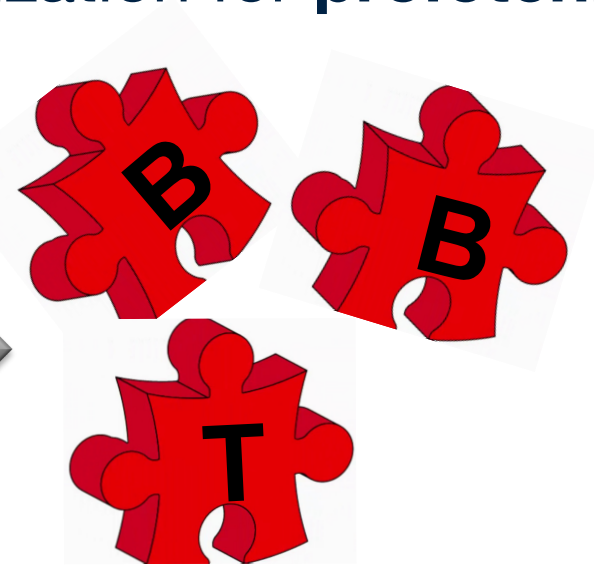
Problem: Conventional BTB cannot accommodate the branch working set of server workloads

- BTB misses stall prefetching

Objective: Improve BTB control flow coverage

Approach: Rethink BTB organization for **prefetching** leveraging **S/W behavior**

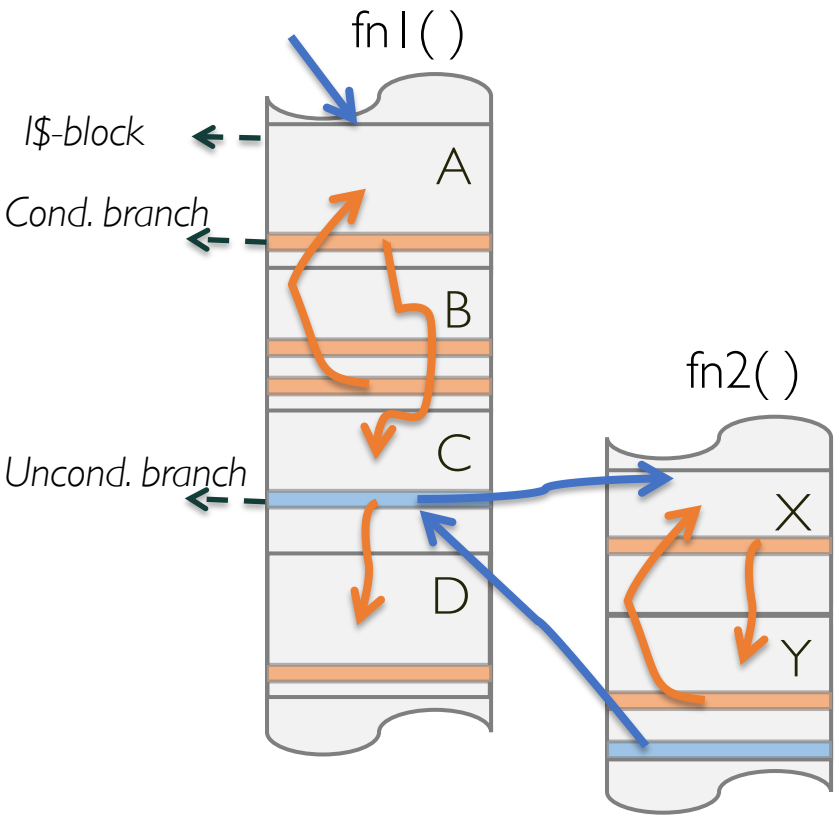




NTNU

THE UNIVERSITY of EDINBURGH

Understanding Control Flow Behavior



Global Control Flow

- Control flow between distinct code regions (e.g. functions).
- Comprised of **unconditional** branches
  - calls, returns, traps,…

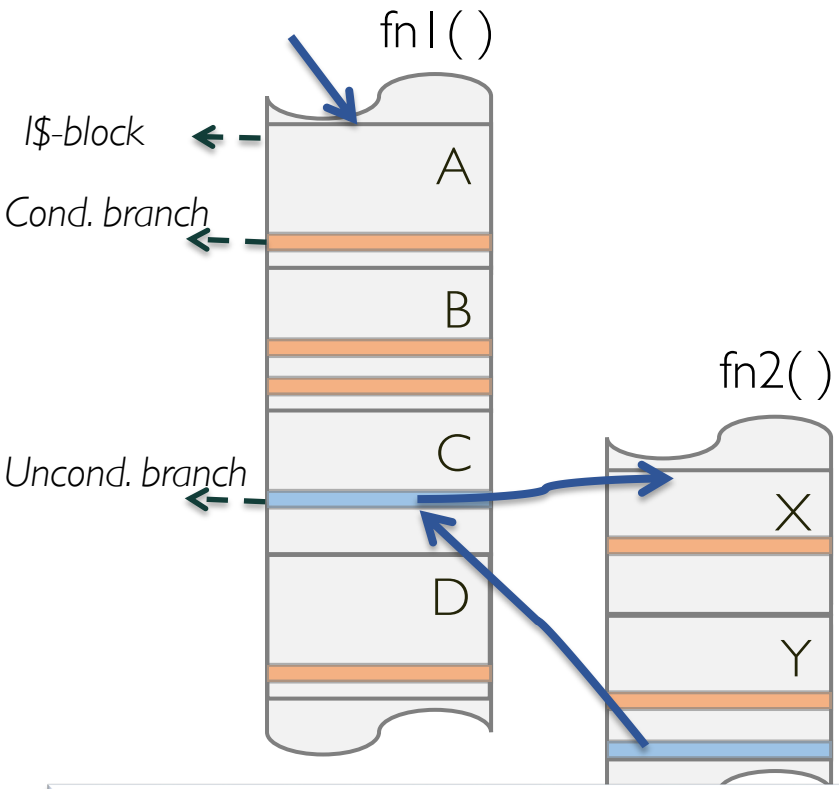
Local Control Flow

- Inside a code region
- Comprised of **conditional** branches

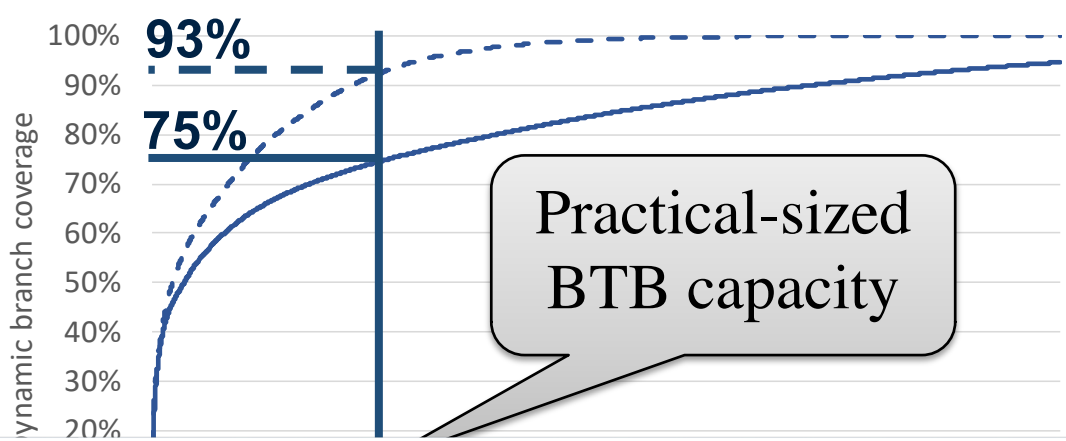
NTNU

THE UNIVERSITY of EDINBURGH

Global Control Flow Insight

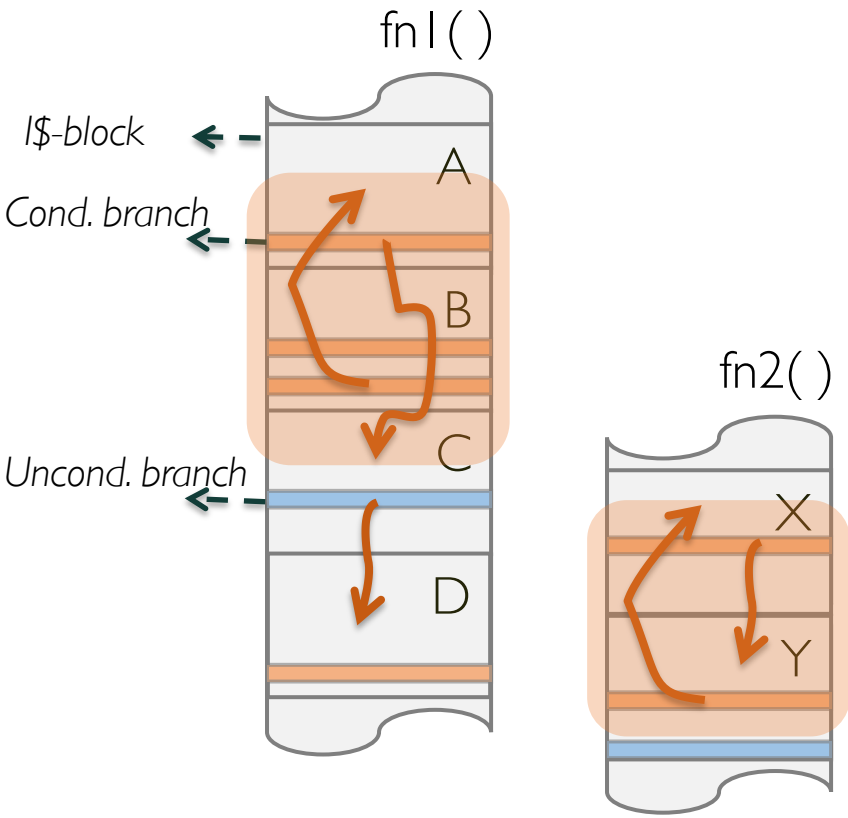


Small working set size



Global control flow fits in a practical-sized BTB

Local Control Flow Insight



High spatial locality within a code region

Local control flow affords a compact spatial representation

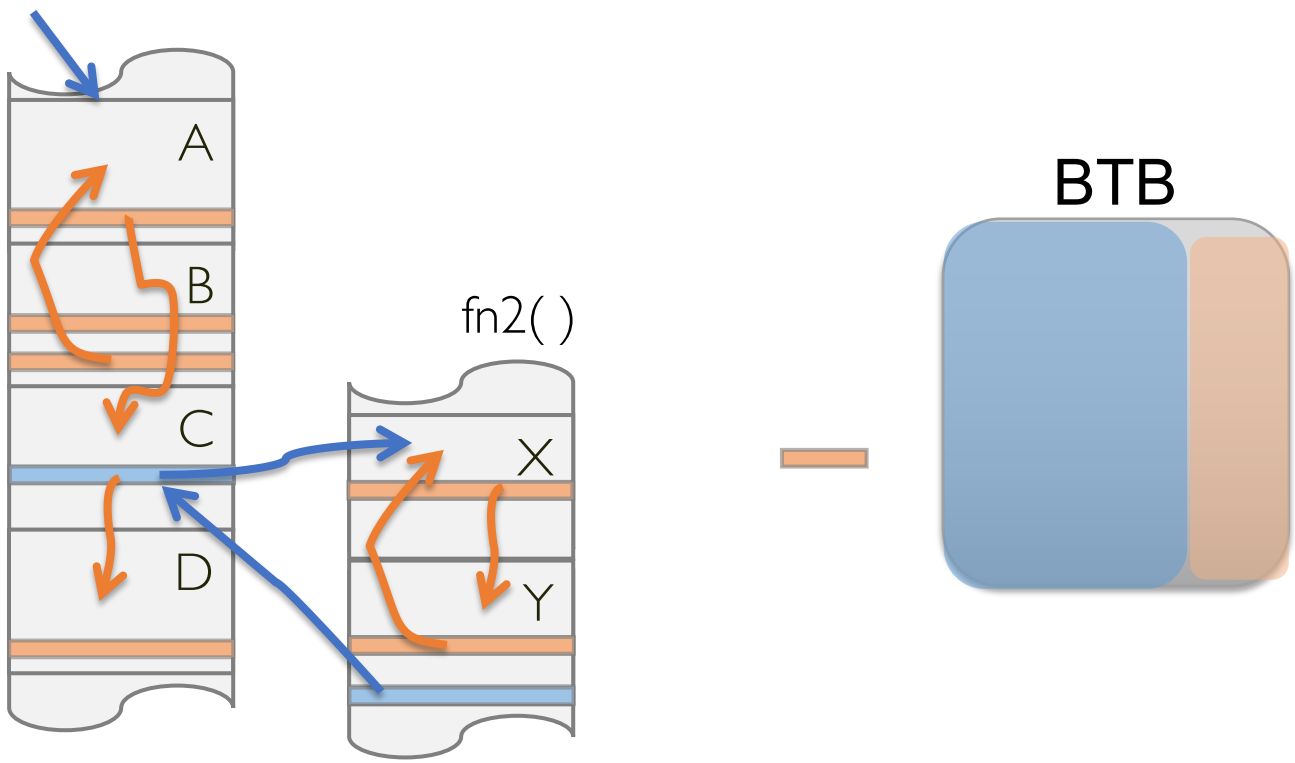
NTNU

THE UNIVERSITY of EDINBURGH

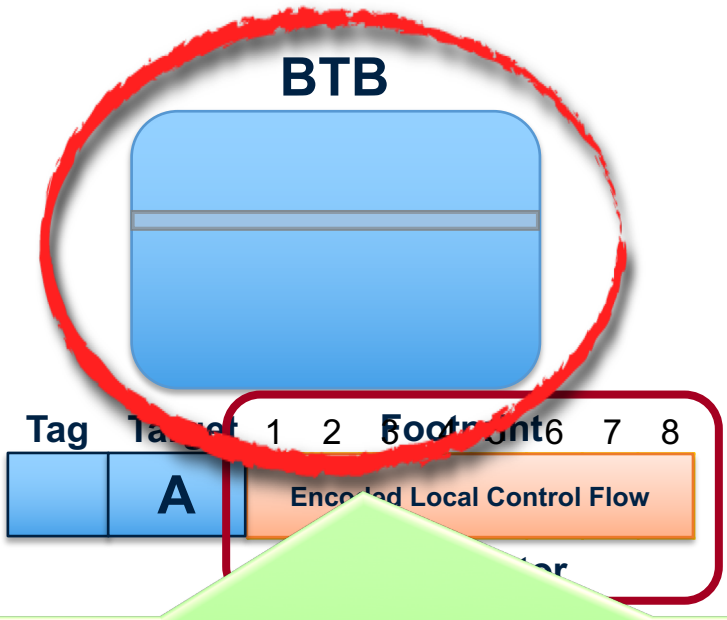
Mapping Control Flow to a BTB

Idea: Control flow footprint can be represented as

- Global control flow: unconditional branches
- Spatial encoding (footprint) of local control flow around each unconditional branch target

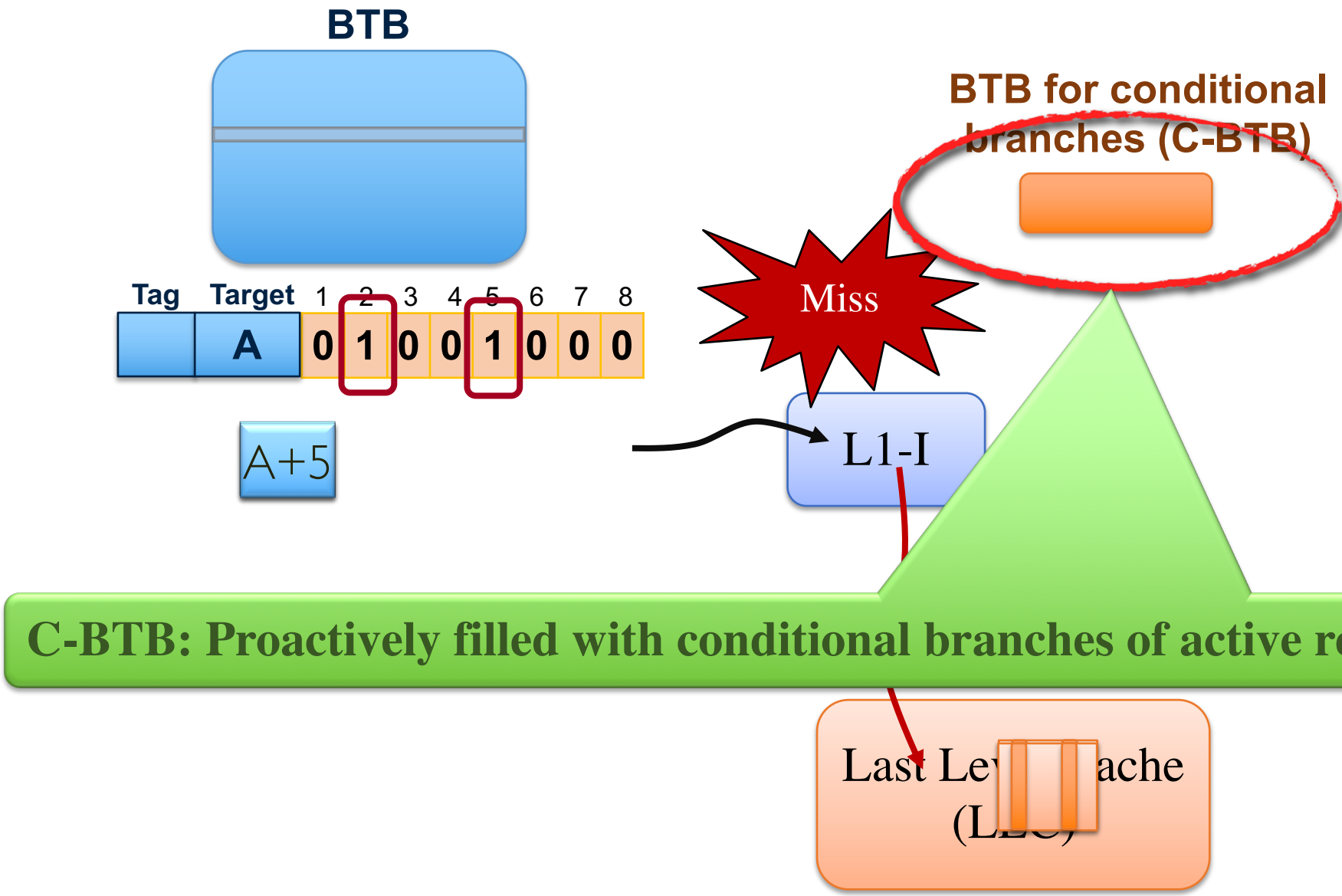


From Idea to Microarchitecture



Unconditional branches + target region footprints enable high-coverage L1-I prefetching

From Idea to Microarchitecture

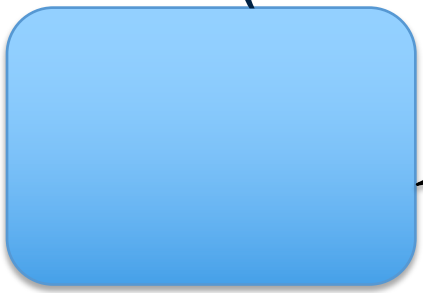


NTNU

THE UNIVERSITY of EDINBURGH

Shotgun: A Specialized BTB Organization for Control Flow Delivery

BTB for unconditional branches (U-BTB)



Bulk of BTB storage budget goes to U-BTB (unconditional branches + spatial footprints)

BTB for conditional branches (C-BTB)

Small size: nearly 20x less storage than U-BTB

Return Instruction Buffer (RIB)

Storage optimization for return instructions: minimal metadata → avoid placement in U-BTB

NTNU

THE UNIVERSITY of EDINBURGH

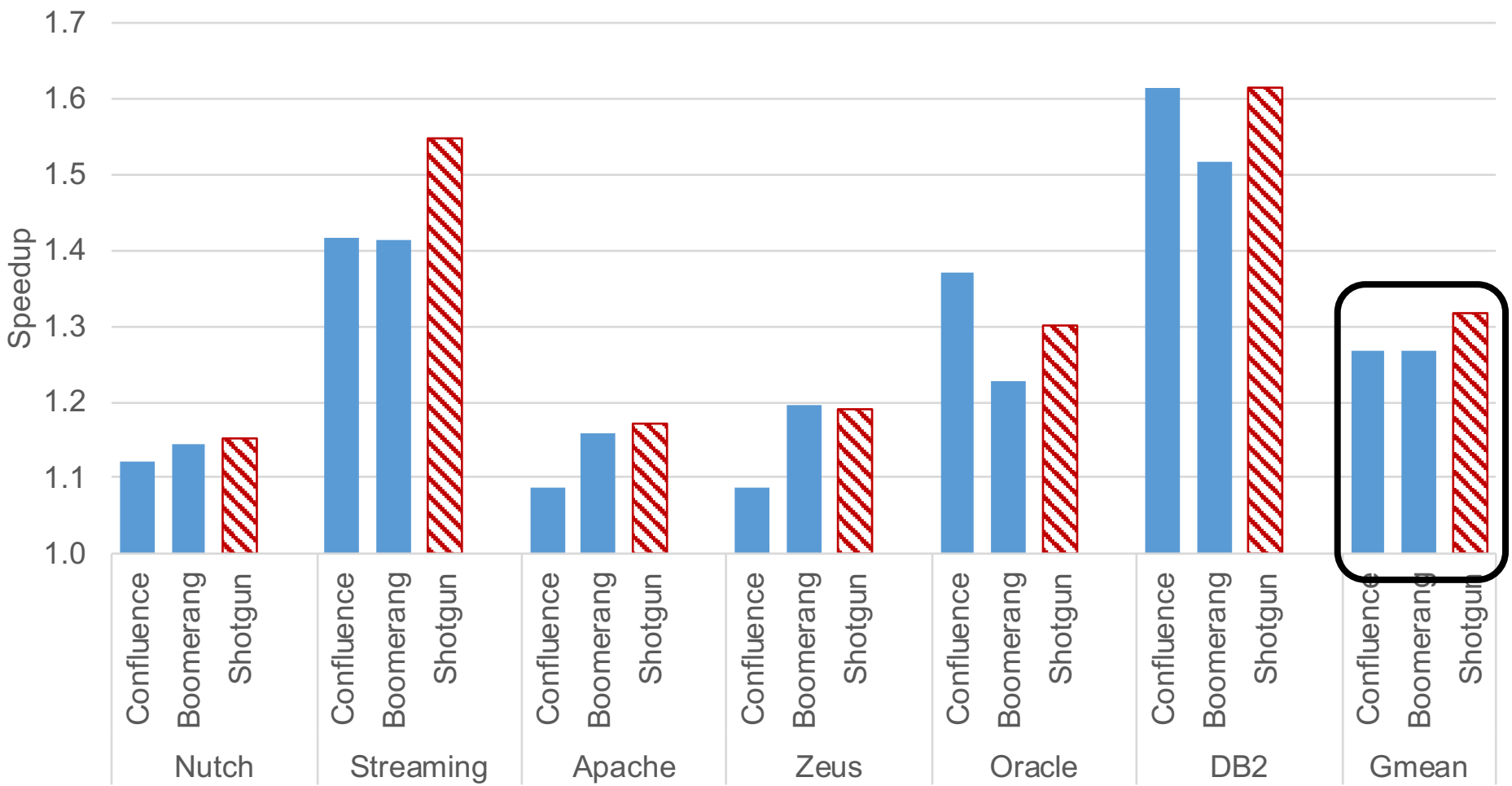
Evaluation Methodology

- 16-core CMP, 8MB LLC
- L1-I: 32KB
- BTB: 2K-entry
  - Equivalent storage budget for Shotgun
- Workloads: Enterprise and scale-out (databases, web search, media streaming, web serving)
- Evaluated prefetchers:
  - Temporal stream prefetcher: Confluence *[MICRO'15]*
  - BTB-directed prefetcher: Boomerang *[HPCA'17]*
  - Shotgun

NTNU

THE UNIVERSITY of EDINBURGH

Performance Comparison



NTNU

THE UNIVERSITY of EDINBURGH

Shotgun Summary

- Front-end bottleneck critical in servers
  - Prior work: trades off between storage and performance
- Control flow behavior-guided BTB design
  - Global control flow fits in a practical-sized BTB
- **High performance core front-end without costly metadata**
  - Uses the BTB to map the instruction working set using control flow behavior insights
    - Enables highly effective front-end prefetching
  - Erases the performance gap between metadata-rich and metadata-free front-end prefetchers