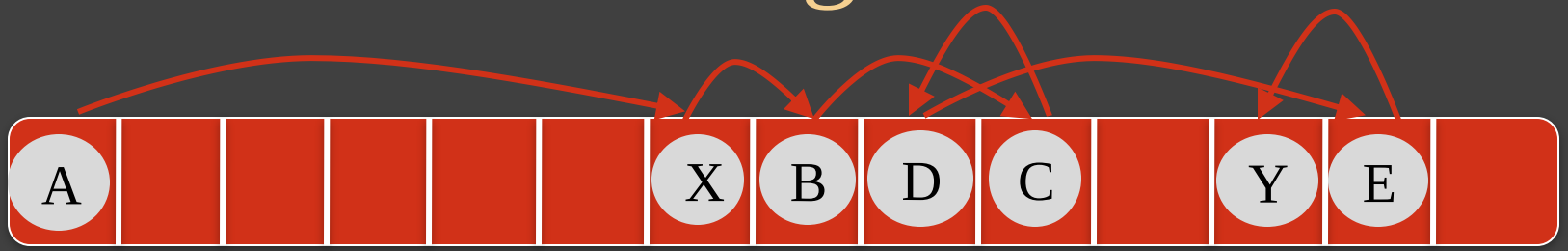
TEXAS
The University of Texas at Austin

arm

Temporal Prefetching Without the Off-Chip Metadata

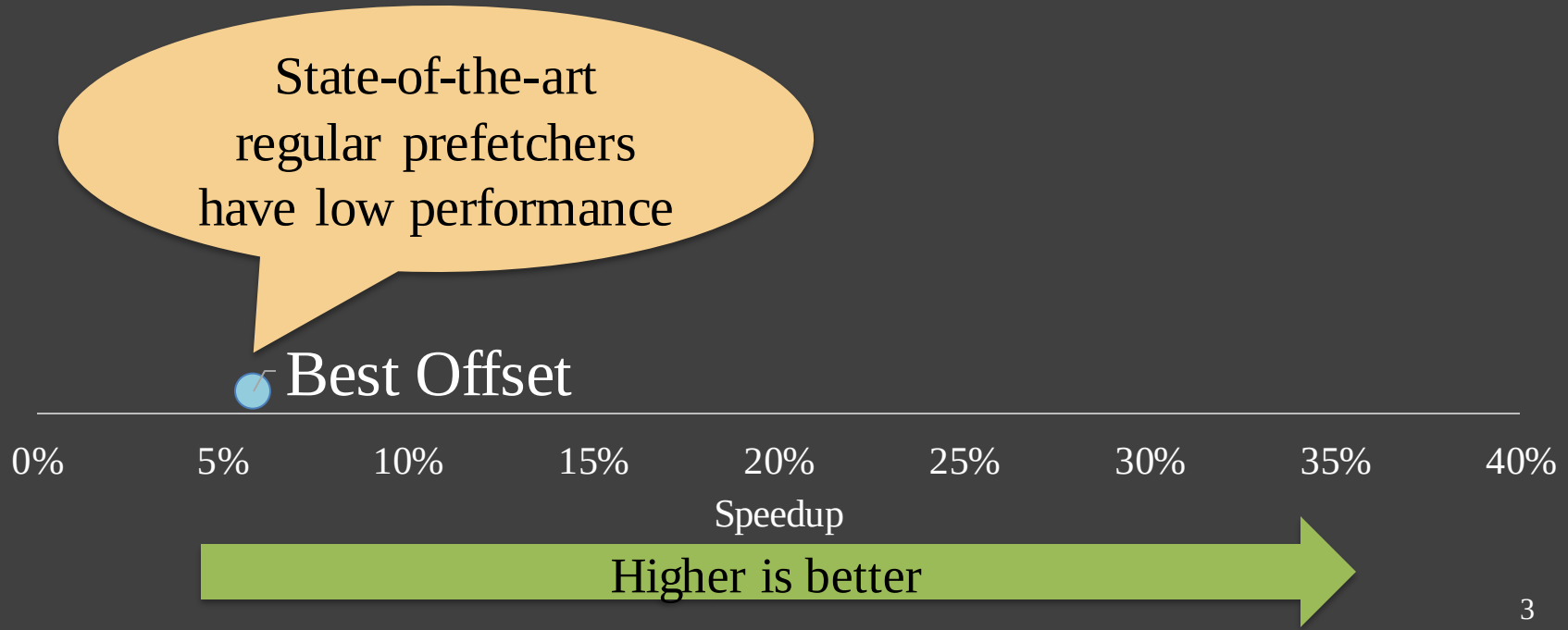
Hao Wu (UT Austin), Krishnendra Nathella (Arm), Joseph Pusdesris (Arm), Dam Sunwoo (Arm), Akanksha Jain *(UT Austin)*, Calvin Lin *(UT Austin)*

The Problem: Irregular Accesses

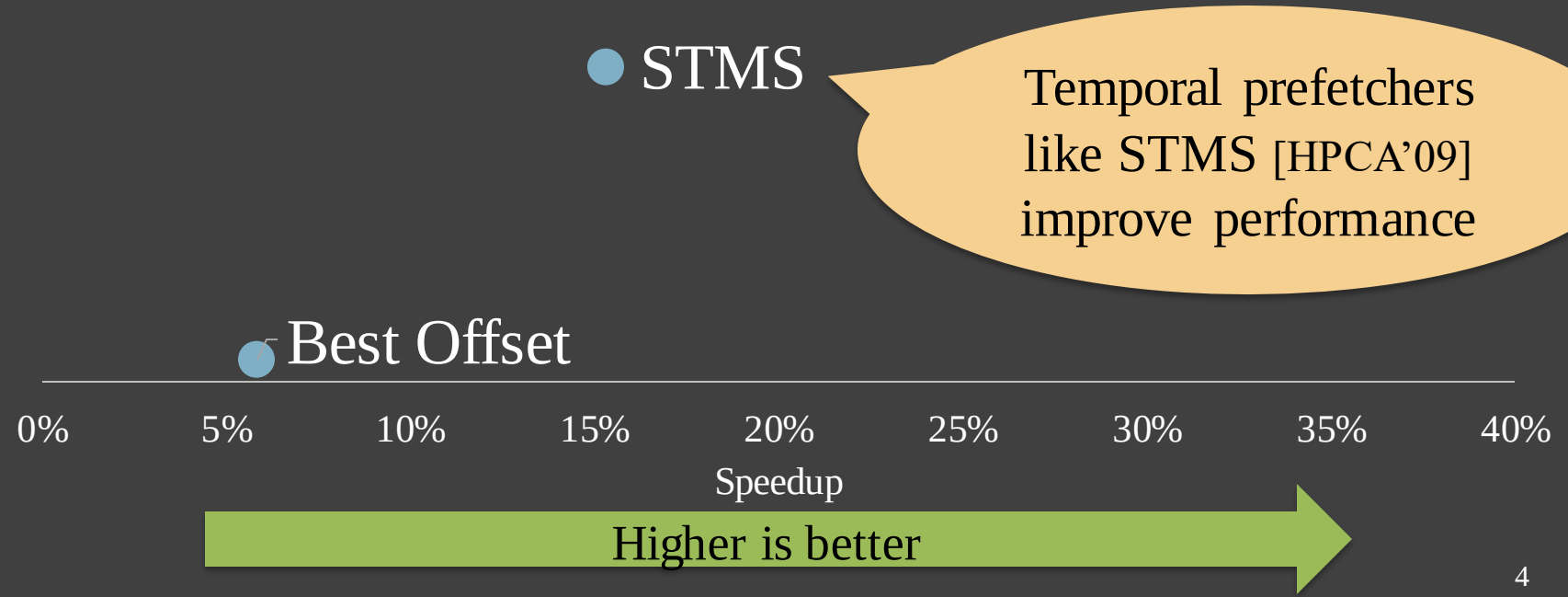


- Common in many programs
- Hard to prefetch

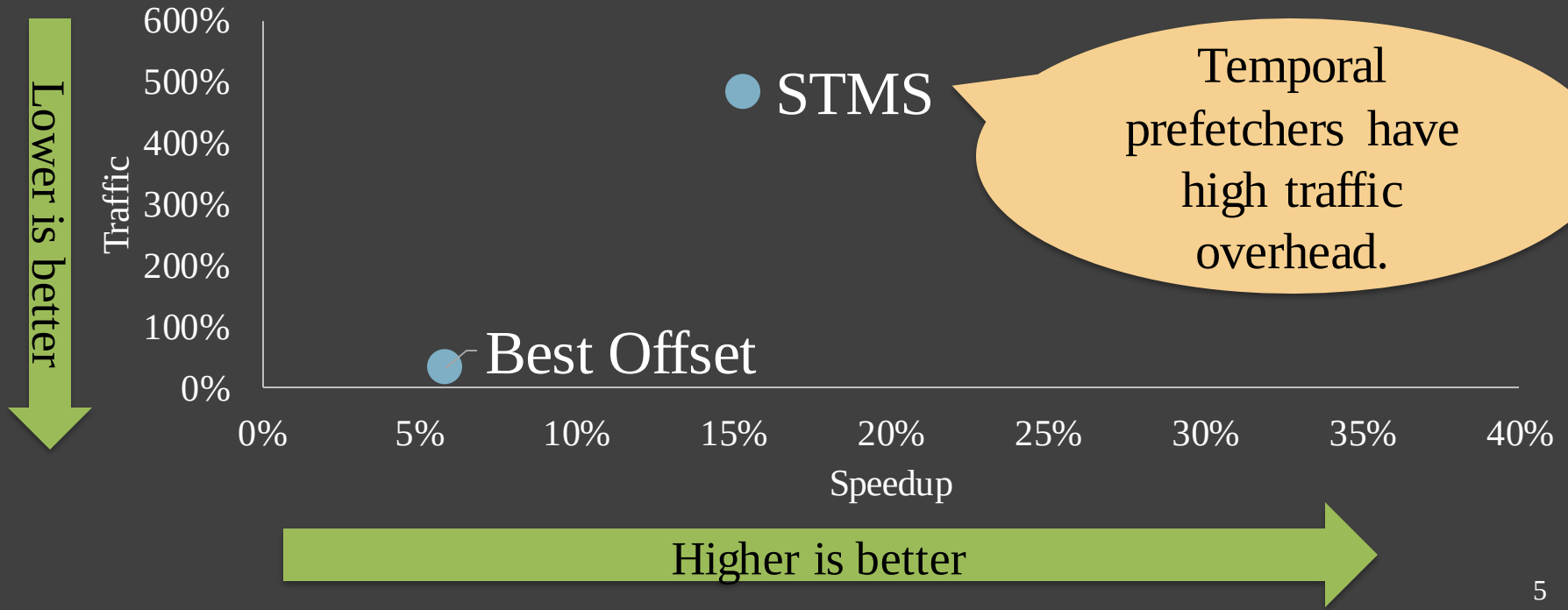
Design Space For Irregular Prefetching



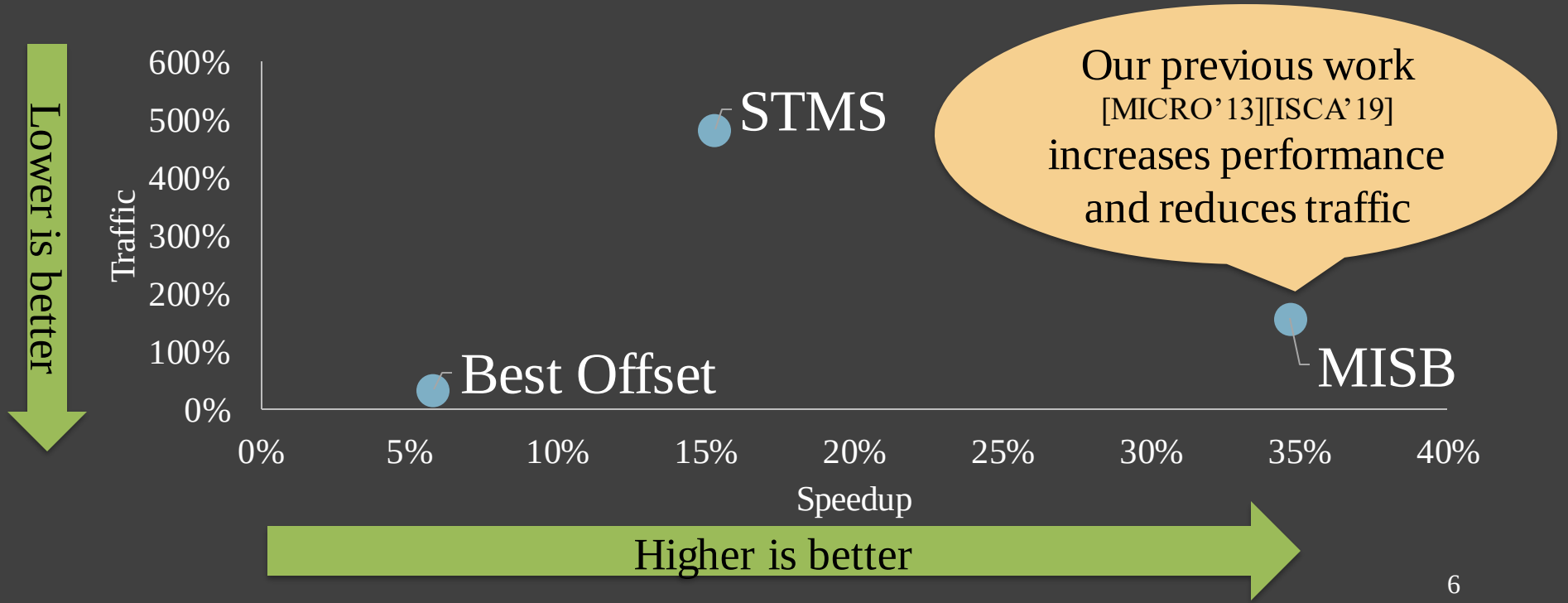
Design Space For Irregular Prefetching



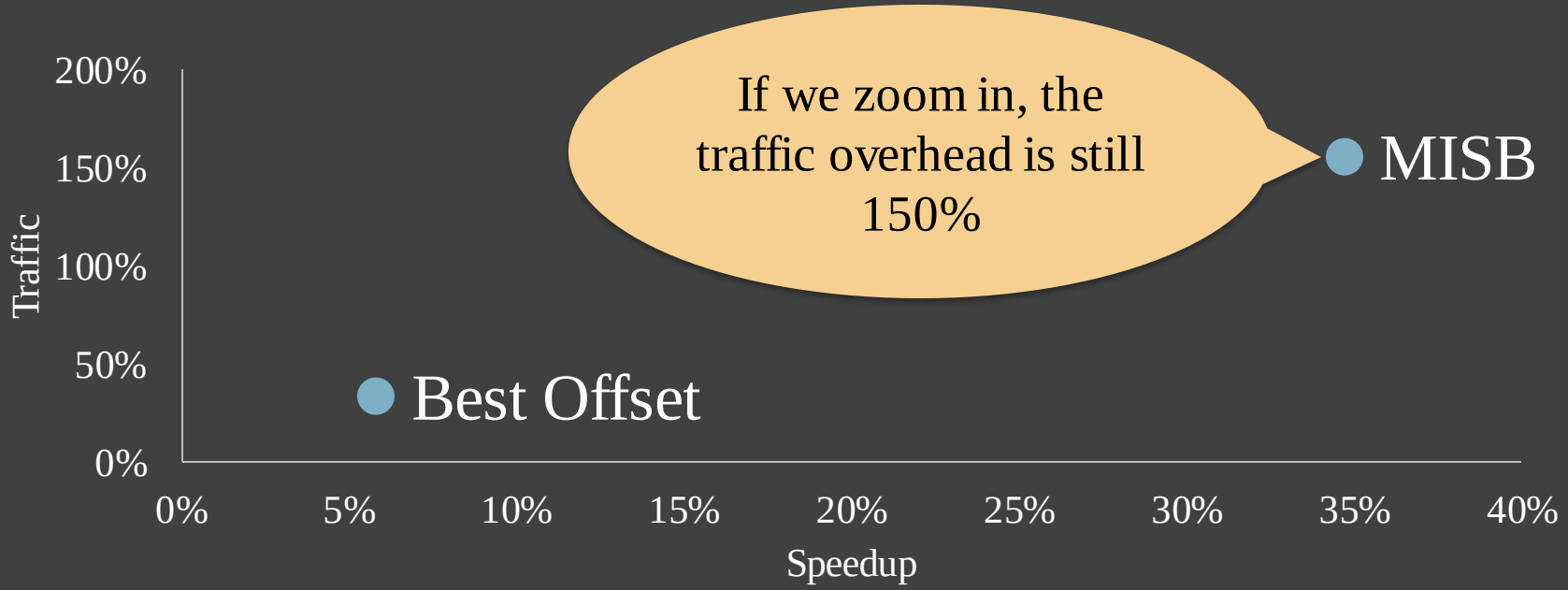
Design Space For Irregular Prefetching



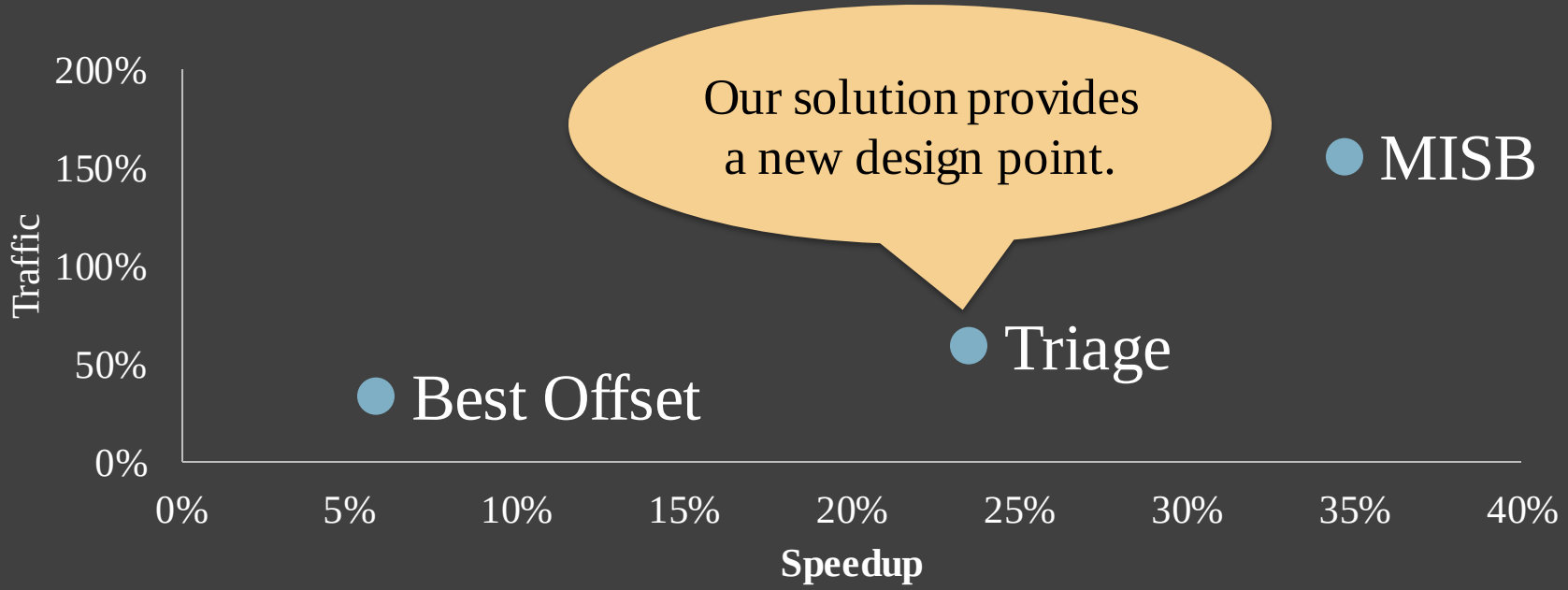
Design Space For Irregular Prefetching



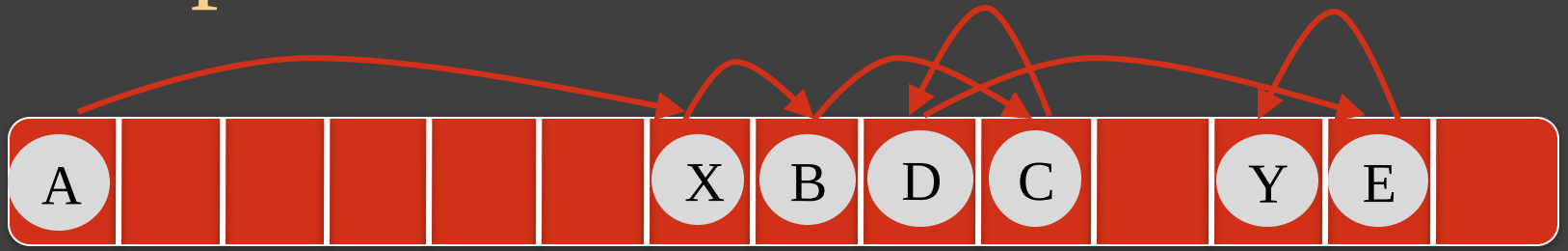
Design Space For Irregular Prefetching



Design Space For Irregular Prefetching



Temporal Prefetchers



- Memorize correlations

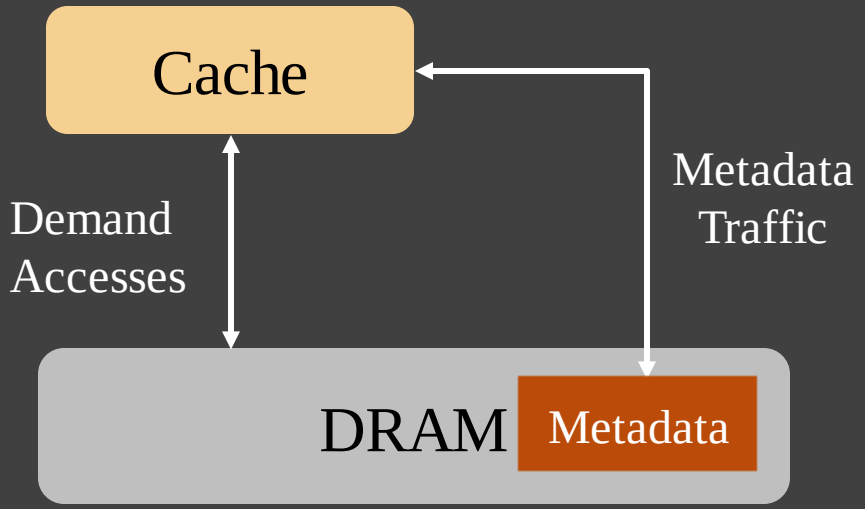
A	X	B	C	D	E	Y	…

- Replay memorized accesses

A							…

10~20 MB
of metadata

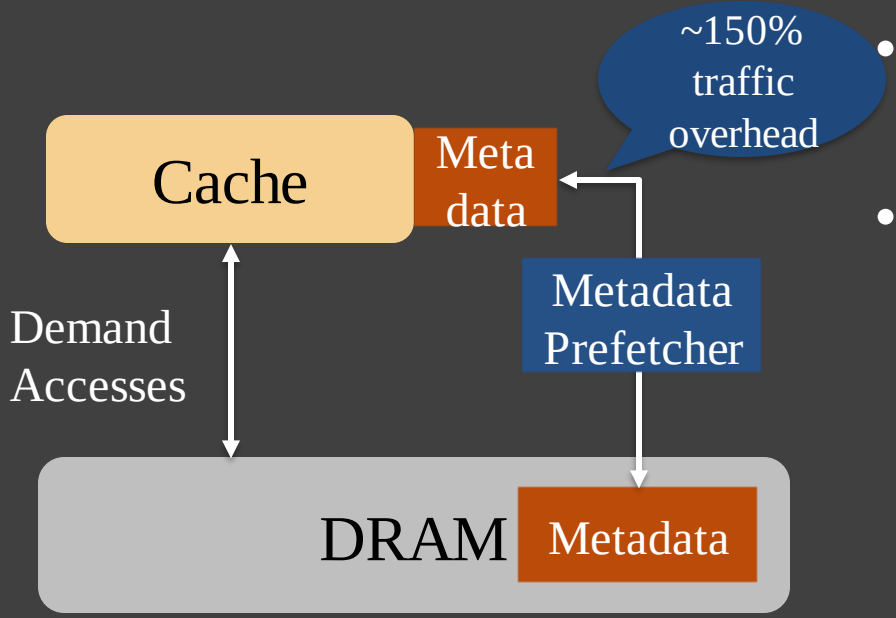
Temporal Prefetchers



- High metadata overhead (10~20 MB)
- Too large to fit on-chip
- Metadata stored off-chip

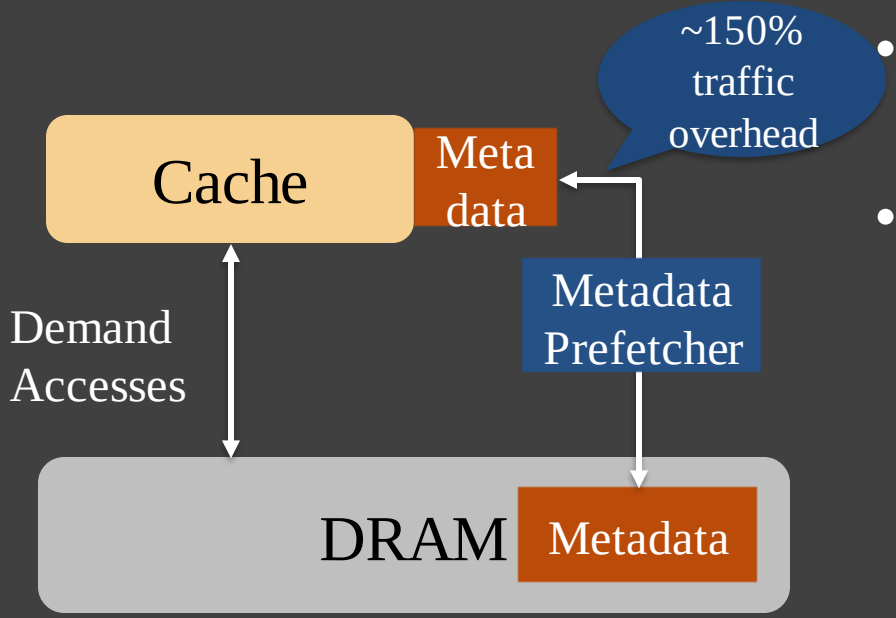
  ~480% traffic overhead

Managed ISB (MISB) [ISCA'19]



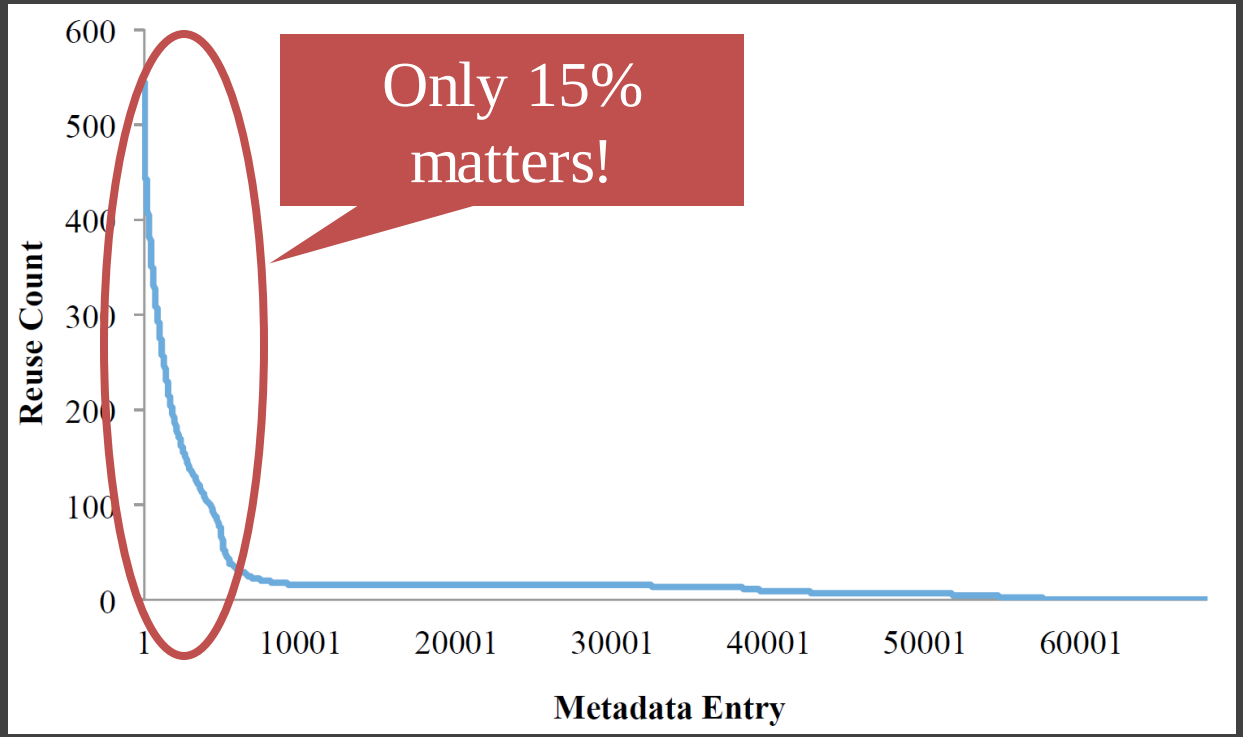
- Caching metadata on-chip
  - Prefetch metadata
- ~150% traffic overhead
  - Reduces performance in bandwidth-constrained environments
  - Complicates hardware design

Our Solution: Triage



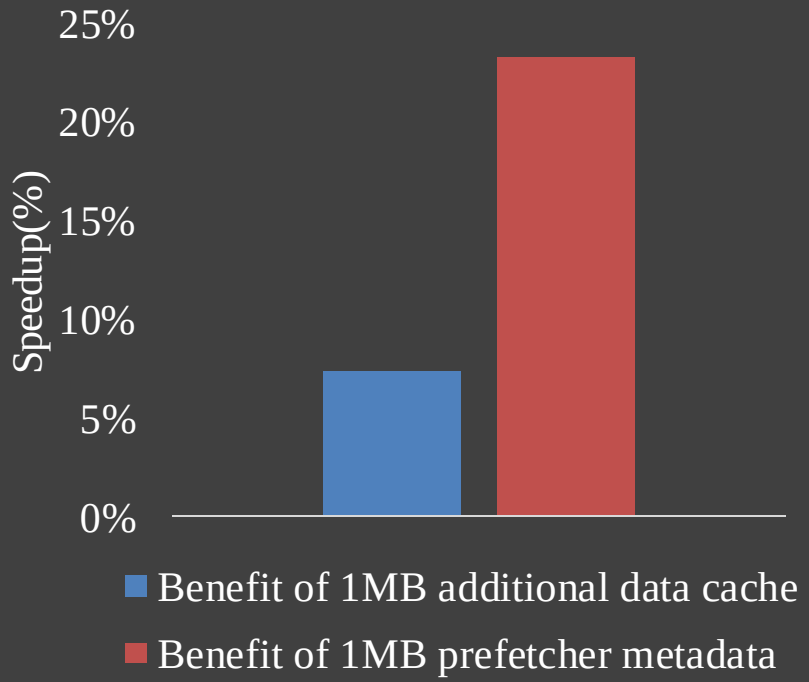
- Completely removes off-chip metadata
- How can we fit metadata on-chip?

Insight 1: Not All Metadata Is Useful



Insight 2: Metadata more useful than data

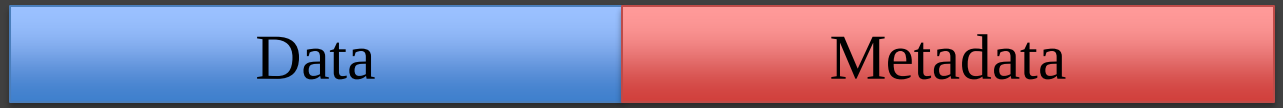
- For irregular benchmarks, benefit of prefetching > benefit of last level cache (LLC)
- We use part of LLC to store metadata



Metadata stored in last level cache

Original LLC

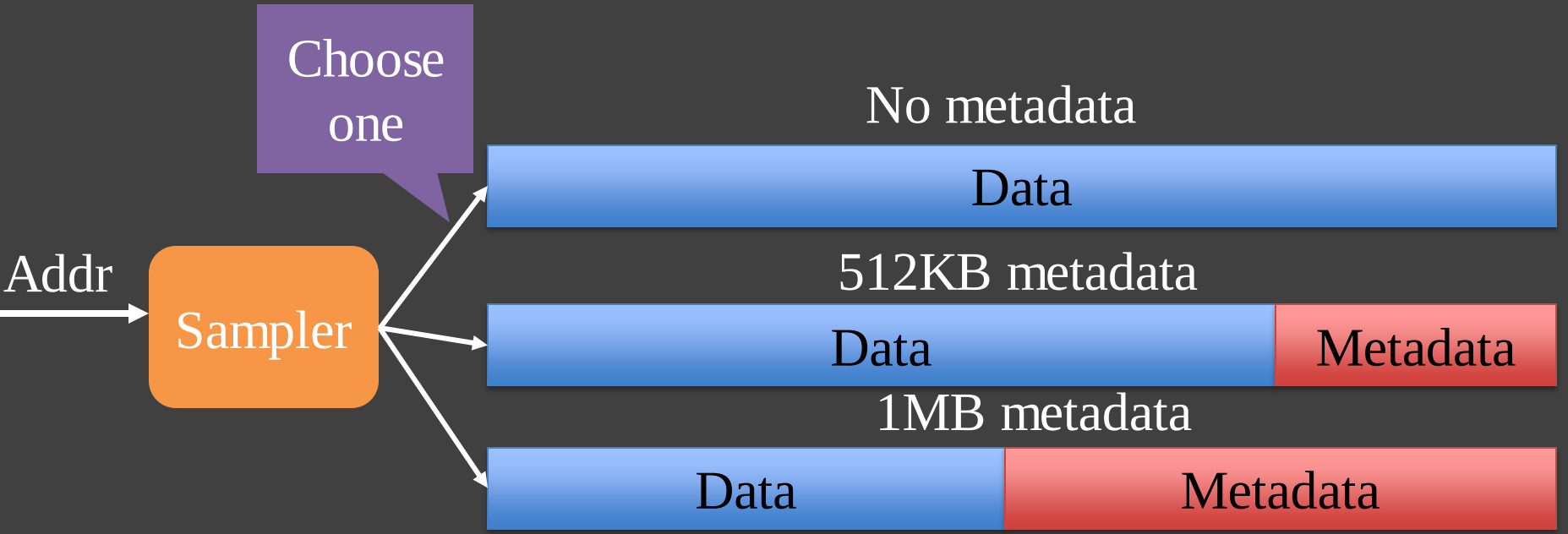
Part of LLC is used to store metadata



How to find out useful metadata?

- This is a cache replacement problem!
- We use state-of-art cache replacement – Hawkeye [ISCA'16] to figure it out

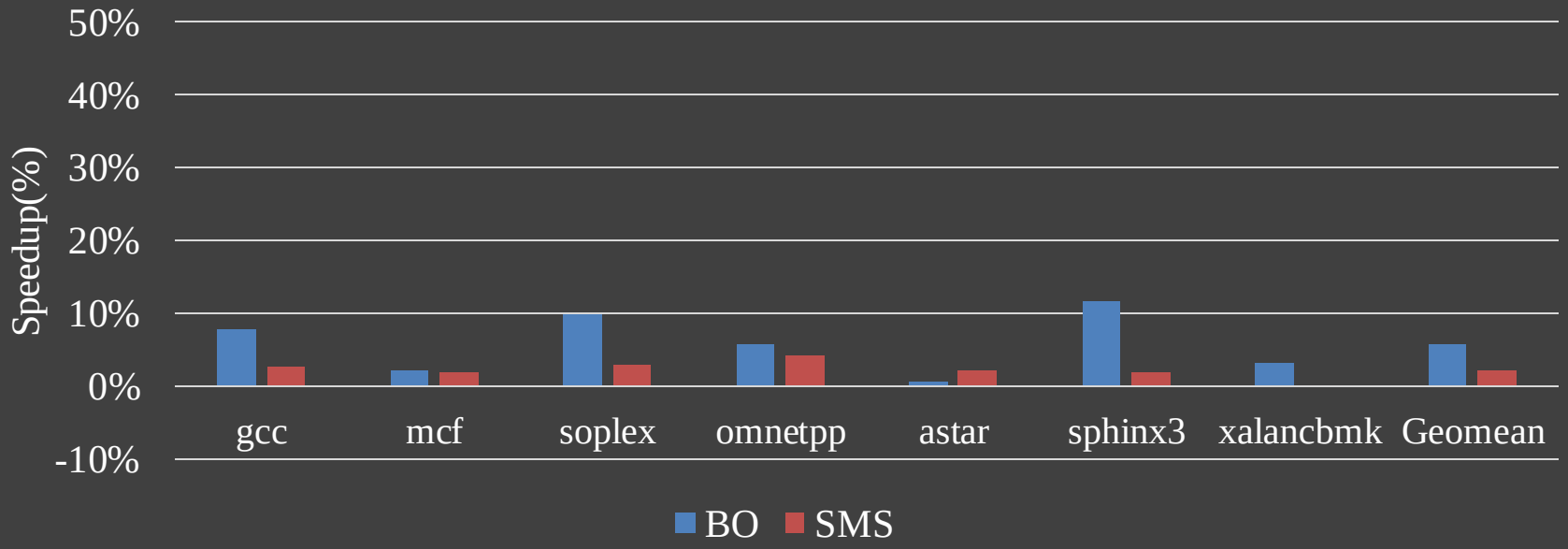
How large should the metadata store be?



Evaluation Methodology

- Industrial Simulator
  - ARMv8 AArch64
  - OoO Core
  - Bandwidth: 32GB/s
- Multicore – ChampSim
  - Similar trends as the industrial simulator
- Benchmark
  - SPEC2006
    Irregular subset
  - CloudSuite
- Prefetchers
  - On chip metadata
    - BO, SMS
  - Off chip metadata
    - STMS, Domino, MISB

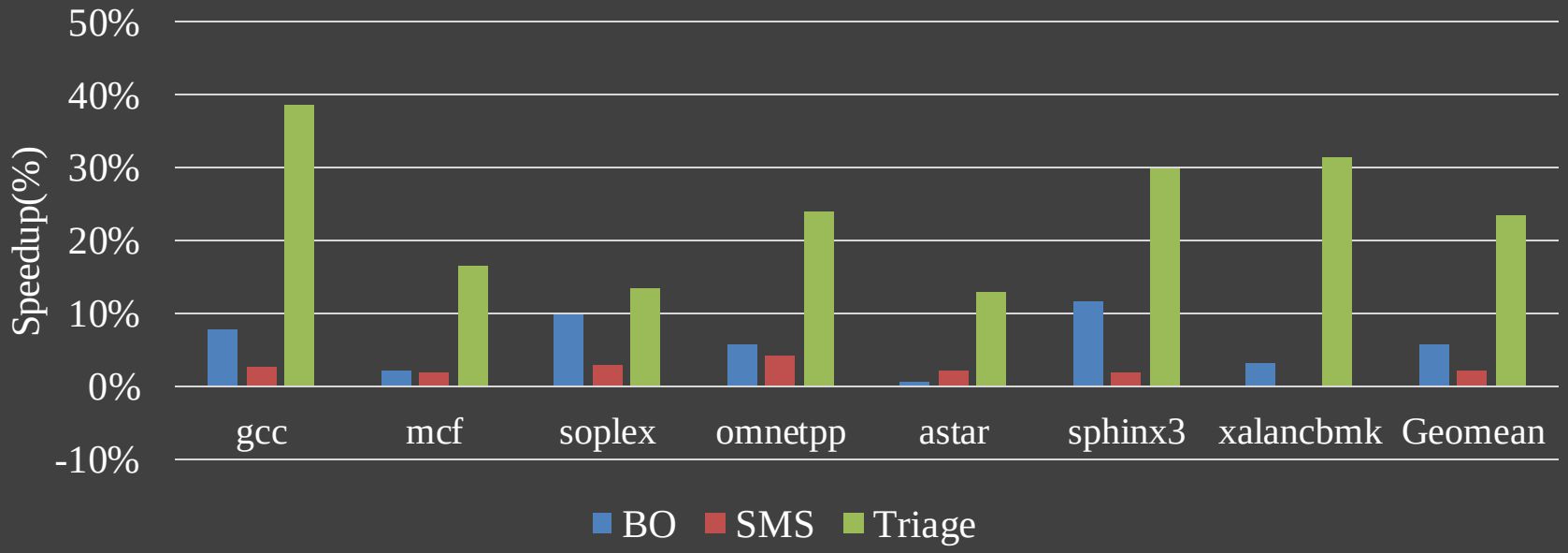
Performance – Single Core



TEXAS
The University of Texas at Austin

arm

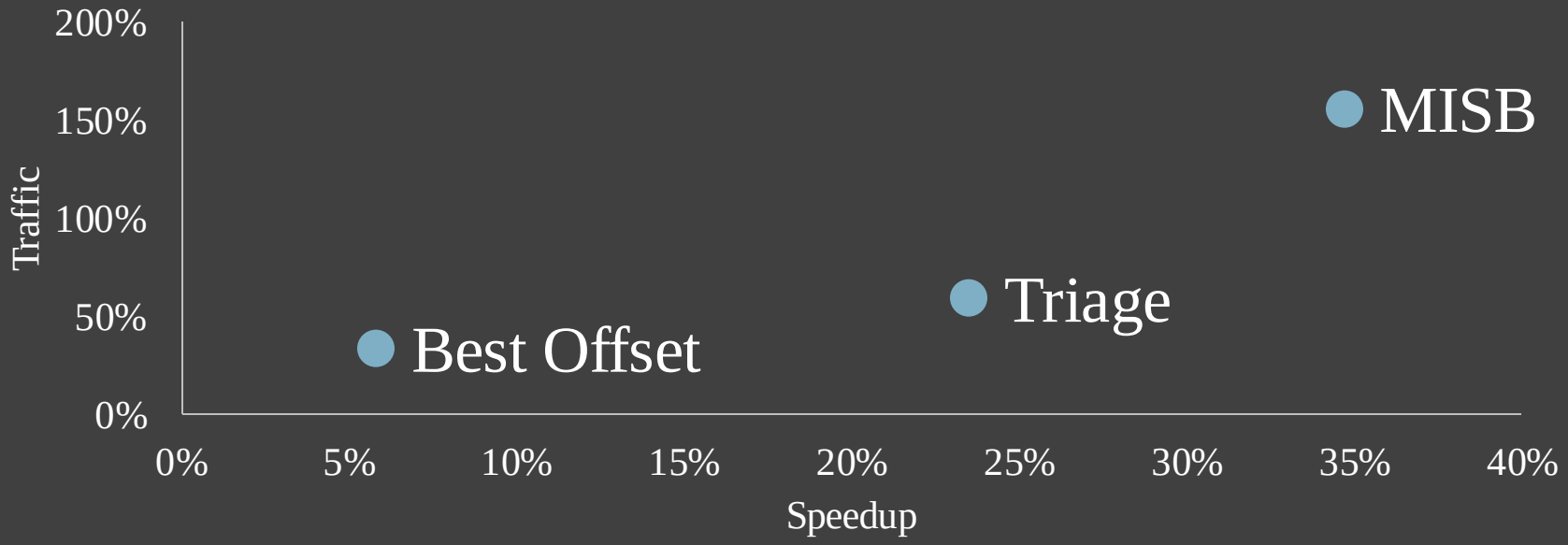
Performance – Single Core



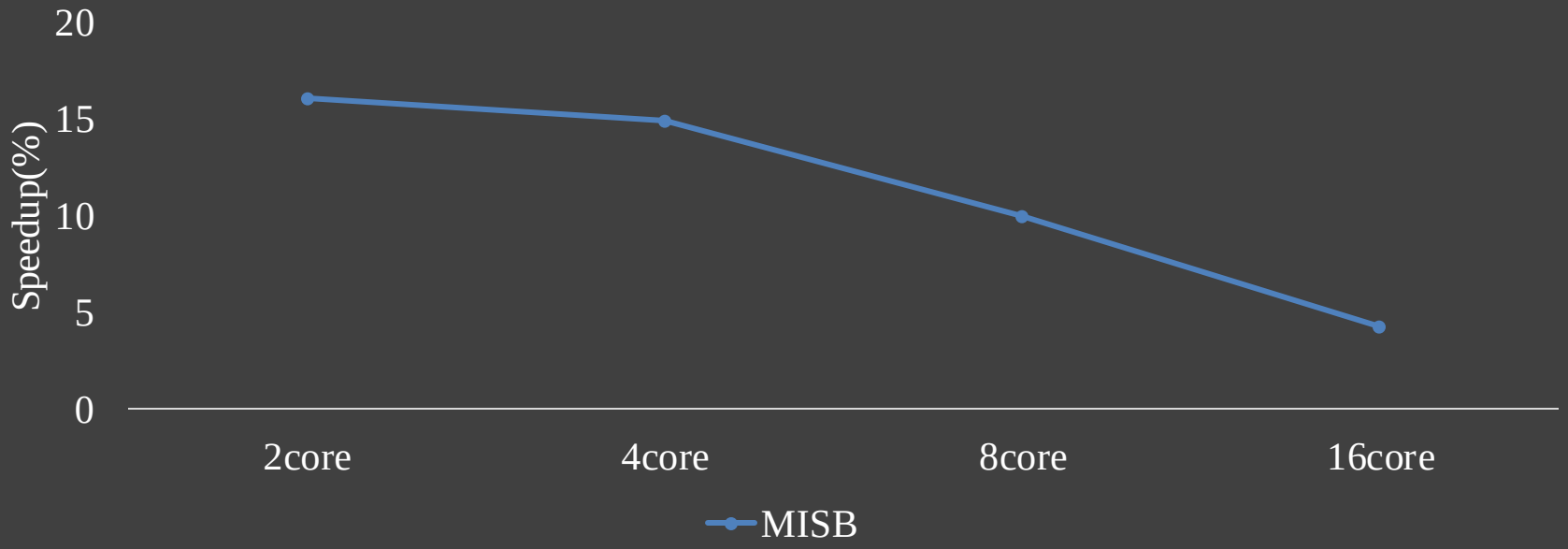
TEXAS
The University of Texas at Austin

arm

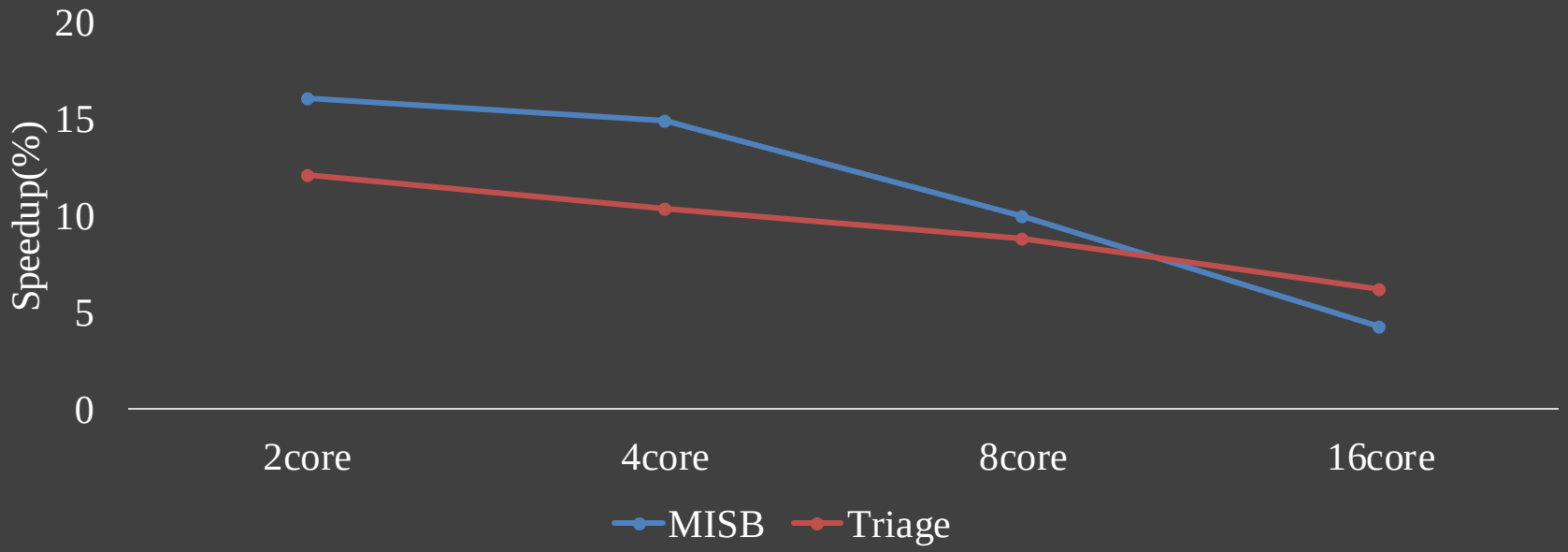
Performance – Single Core



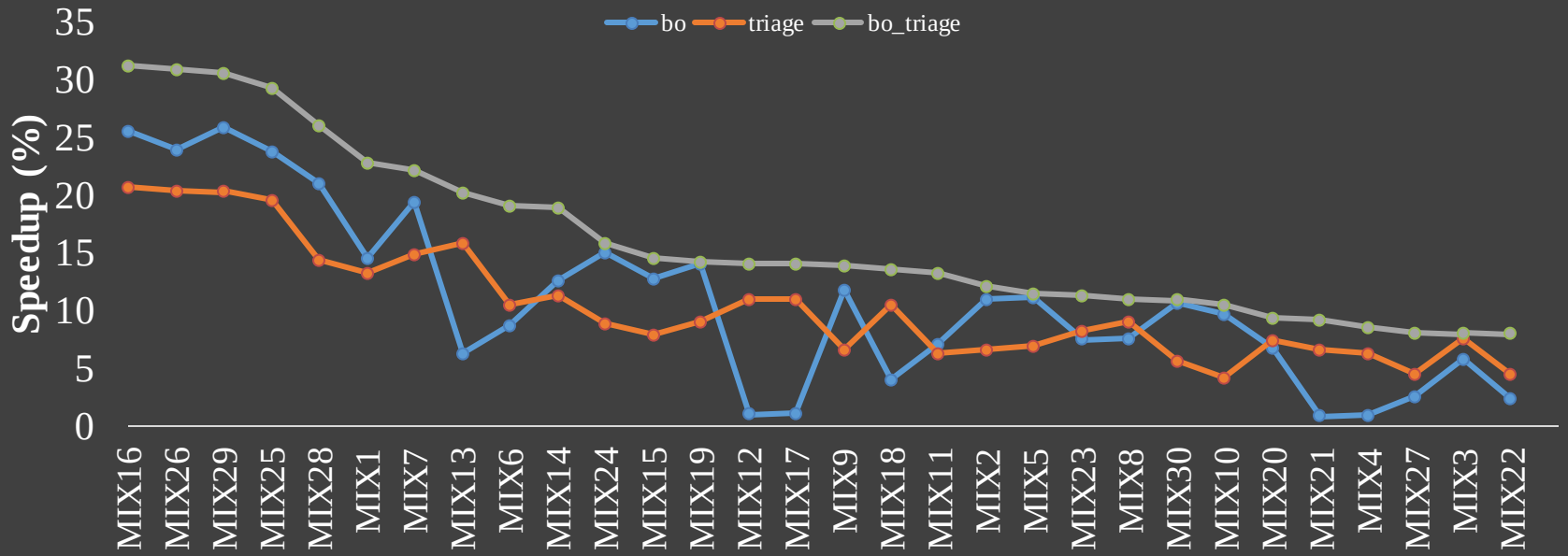
Performance – Multi Core



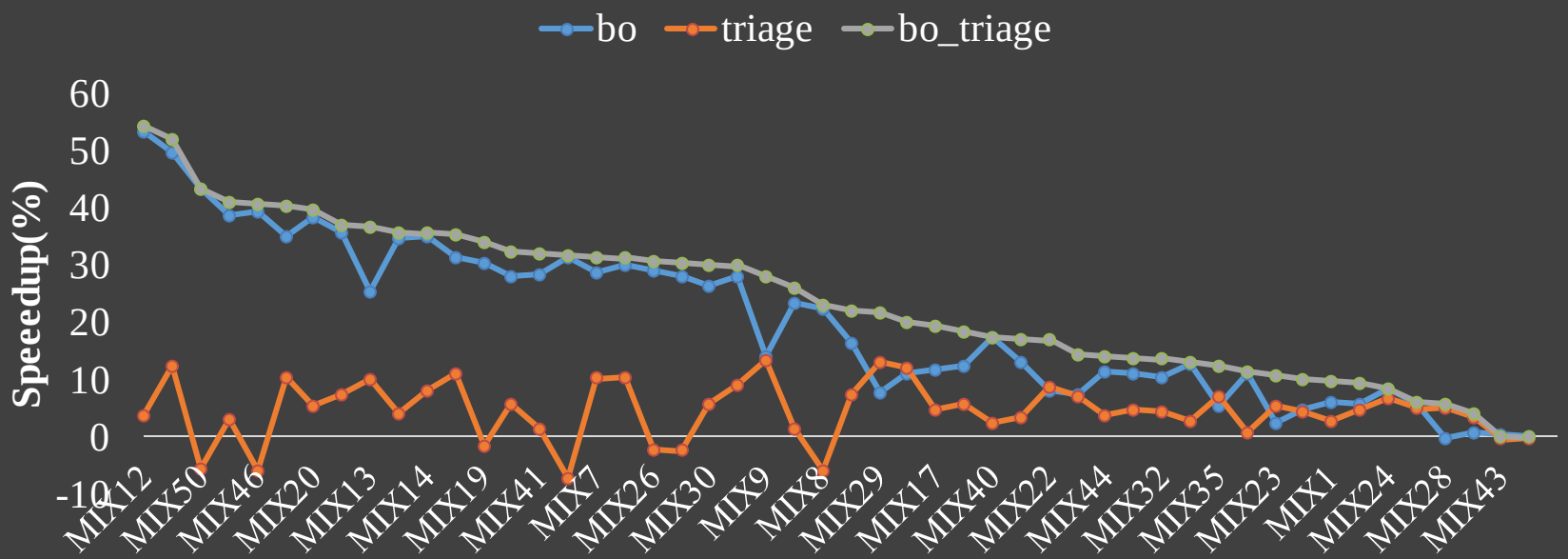
Performance – Multi Core



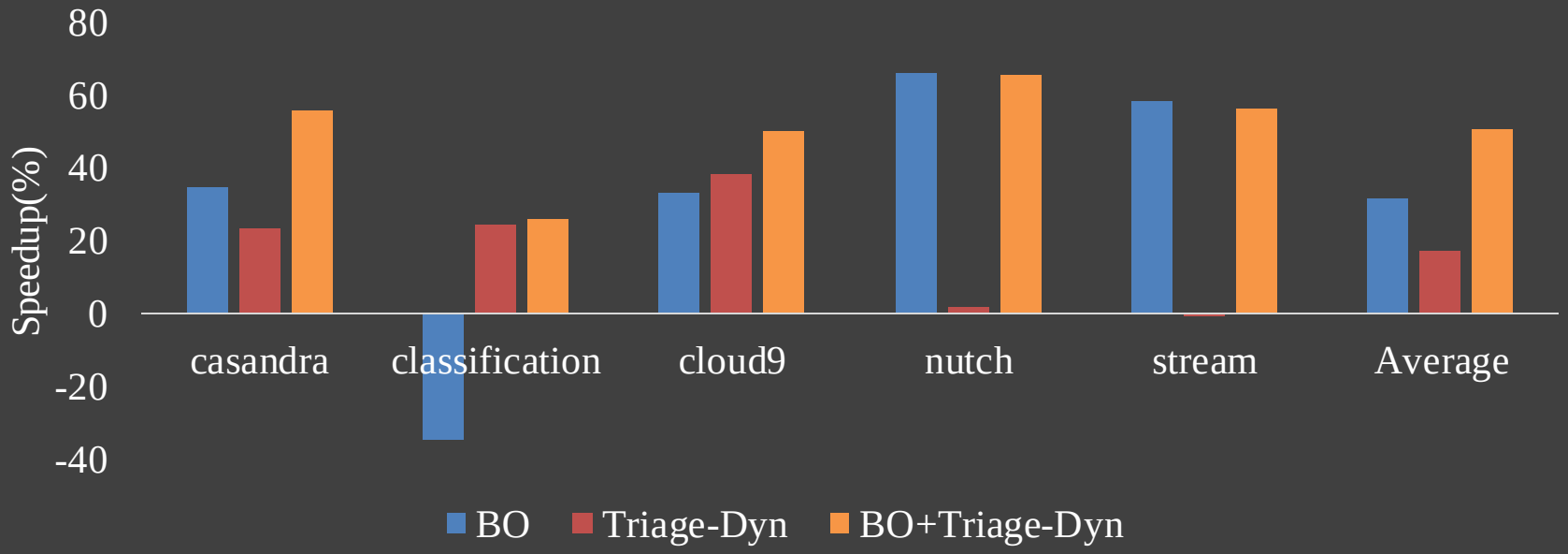
Performance – 4-Core SPEC Irregular



Performance – 4-Core All SPEC



Performance - Cloudsuite



Conclusion

- Triage provides a new design point for temporal prefetchers
- Triage has good performance
  - 23.5% speedup vs. 5.8% for BO
  - 59.3% traffic vs. 156.4% for MISB
  - Better performance than MISB in 16-core systems
- For more information please read our MICRO paper

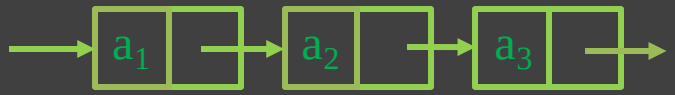
Thank you!

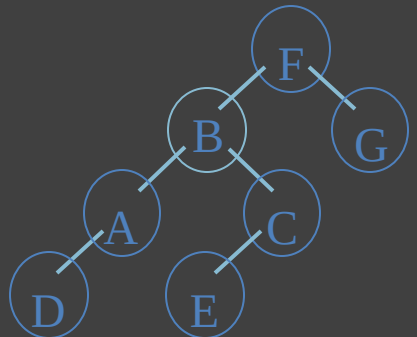
Regular Prefetching

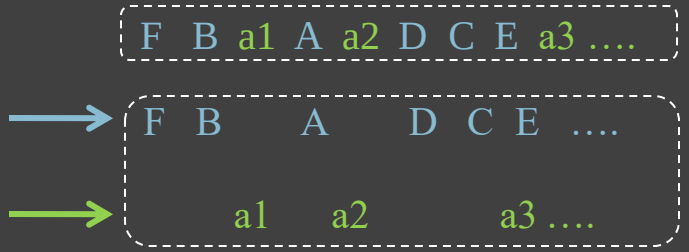
- Some programs access memory sequentially
  - e.g. MPEG player
- Regular prefetchers are effective and widely used
  - e.g. Best offset prefetcher

Global vs. PC-Localization

```
while ( ! end ) {
  read tree->next;
  if (condition)
    read linked_list->next;
}
```

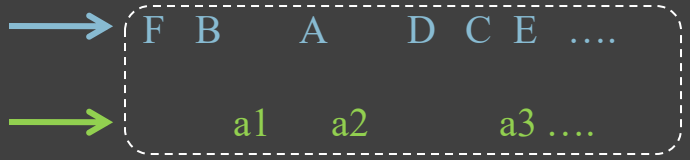






- PC-localization: Segregate the global stream by the load instruction's PC
- PC-localized streams are more predictable!

Metadata Representation

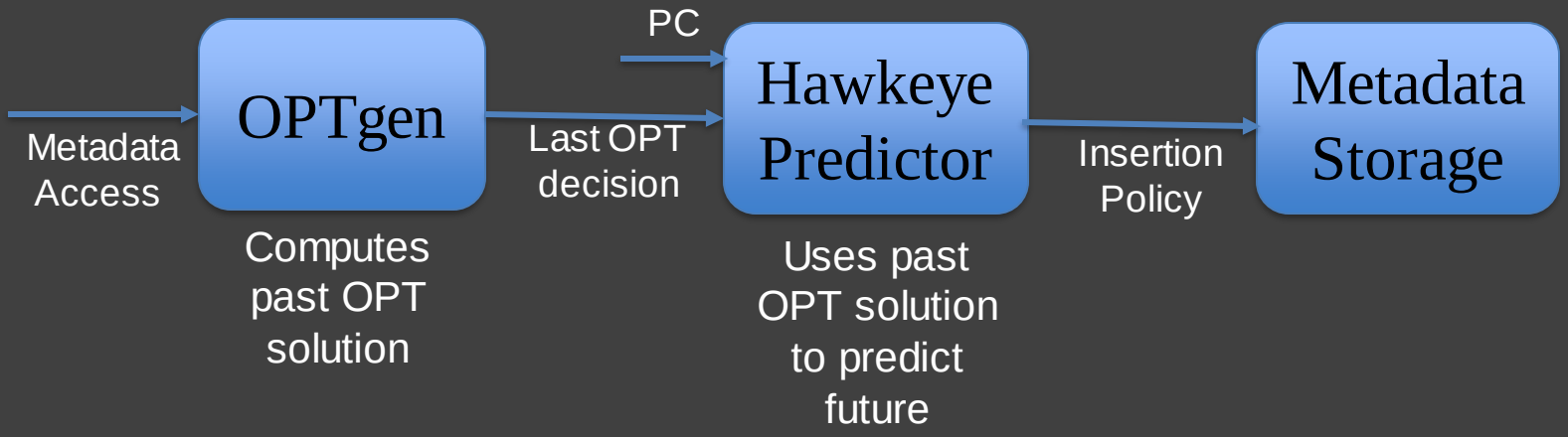


Address	Neighbor
F	B
B	A
A	D
…	…
a1	a2
a2	a3

Metadata Replacement

- Hawkeye[Jain et al, ISCA'18]
  - OPTgen learns from Belady's OPT policy
  - PC-based predictor predicts whether an access is hit or miss using OPTgen

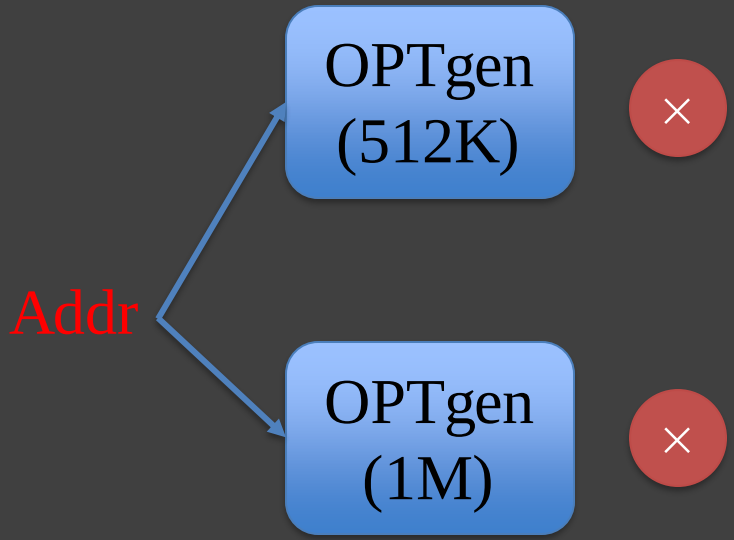
Metadata Replacement



Metadata Storage

- We use last level cache (LLC) to store metadata
- For irregular workloads, the benefit from prefetching is higher than performance reduction from reducing LLC size

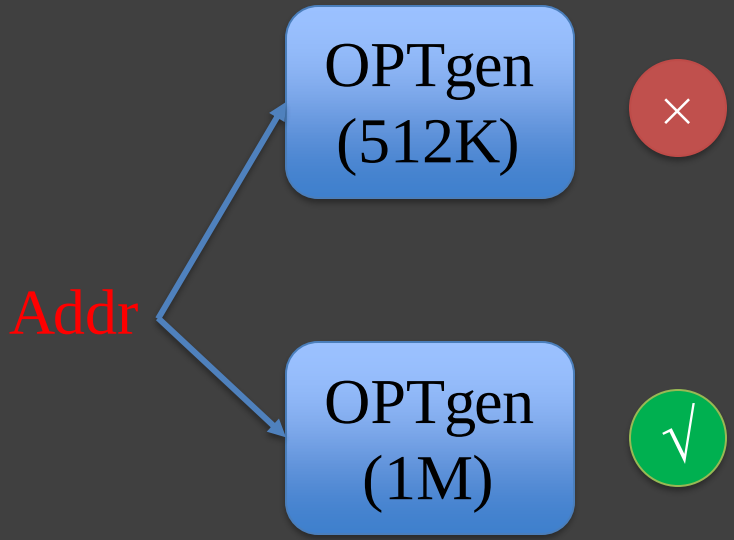
Dynamic Allocation

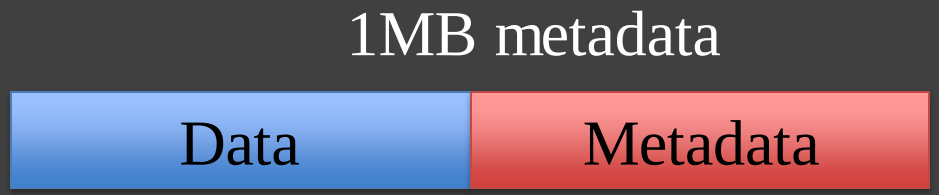


No metadata

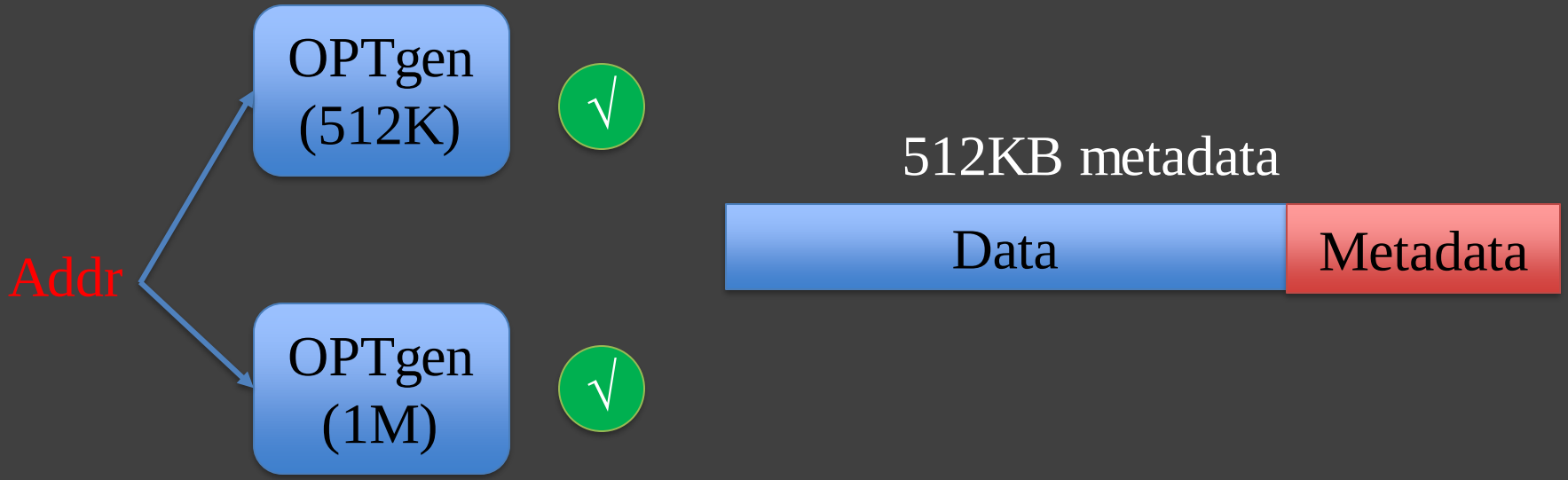
Data

Dynamic Allocation

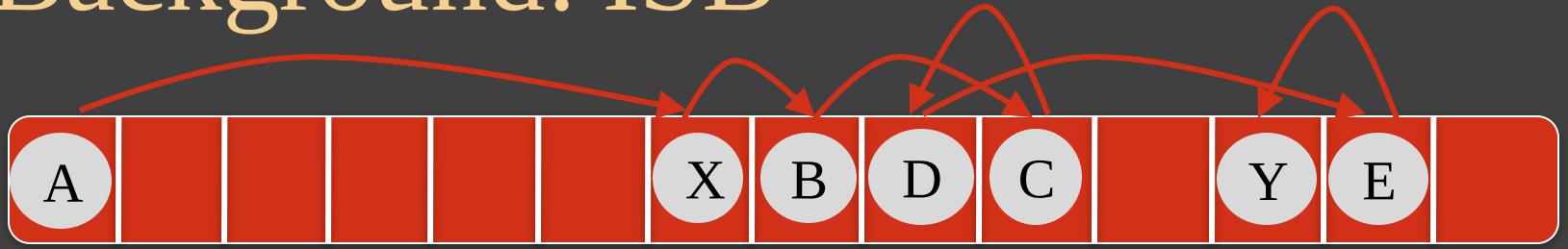




Dynamic Allocation



Background: ISB

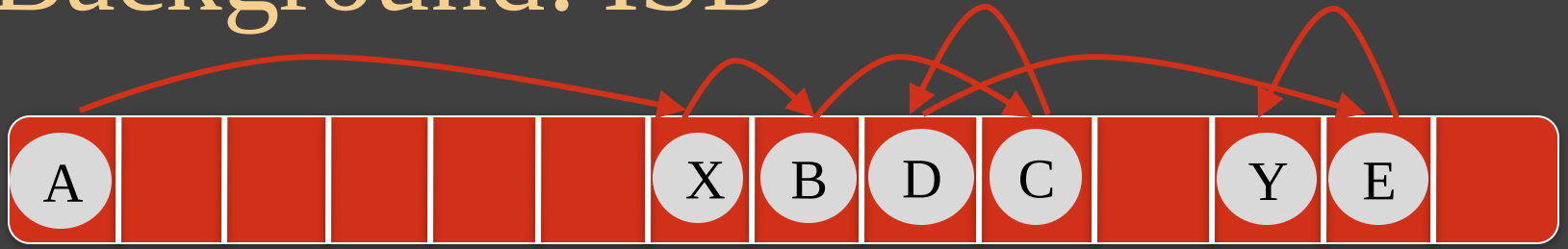


- Assign a structural address for each access in a stream
- Convert irregular access streams to sequential streams

Metadata

Physical	Structural
A	71
X	72
B	73
…	…

Background: ISB

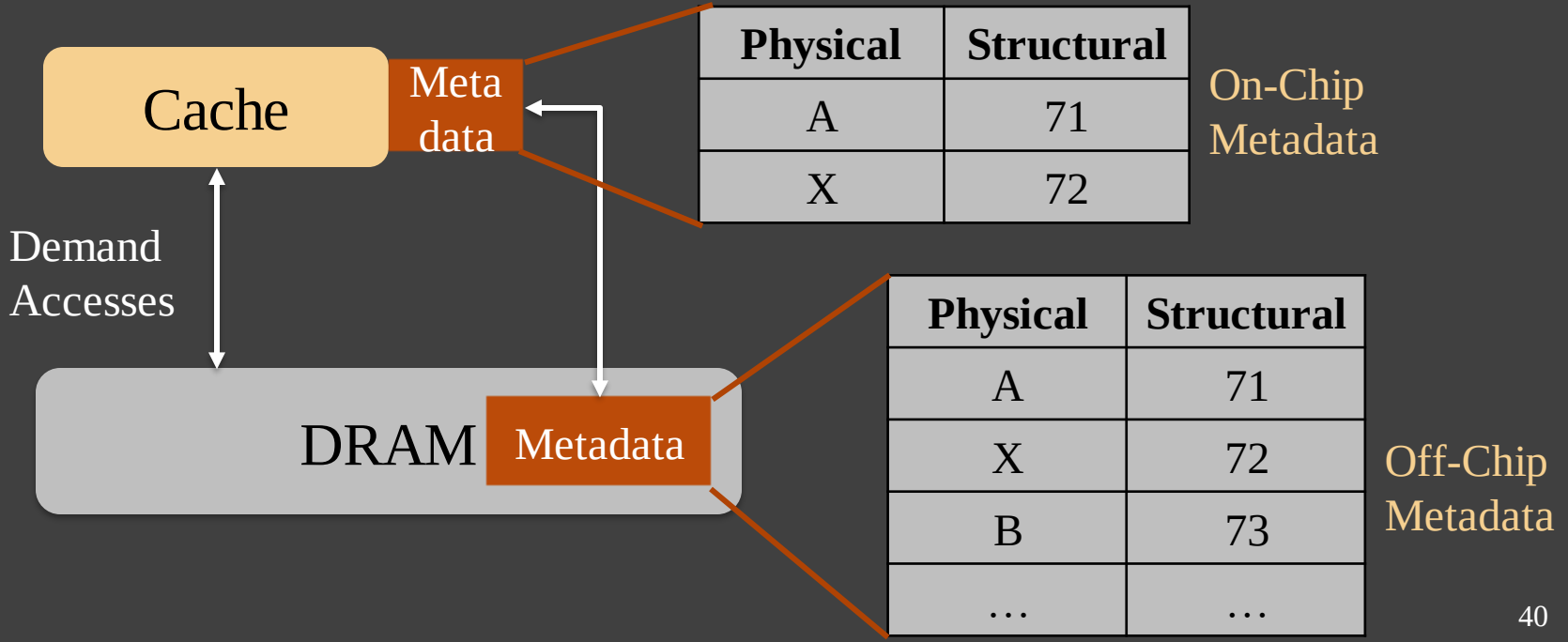


- Prefetch the next address in structural address space

Metadata

Physical	Structural
A	71
X	72
B	73
…	…

Background: ISB

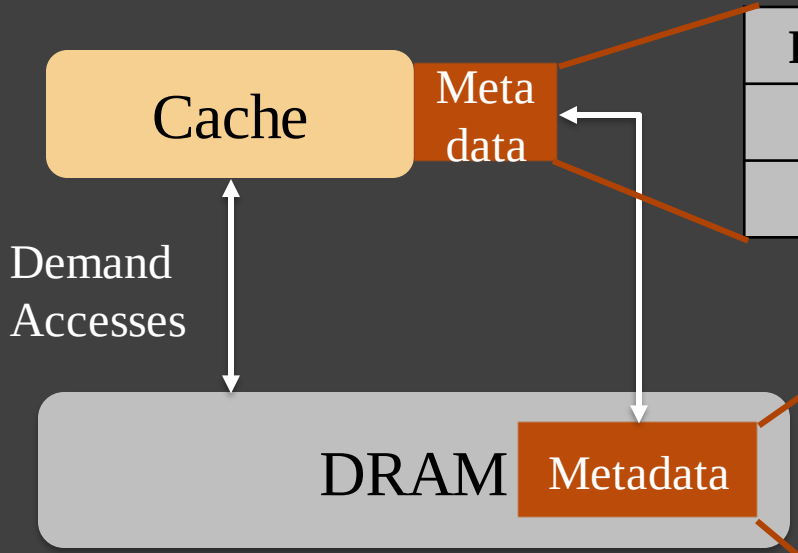


Background: ISB

ISB's metadata cache is synchronized with the TLB

TLB

VA_page	PA_page
V1	A
V2	X



Physical	Structural
A	71
X	72

On-Chip Metadata

Physical	Structural
A	71
X	72
B	73
…	…

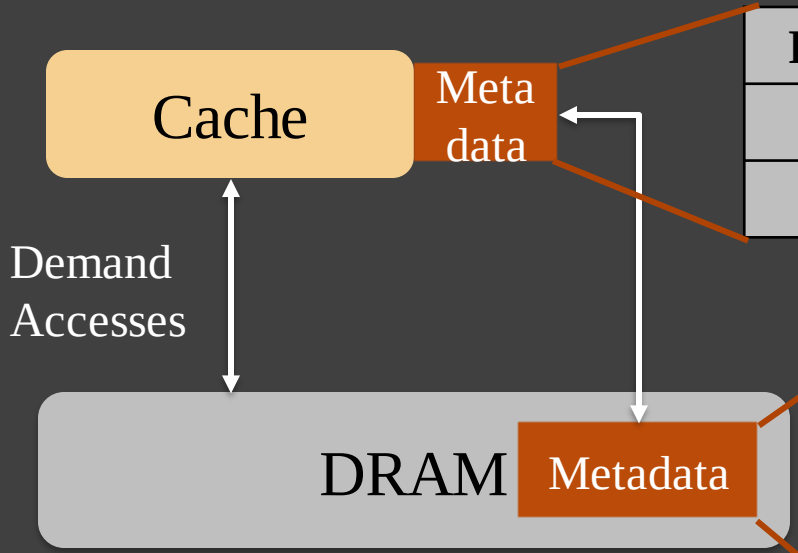
Off-Chip Metadata

Background: ISB

ISB's metadata cache is synchronized with the TLB

TLB

VA_page	PA_page
V1	A
V2	X



Physical	Structural
A	71
X	72

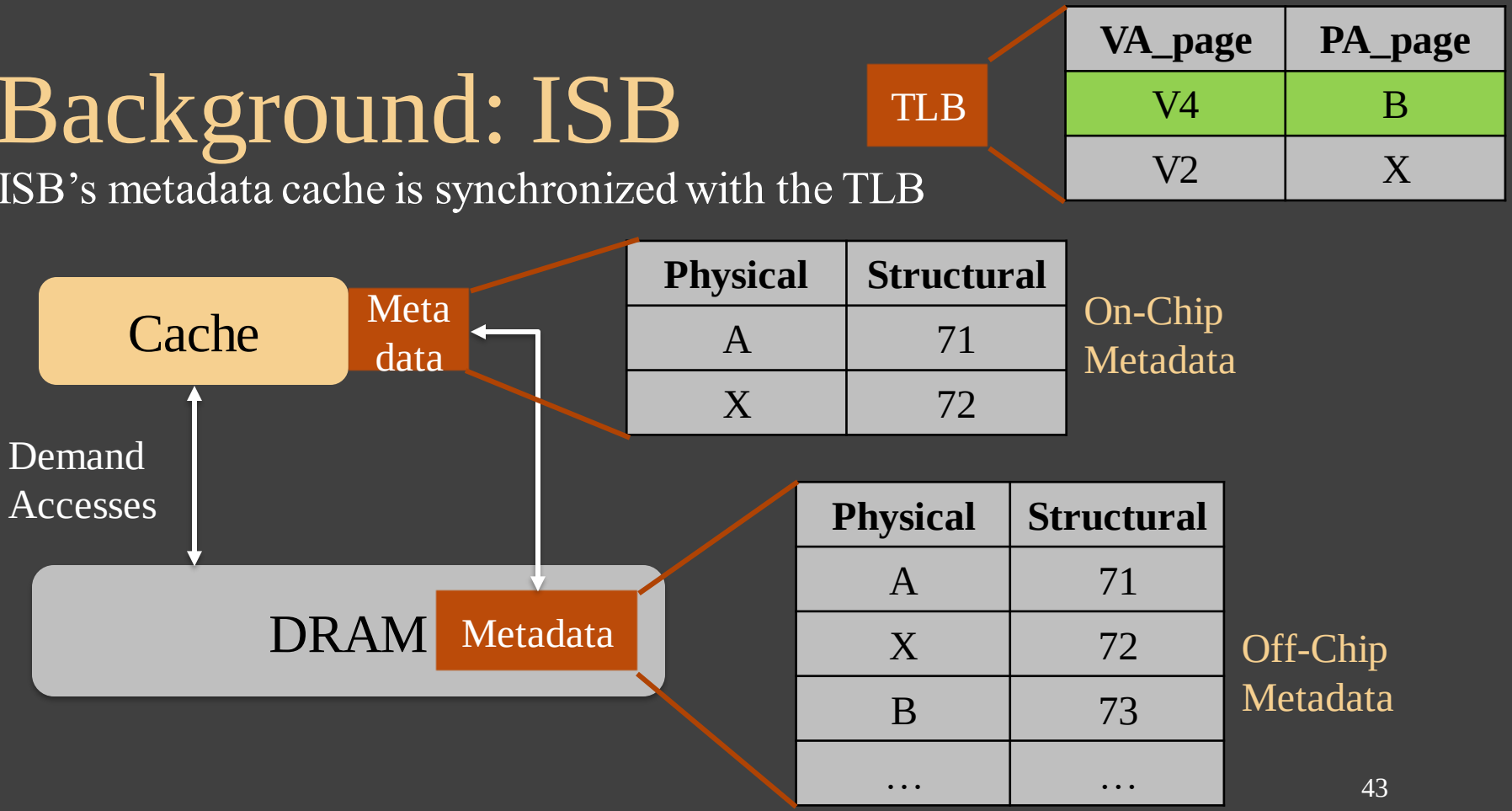
On-Chip Metadata

Physical	Structural
A	71
X	72
B	73
…	…

Off-Chip Metadata

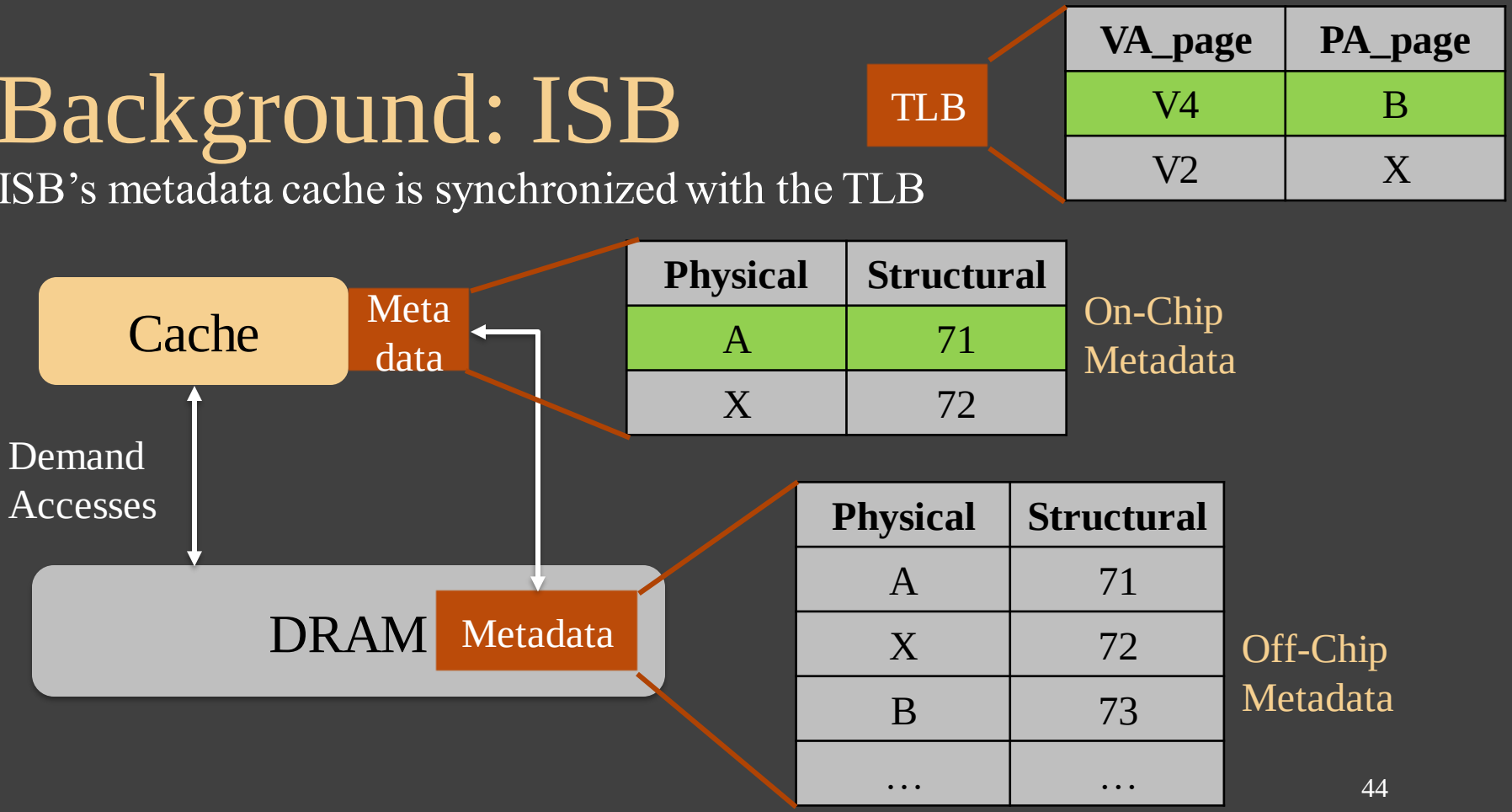
Background: ISB

ISB's metadata cache is synchronized with the TLB



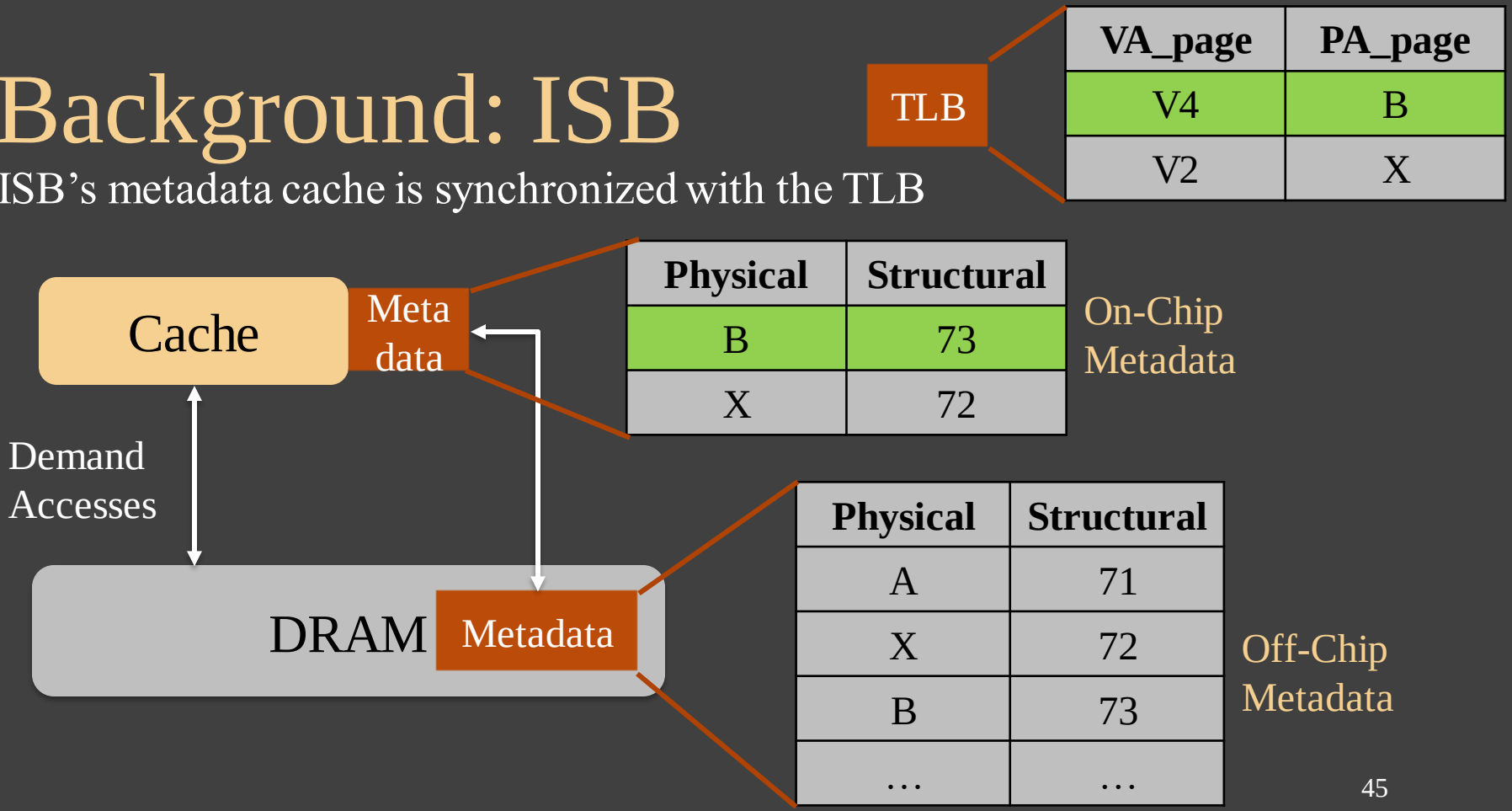
Background: ISB

ISB's metadata cache is synchronized with the TLB



Background: ISB

ISB's metadata cache is synchronized with the TLB



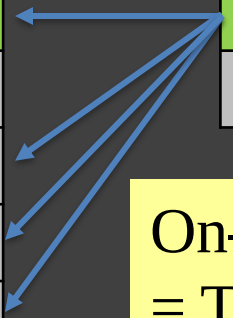
Deficiencies of ISB

On-Chip Metadata

Physical	Structural
B	73
X	72
B+1	INVALID
…	…
B+63	INVALID

TLB

VA_page	PA_page
V4	B
V2	X



On-Chip Metadata Size Required
= TLB Size * Cache Lines Per Page

Deficiencies of ISB

On-Chip Metadata Size Required
= TLB Size * Cache Lines Per Page

- Metadata is managed at coarse granularity
  - ~90% traffic is useless due to lack of spatial locality
- Metadata size is proportional to page size
  - ISB does not scale to large pages
- Metadata size is proportional to TLB size
  - ISB does not work for two-level TLBs

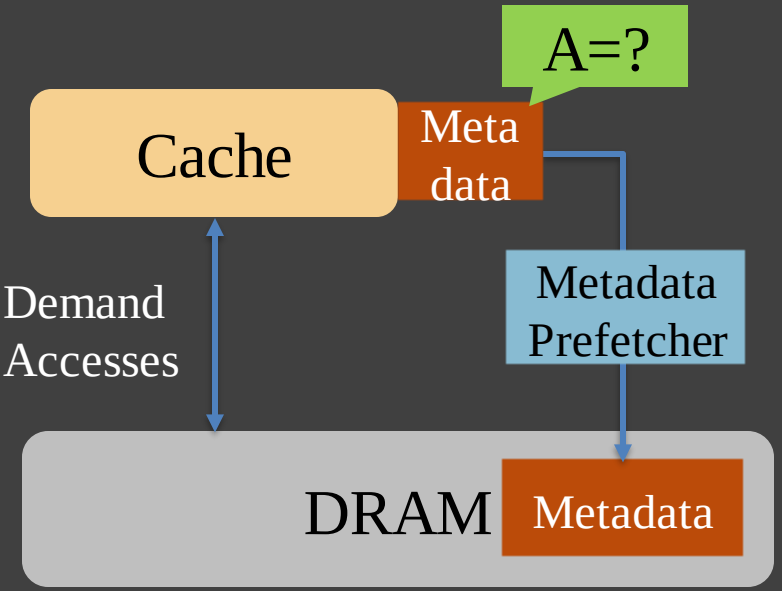
Components of MISB

- Manage metadata at a fine granularity
- Prefetch metadata
- Filter out unnecessary accesses

Components of MISB

- Manage metadata at a fine granularity
- Prefetch metadata
- Filter out unnecessary accesses

MISB Operation



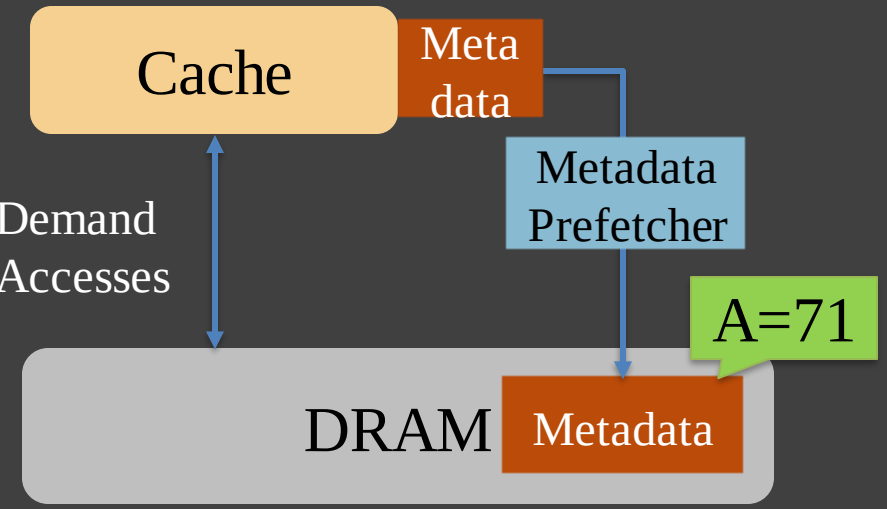
On-Chip Metadata

Physical	Structural
A	?

Off-Chip Metadata

Physical	Structural
A	71
X	72
B	73

MISB Operation



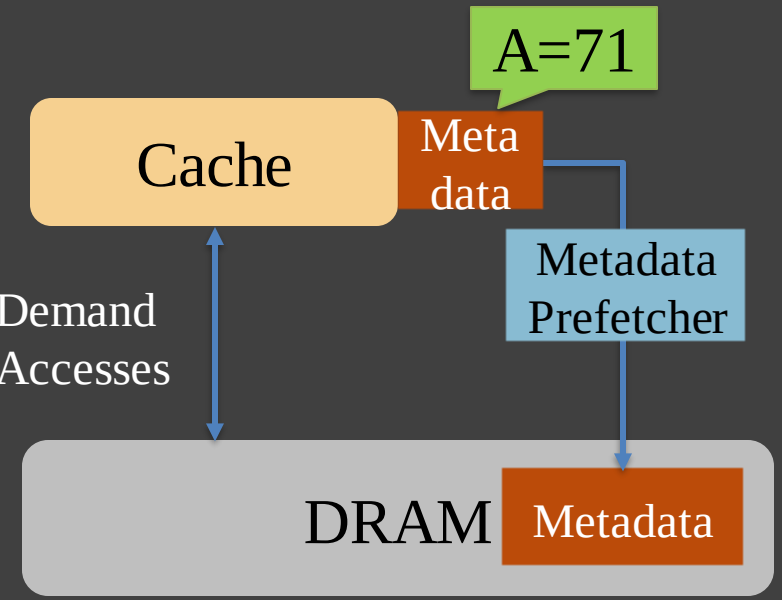
On-Chip Metadata

Physical	Structural
A	?

Off-Chip Metadata

Physical	Structural
A	71
X	72
B	73

MISB Operation



On-Chip Metadata

Physical	Structural
A	71

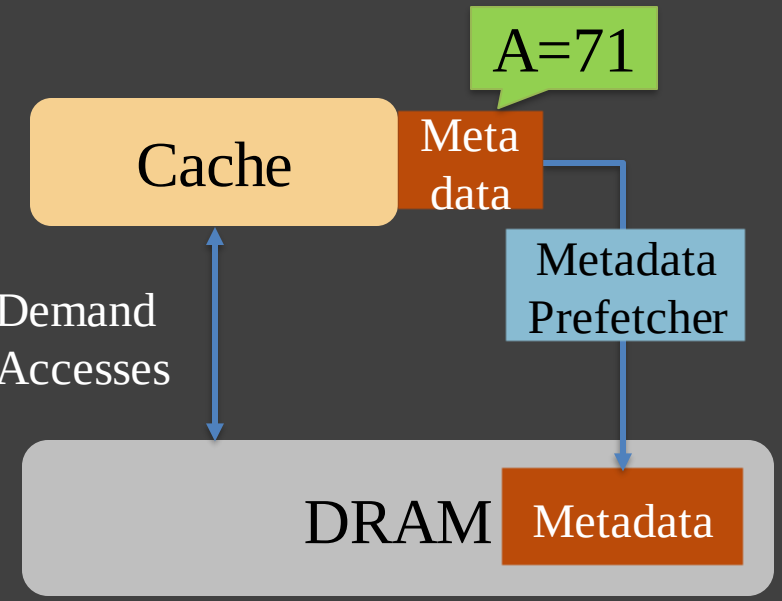
Off-Chip Metadata

Physical	Structural
A	71
X	72
B	73

Components of MISB

- Manage metadata at a fine granularity
- Prefetch metadata
- Filter out unnecessary accesses

MISB Operation



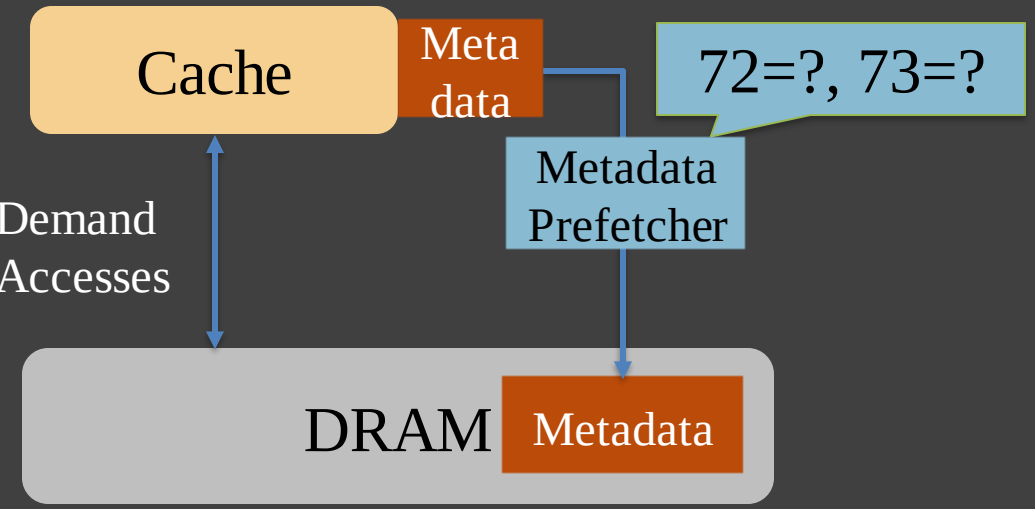
On-Chip Metadata

Physical	Structural
A	71

Off-Chip Metadata

Physical	Structural
A	71
X	72
B	73

MISB Operation



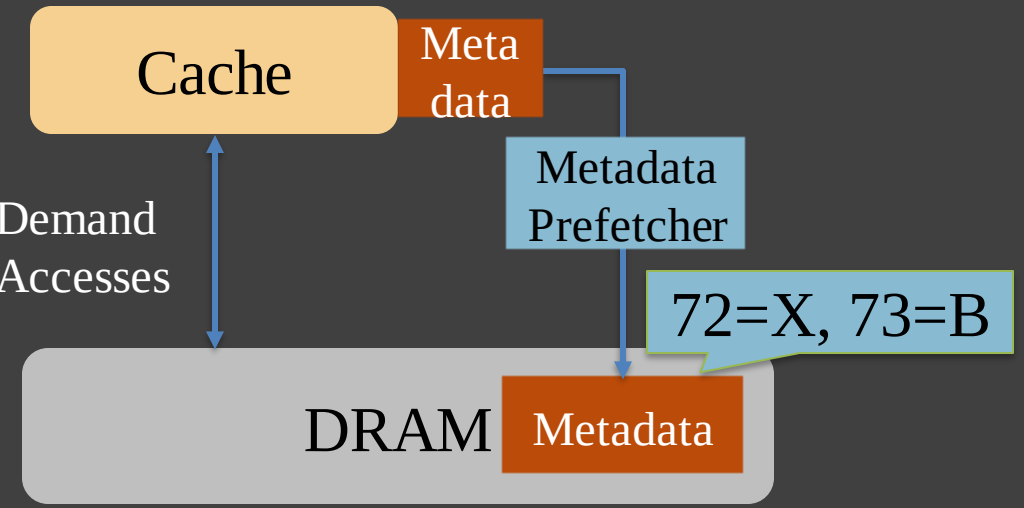
On-Chip Metadata

Physical	Structural
A	71

Off-Chip Metadata

Physical	Structural
A	71
X	72
B	73

MISB Operation



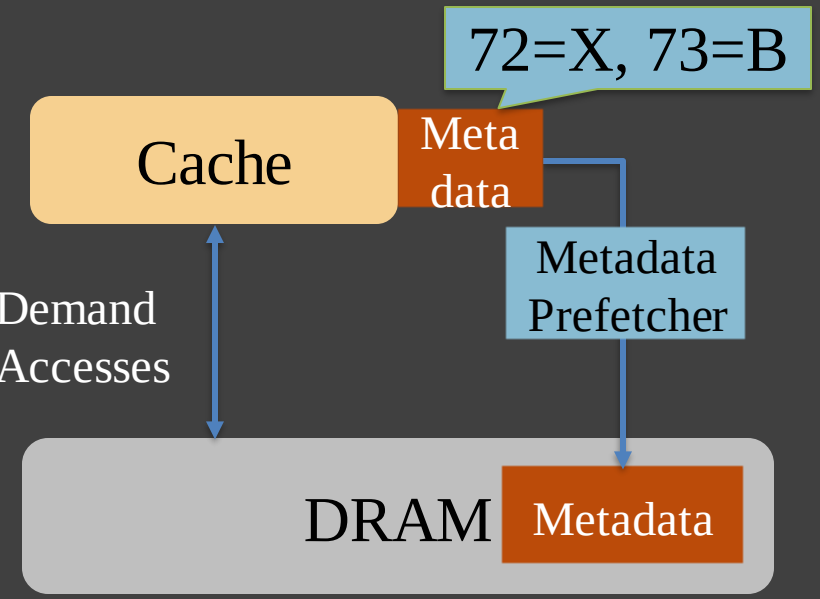
On-Chip Metadata

Physical	Structural
A	71

Off-Chip Metadata

Physical	Structural
A	71
X	72
B	73

MISB Operation



On-Chip Metadata

Physical	Structural
A	71
X	72
B	73

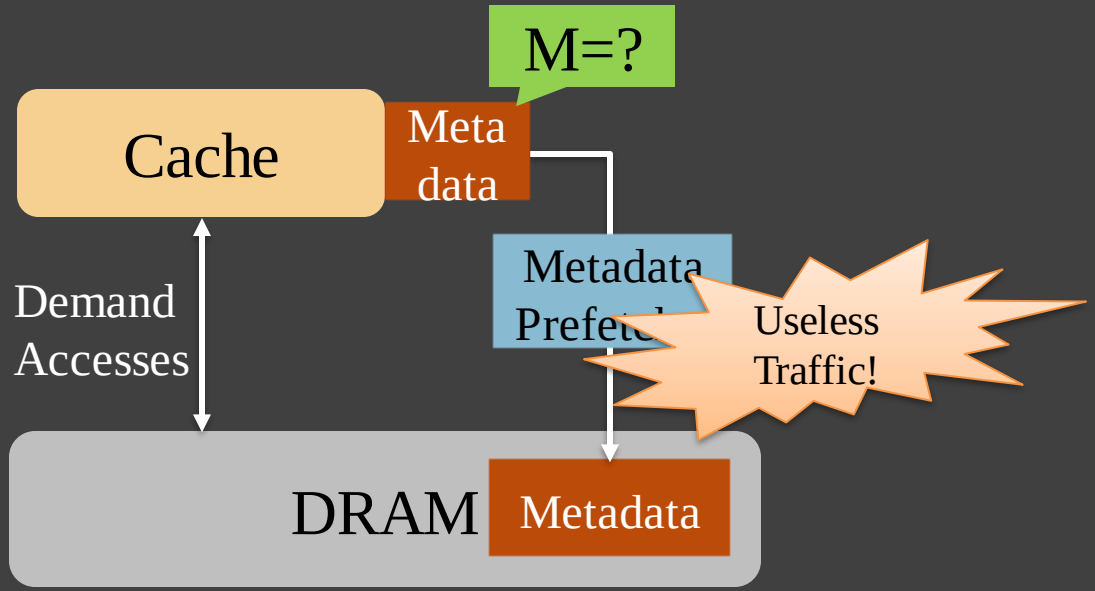
Off-Chip Metadata

Physical	Structural
A	71
X	72
B	73

Components of MISB

- Manage metadata at a fine granularity
- Prefetch metadata
- Filter out unnecessary accesses

MISB Operation



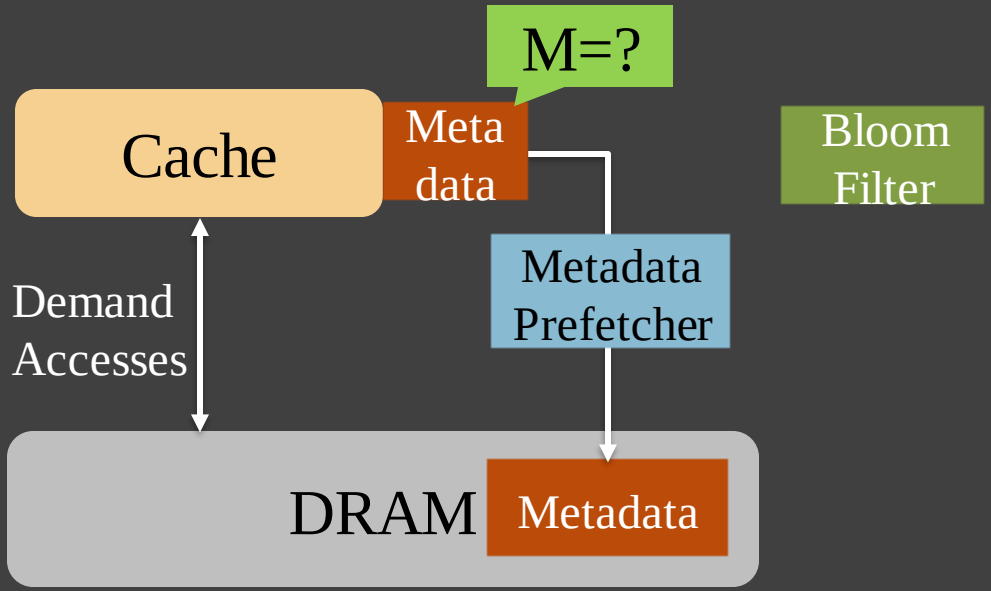
On-Chip Metadata

Physical	Structural
M	?

Off-Chip Metadata

Physical	Structural
A	71
X	72
B	73

MISB Operation



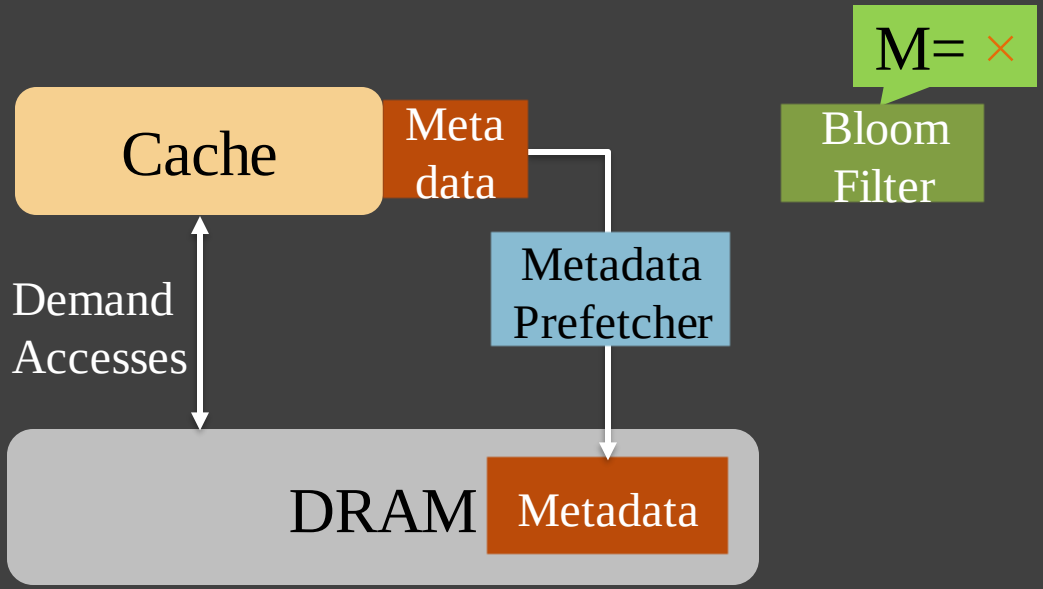
On-Chip Metadata

Physical	Structural
M	?

Off-Chip Metadata

Physical	Structural
A	71
X	72
B	73

MISB Operation



On-Chip Metadata

Physical	Structural
M	?

Off-Chip Metadata

Physical	Structural
A	71
X	72
B	73

Components of MISB

- Manage metadata at a fine granularity
- Prefetch metadata
- Filter out unnecessary accesses

Evaluation Methodology

- Industrial Simulator
  - ARMv8 AArch64
  - OoO Core
  - 2-level TLB
  - Bandwidth: 32GB/s
- Multicore – ChampSim
  - Similar trends as the industrial simulator
- SPEC2006
  - Irregular Subset
- CloudSuite

Evaluated Prefetchers

STMS & Domino

- Global correlation

ISB

- PC localization

MISB

- PC localization

Evaluated Prefetchers

Idealized

STMS & Domino

- Global correlation
- Metadata not cacheable

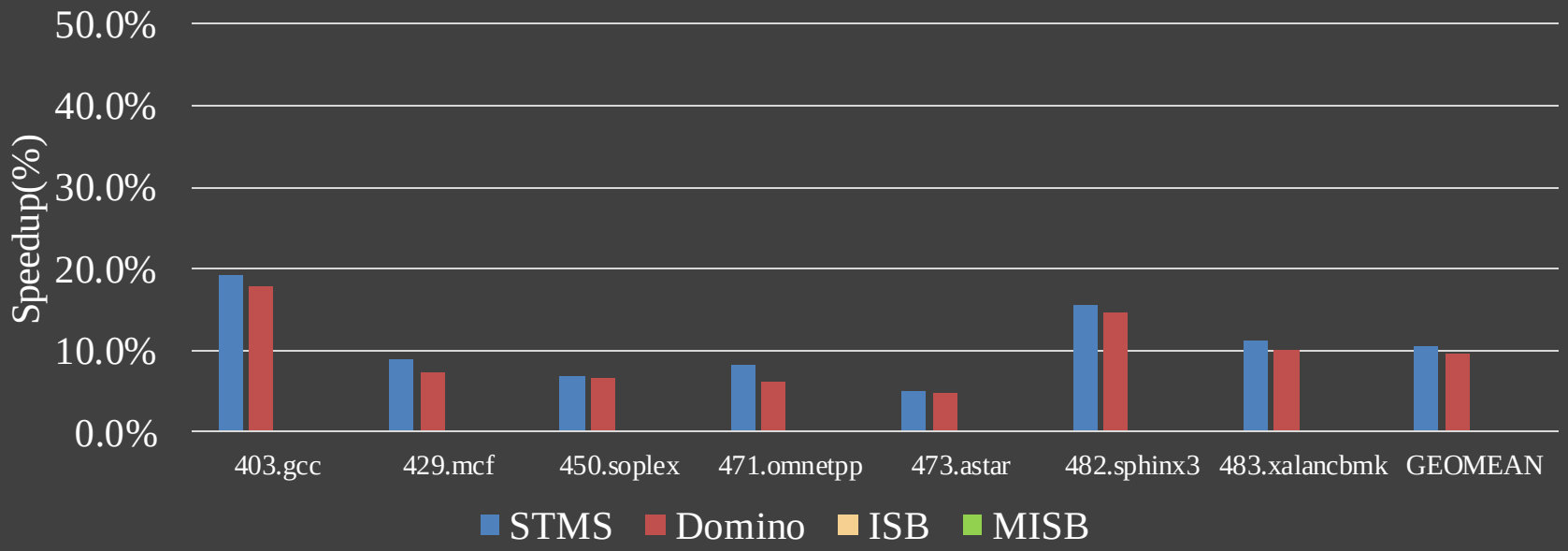
ISB

- PC localization
- Metadata cacheable
- Syncs metadata with TLB

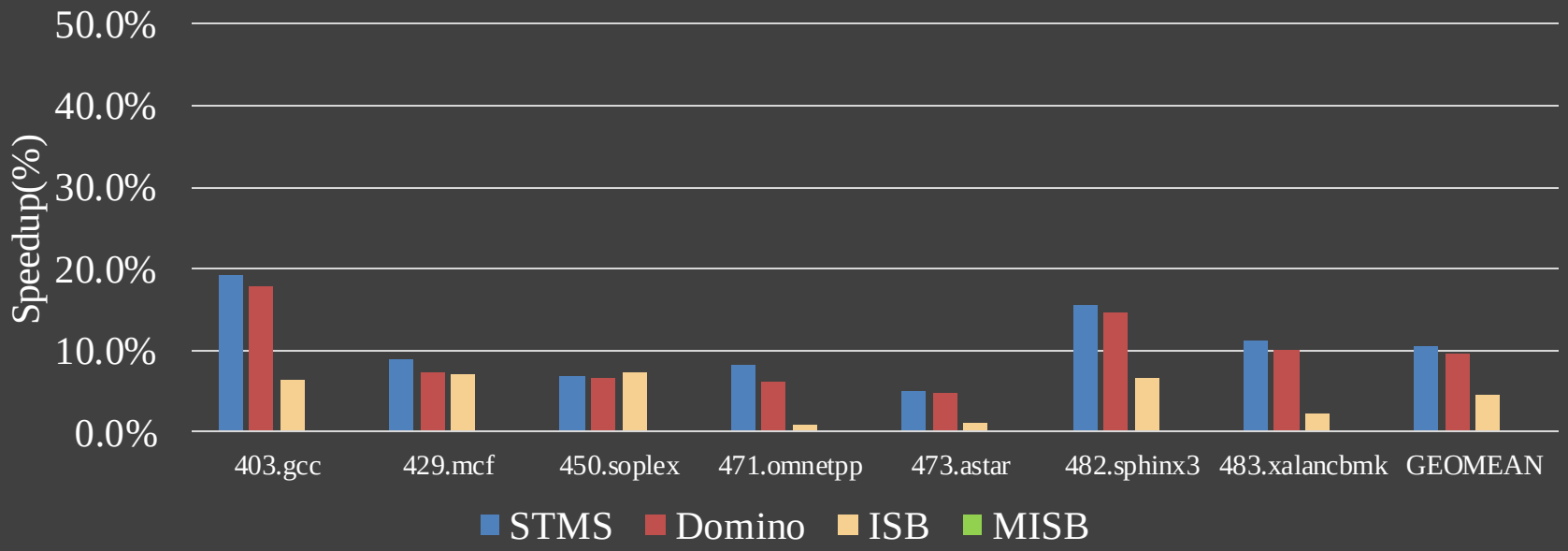
MISB

- PC localization
- Metadata cacheable
- Prefetches metadata

Performance



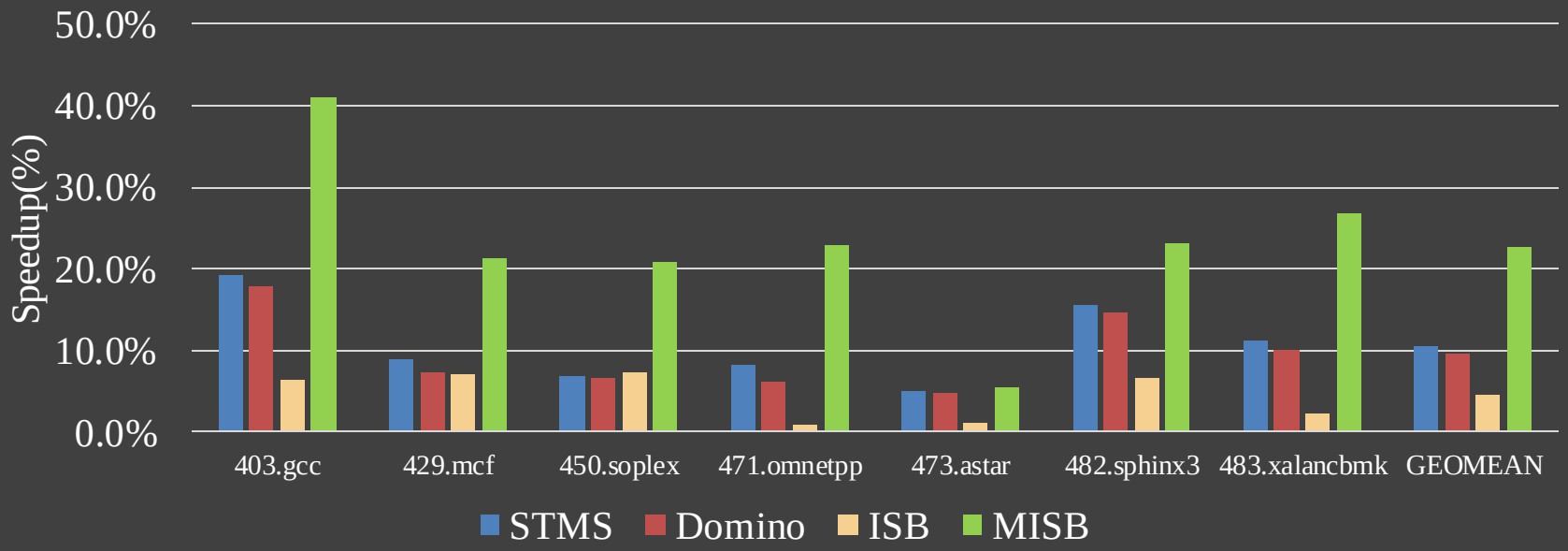
Performance



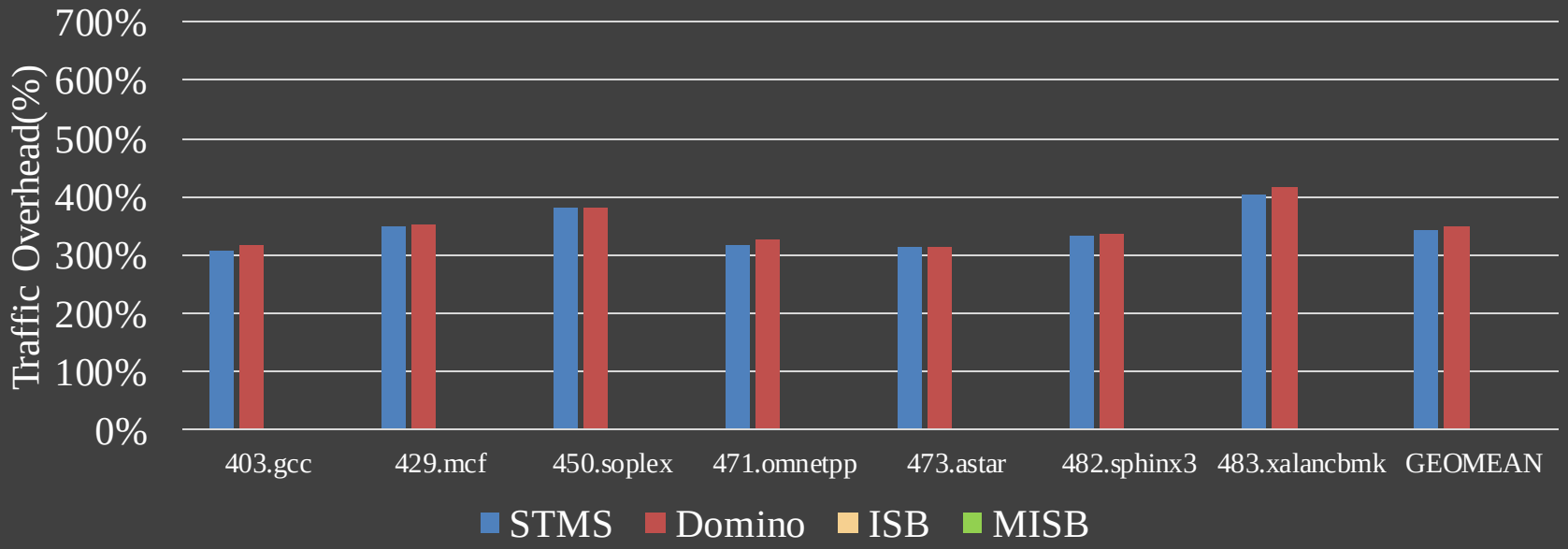
TEXAS
The University of Texas at Austin

arm

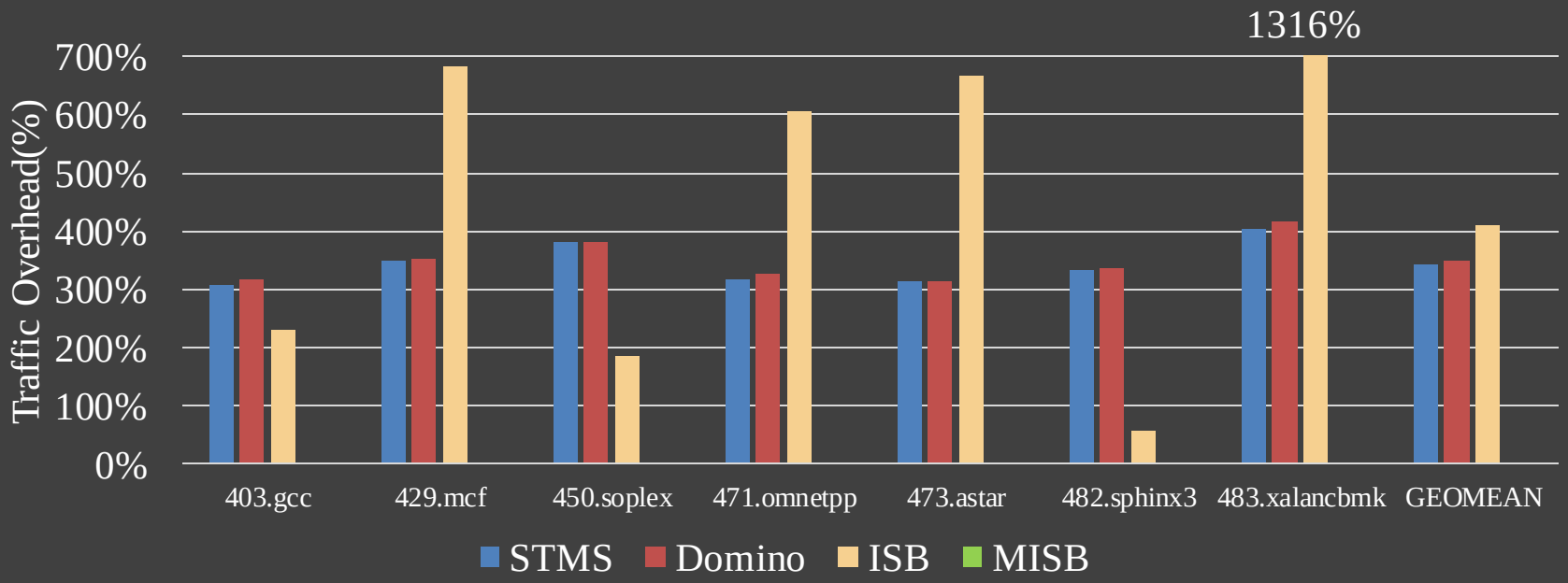
Performance



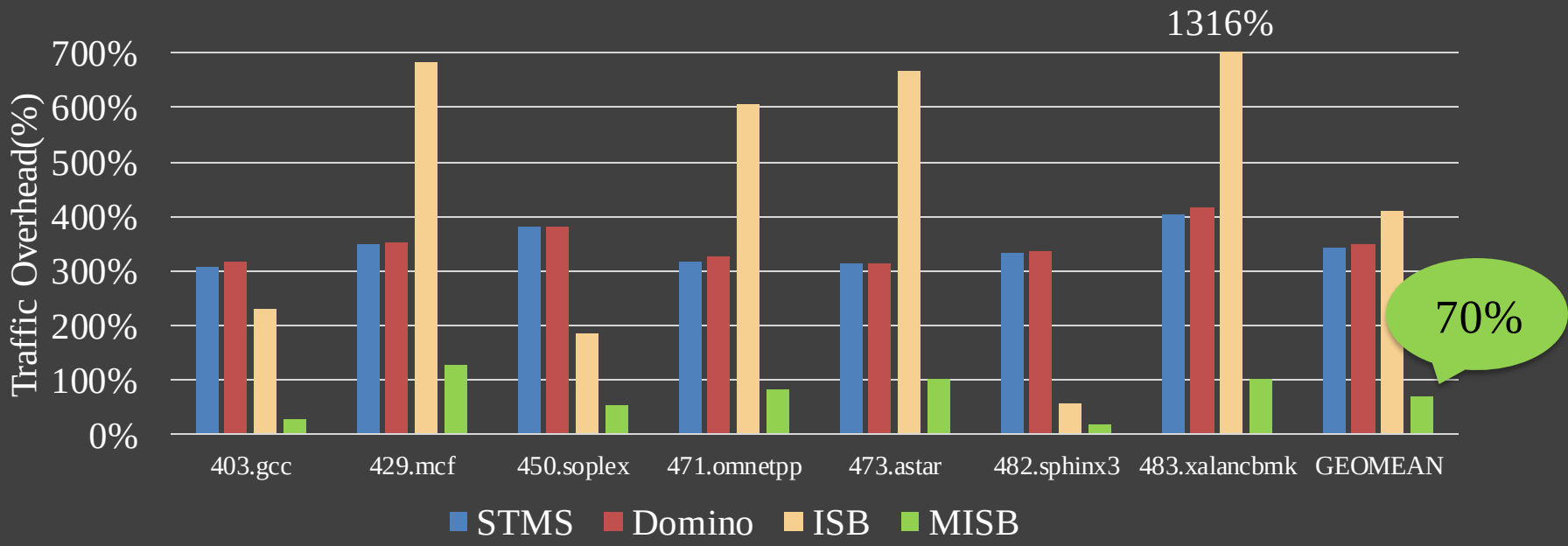
Traffic Overhead



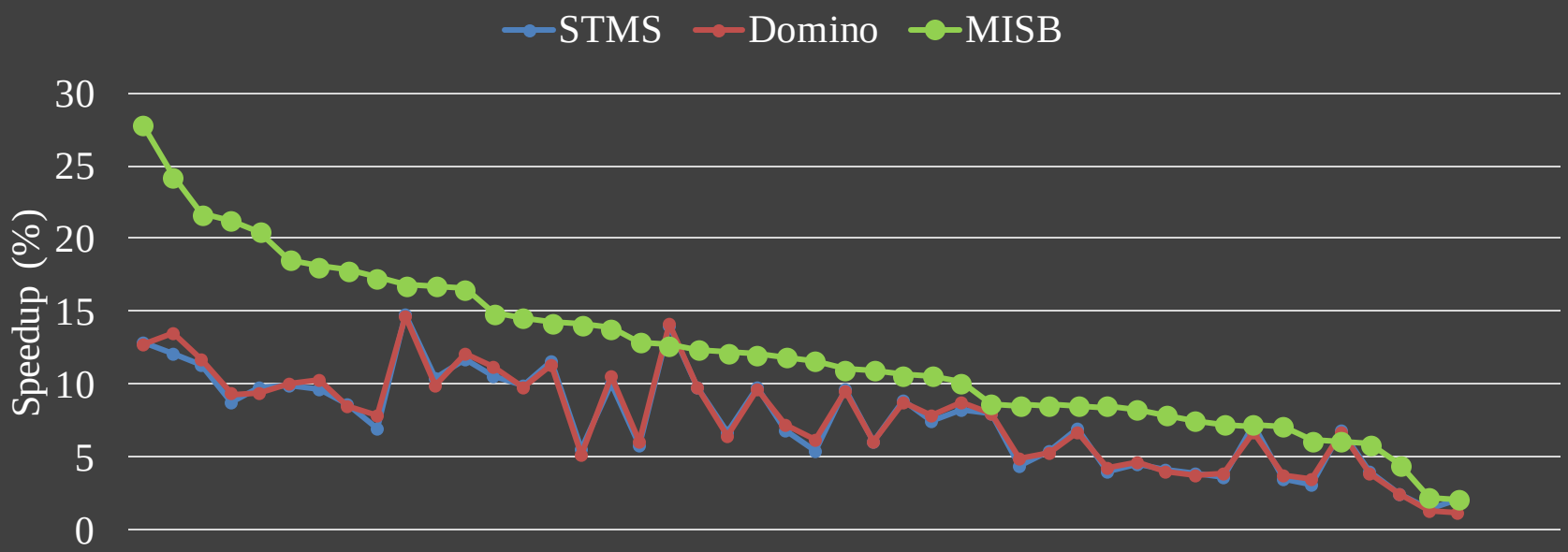
Traffic Overhead



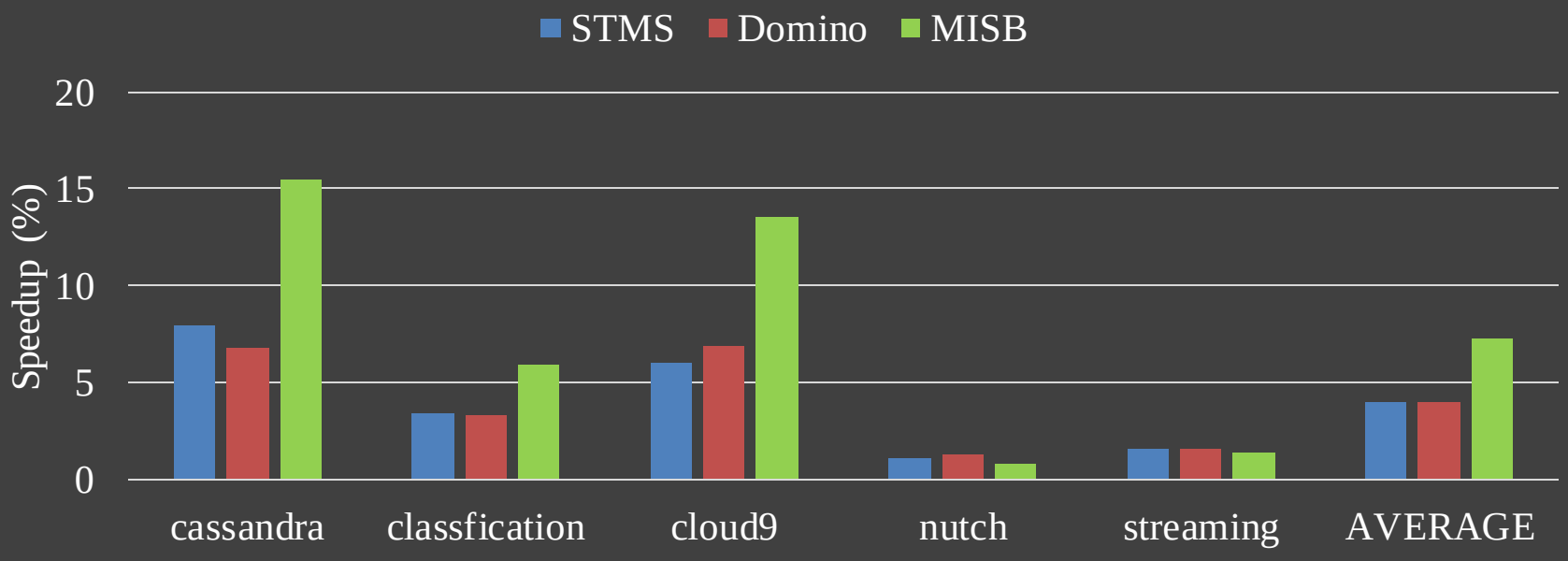
Traffic Overhead



Performance - SPEC2006 on 8 Cores



Performance – CloudSuite on 4 cores



Traffic – CloudSuite on 4 Cores