

Understanding Branch Target Buffer Tag Sizing

Roman K. Brunner and Rakesh Kumar

Norwegian University of Science and Technology

 **NTNU** | Department of Computer Science
Computer Architecture Lab

1-Minute Overview

Problem:

- The core frontend is a major bottleneck
- Restricted BTB capacity is a major performance limiter in the core frontend
- Commercial CPUs feature massive BTB footprints

Insights:

- Conventional BTB: Targets occupy most of the storage
- Recent research has optimized target representation
- Tags emerge as the most storage consuming component in the BTB

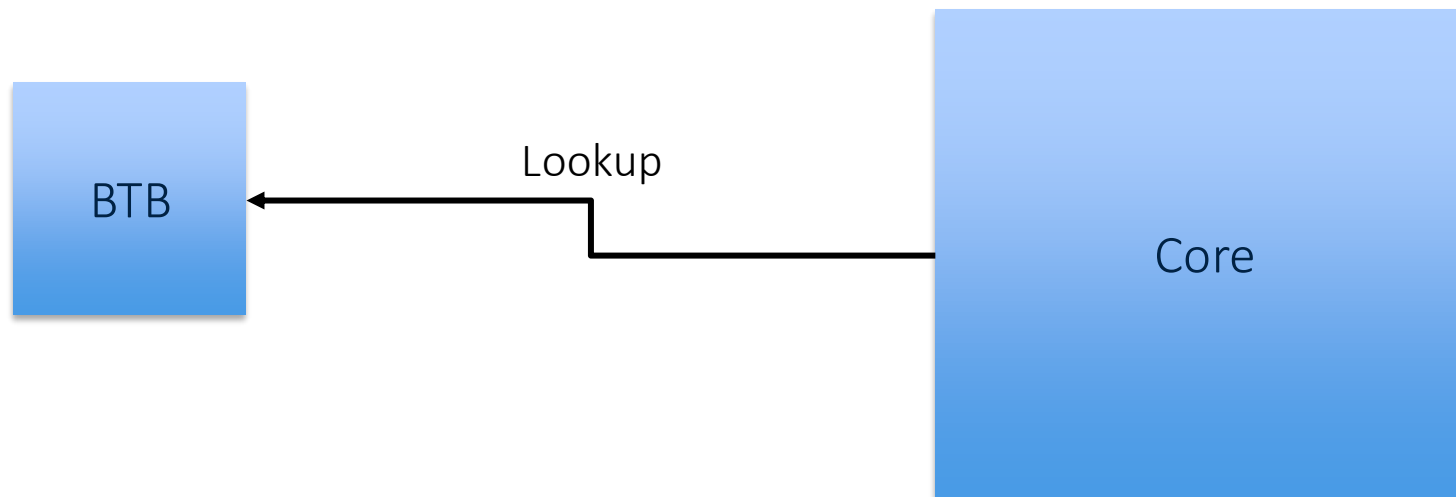
Goal:

- Performance impact of tag sizes across BTB capacities and configurations

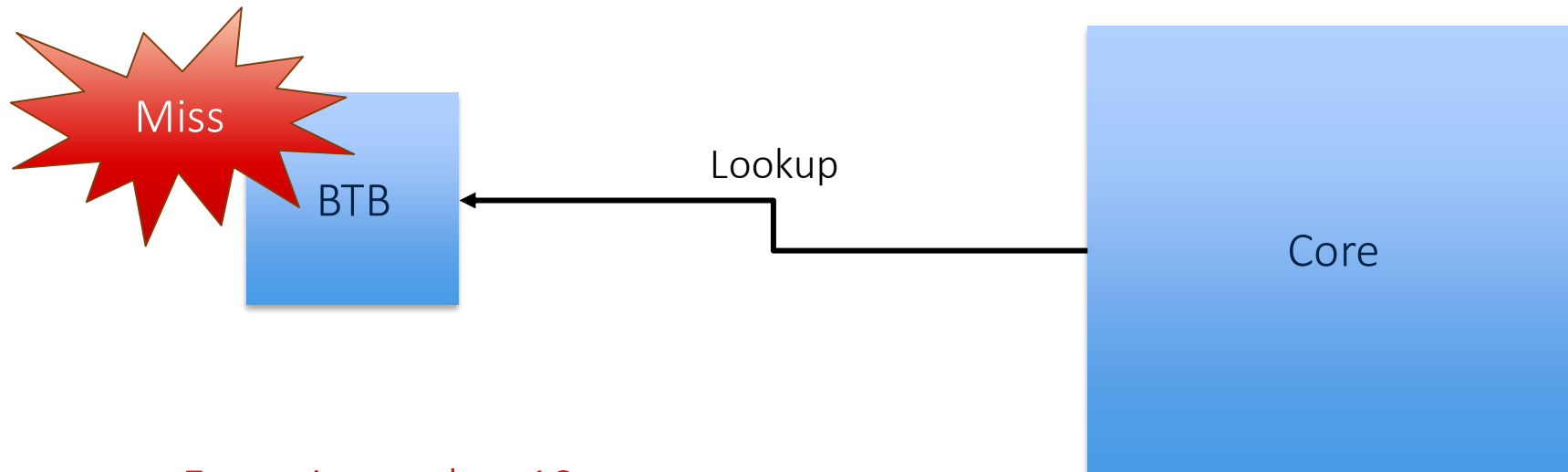
BTB: The Key to Frontend Performance

- BTB: identifies branches and their targets
- Crucial for frontend performance
 - Drives the instruction fetch
 - Identifies conditional branches for the direction predictor
 - Enables a key class of instruction prefetchers (FDIP)

Anatomy of a BTB Miss

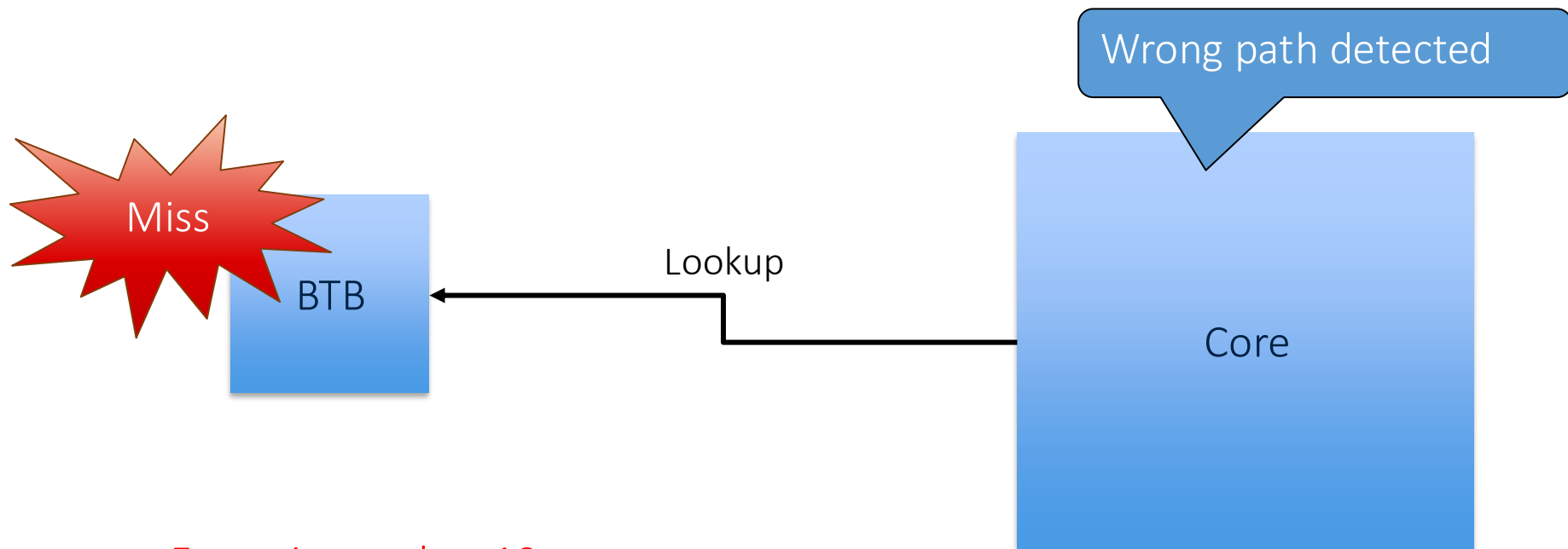


Anatomy of a BTB Miss



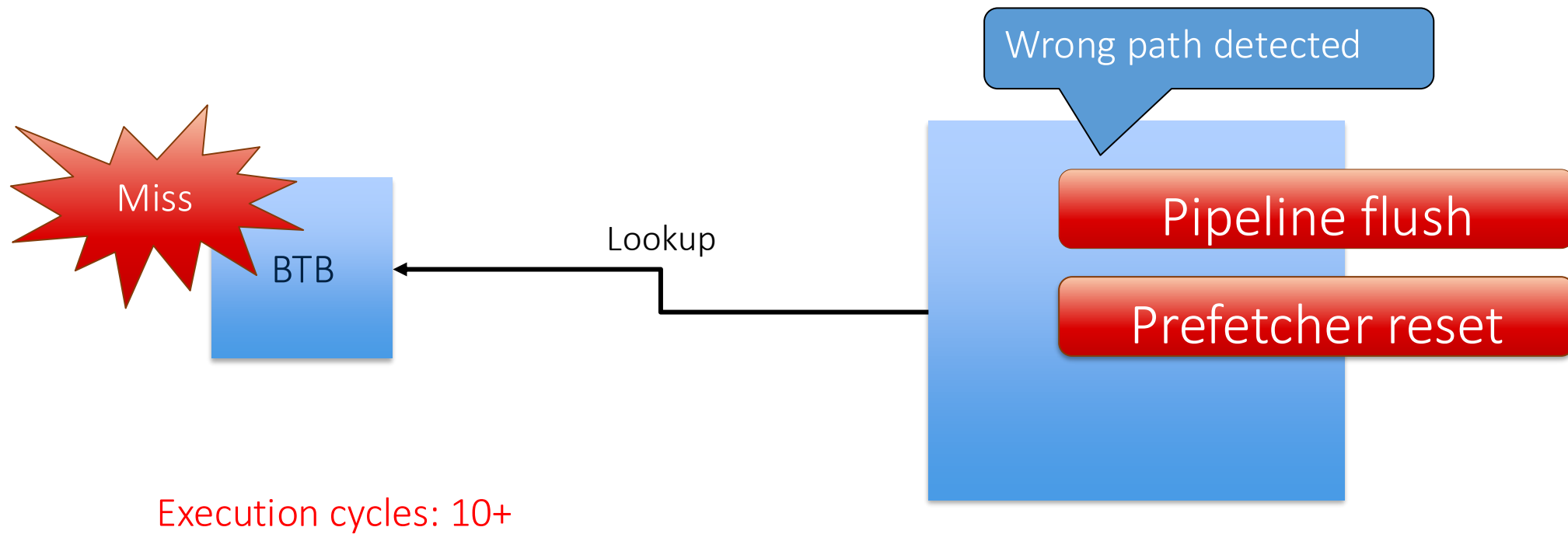
Execution cycles: 10+

Anatomy of a BTB Miss

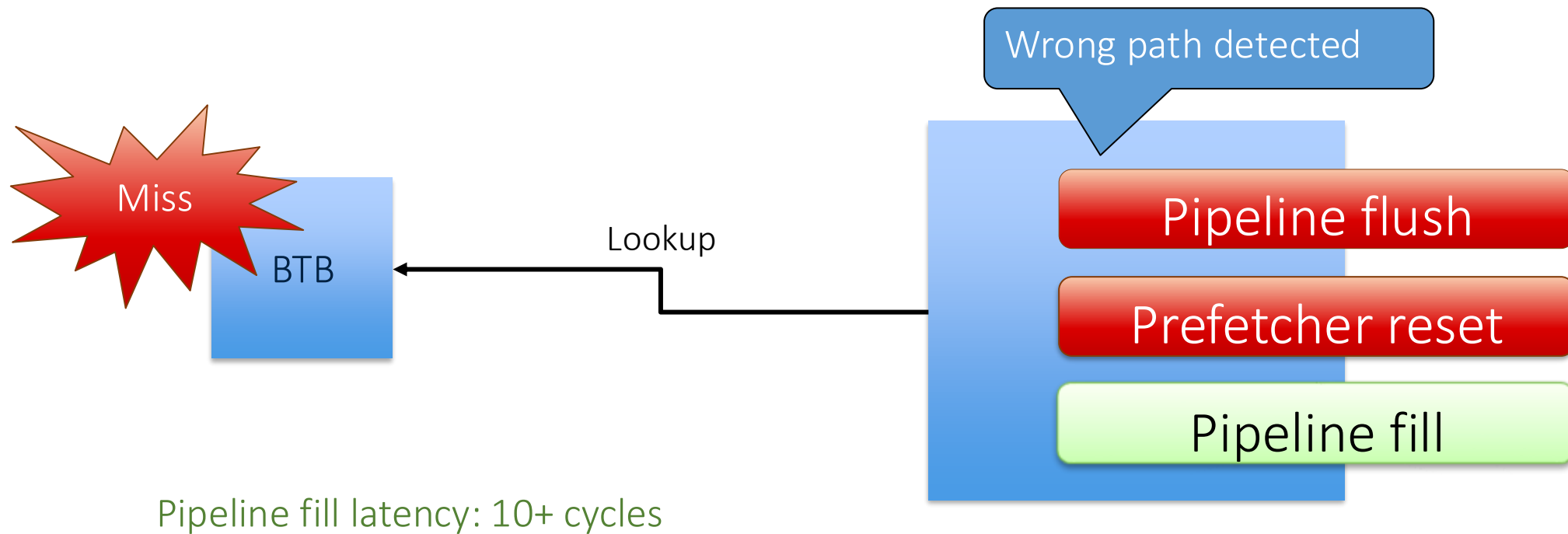


Execution cycles: 10+

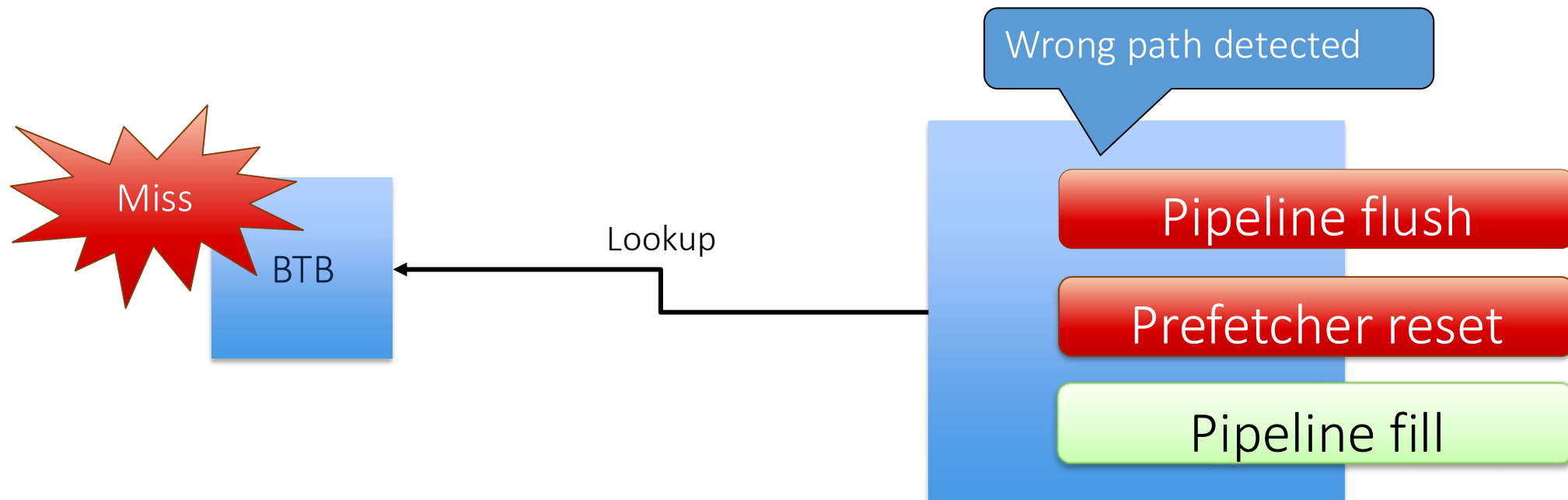
Anatomy of a BTB Miss



Anatomy of a BTB Miss



Anatomy of a BTB Miss



BTB miss recovery costs 10s of cycles → hurting performance

BTB: Big and growing fast

IBM

CPU	Total BTB Entries
zEC12	28K
z13	102K
z14	136K
z15	144K
z16	272K

Samsung

CPU	Total Storage
M1/M2	90.9KB
M3	159.8KB
M4	272.0KB
M5	378.8KB
M6	529.5KB

BTB: Big and growing fast

IBM

CPU	Total BTB Entries
zEC12	28K
z13	102K
z14	136K
z15	144K
z16	272K



10x increase

Samsung

CPU	Total Storage
M1/M2	90.9KB
M3	159.8KB
M4	272.0KB
M5	378.8KB
M6	529.5KB

BTB: Big and growing fast

IBM

CPU	Total BTB Entries
zEC12	28K
z13	102K
z14	136K
z15	144K
z16	272K



10x increase

Samsung

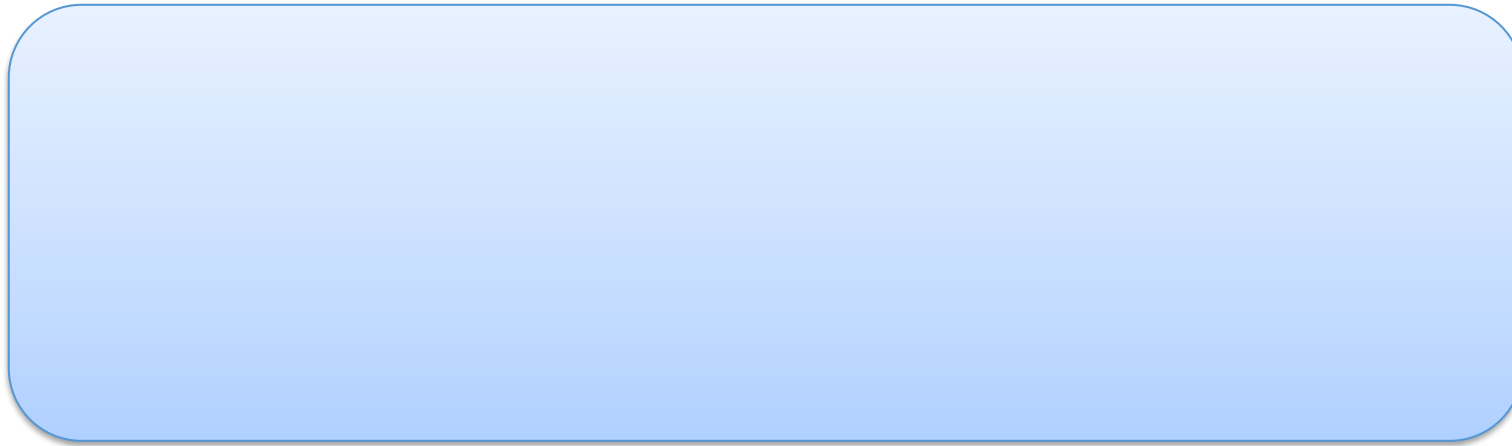
CPU	Total Storage
M1/M2	90.9KB
M3	159.8KB
M4	272.0KB
M5	378.8KB
M6	529.5KB



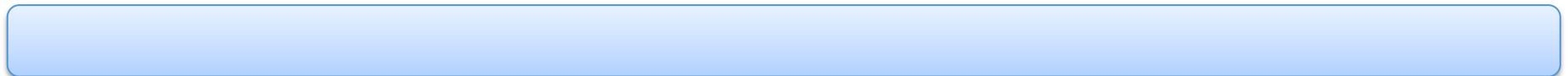
6x increase

BTB Accesses

BTB

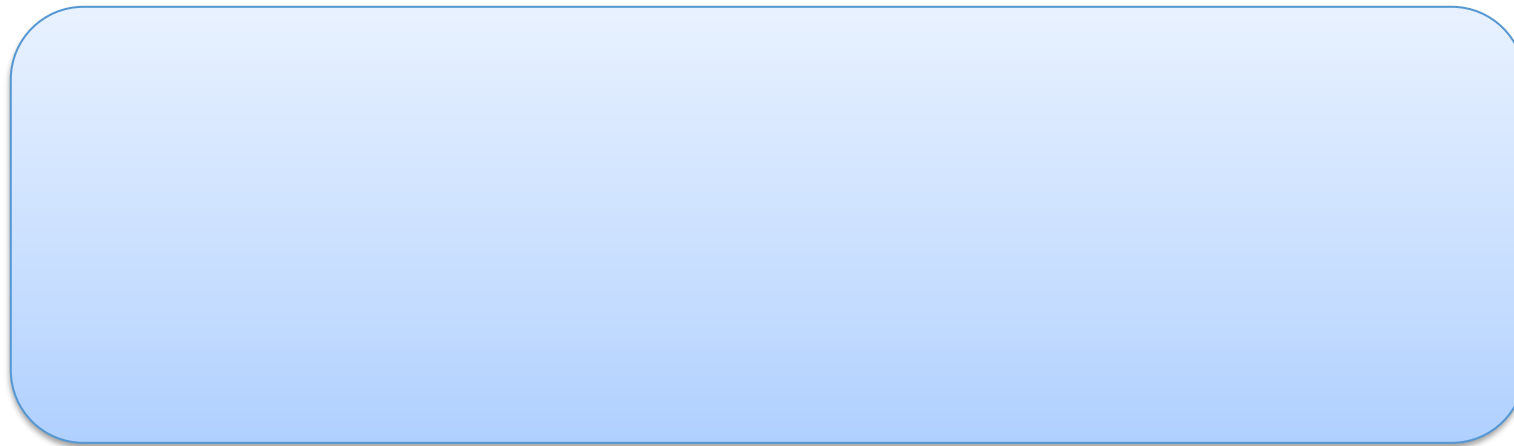


PC



BTB Accesses

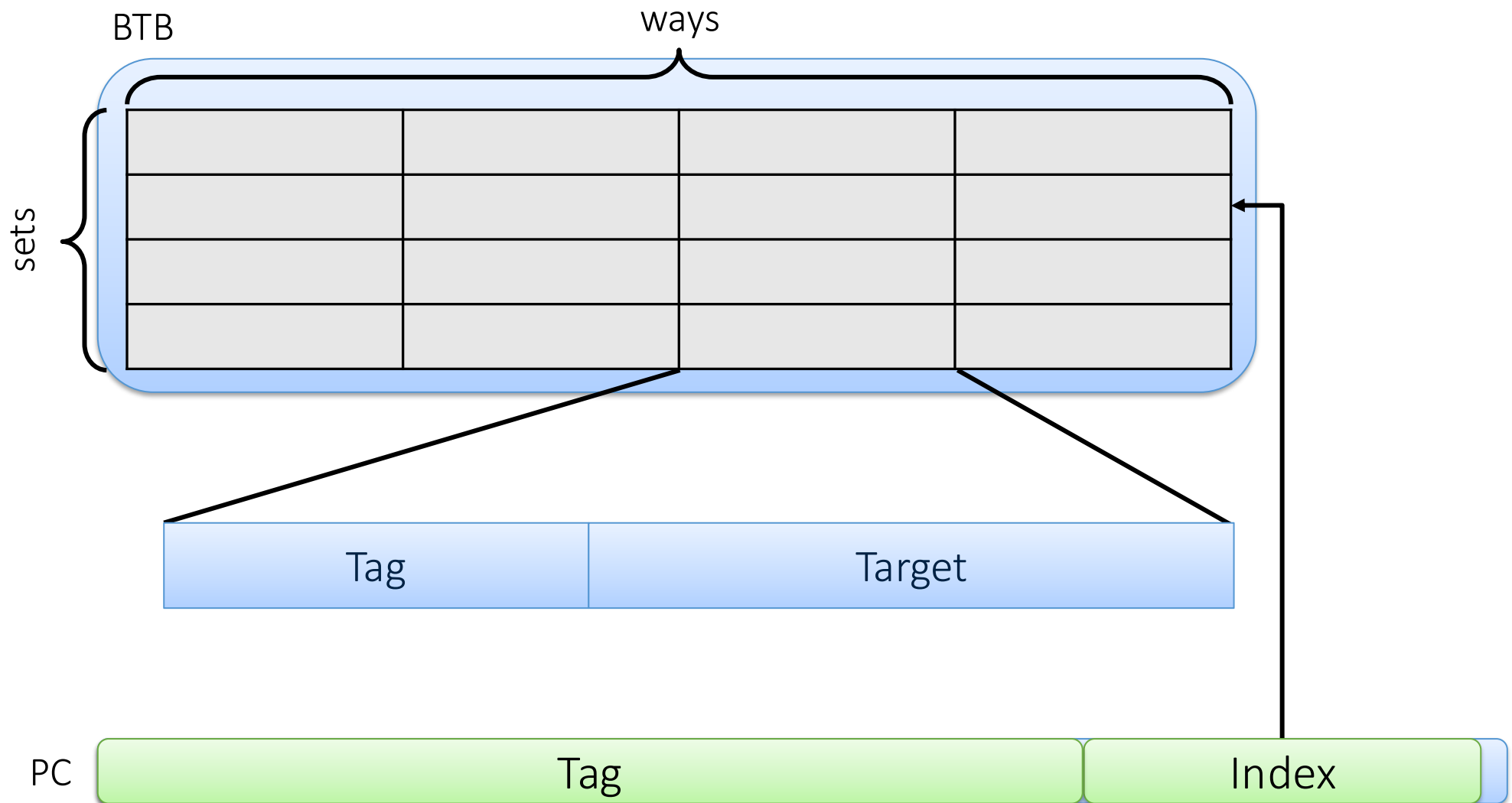
BTB



PC



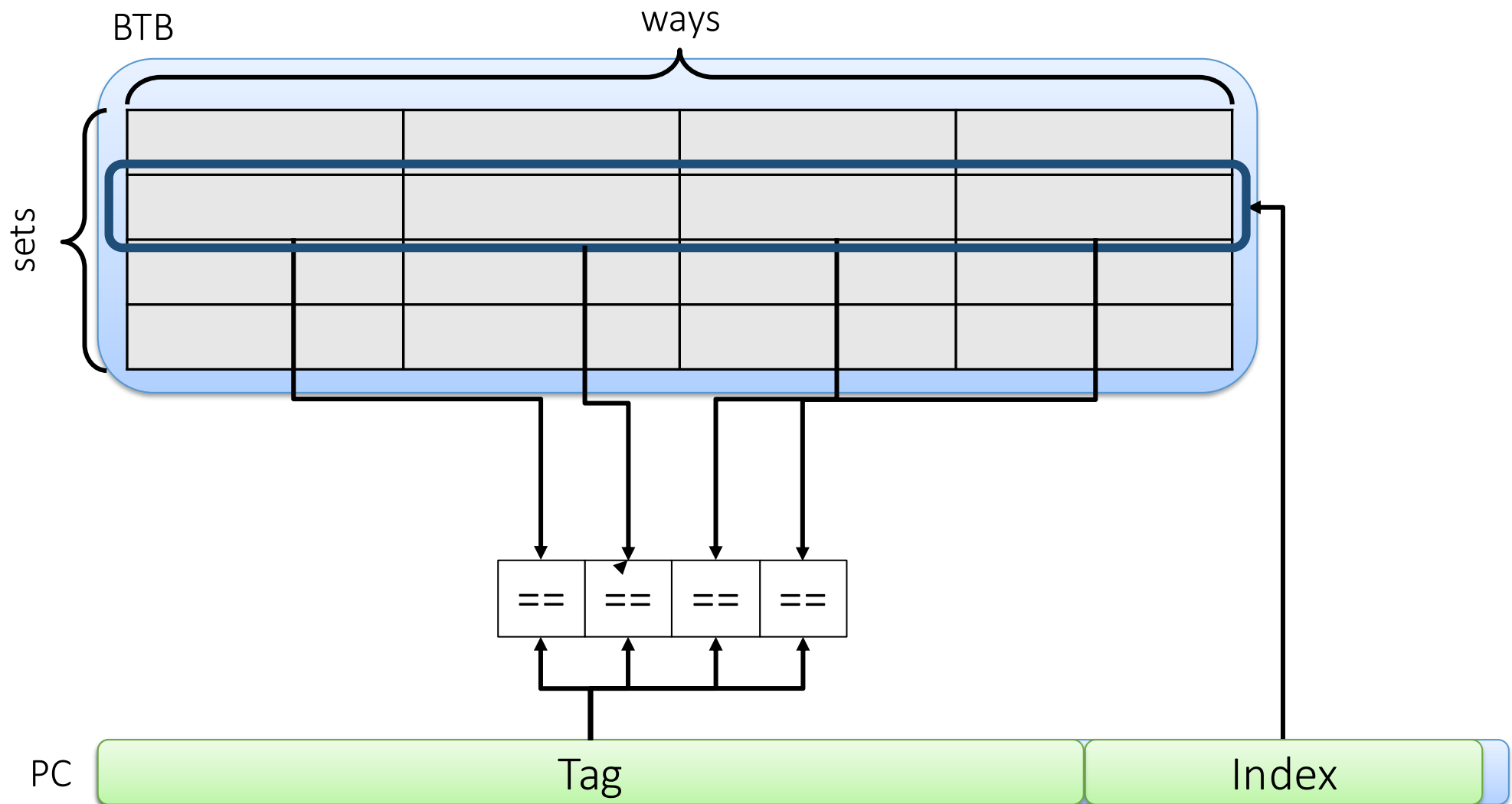
BTB Accesses



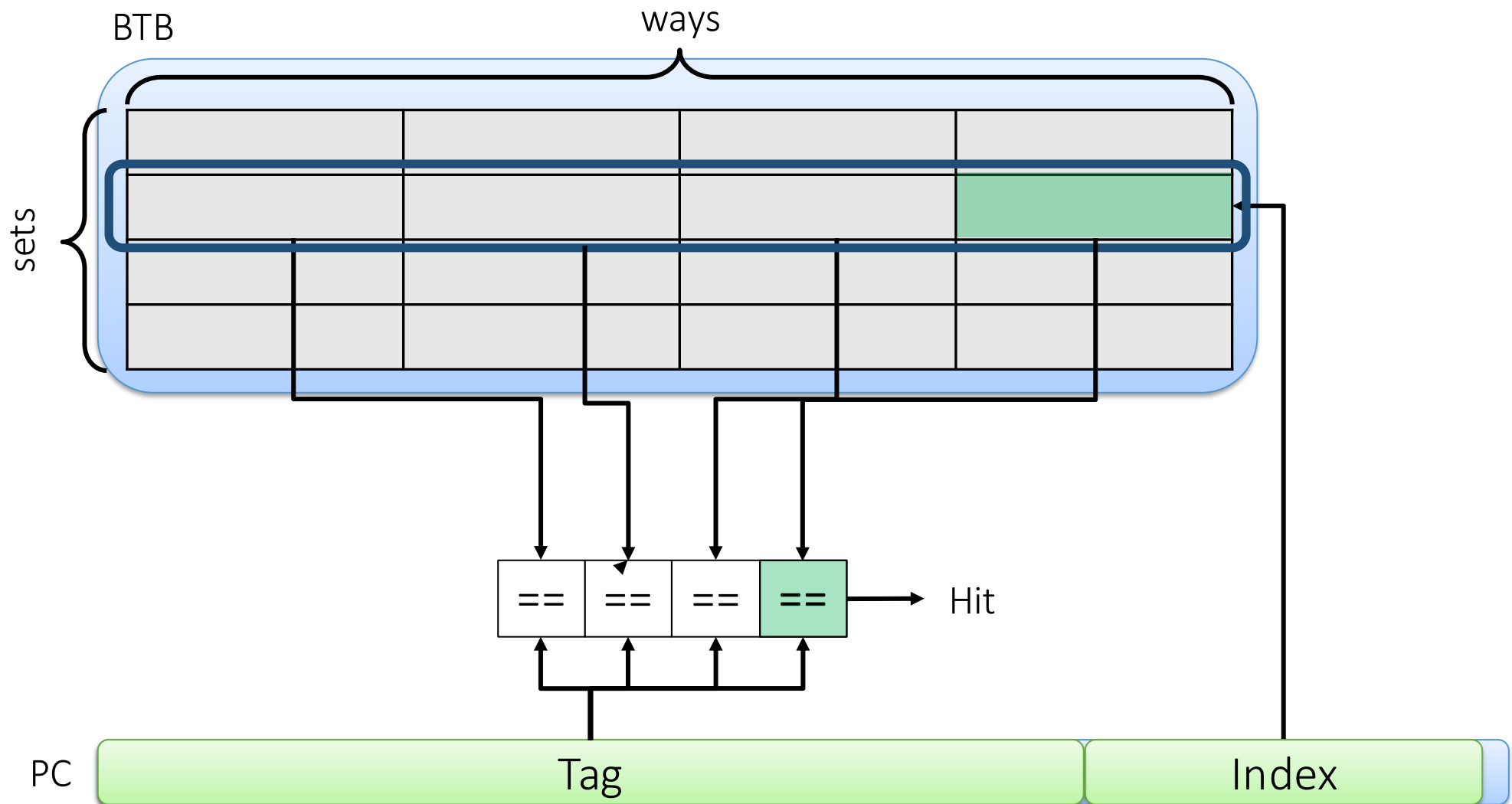
BTB Accesses



BTB Accesses



BTB Accesses



BTB Entry

V	Type	Repl.-Policy	Tag	Target
1b	2b	3b	36b	46b

Conventional BTB

BTB Entry

V 1b	Type 2b	Repl.-Policy 3b	Tag 36b	Target 46b
---------	------------	--------------------	------------	---------------

Conventional BTB

V 1b	Type 2b	Repl.-Policy 3b	Tag 36b	Target 10b
---------	------------	--------------------	------------	---------------

State of the Art BTB

BTB Tag Sizing

Academic:

- BTB Tag Truncation: 2 bit [B. Fagin, K. Russell, MICRO '95]
- Pdede: 12 bit [N. K. Soundararajan et al., MICRO '21]
- BTB-X: 12 bit [T. Asheim et al., HPCA '23]

Industry: [L. Li et al., USENIX Security '24]

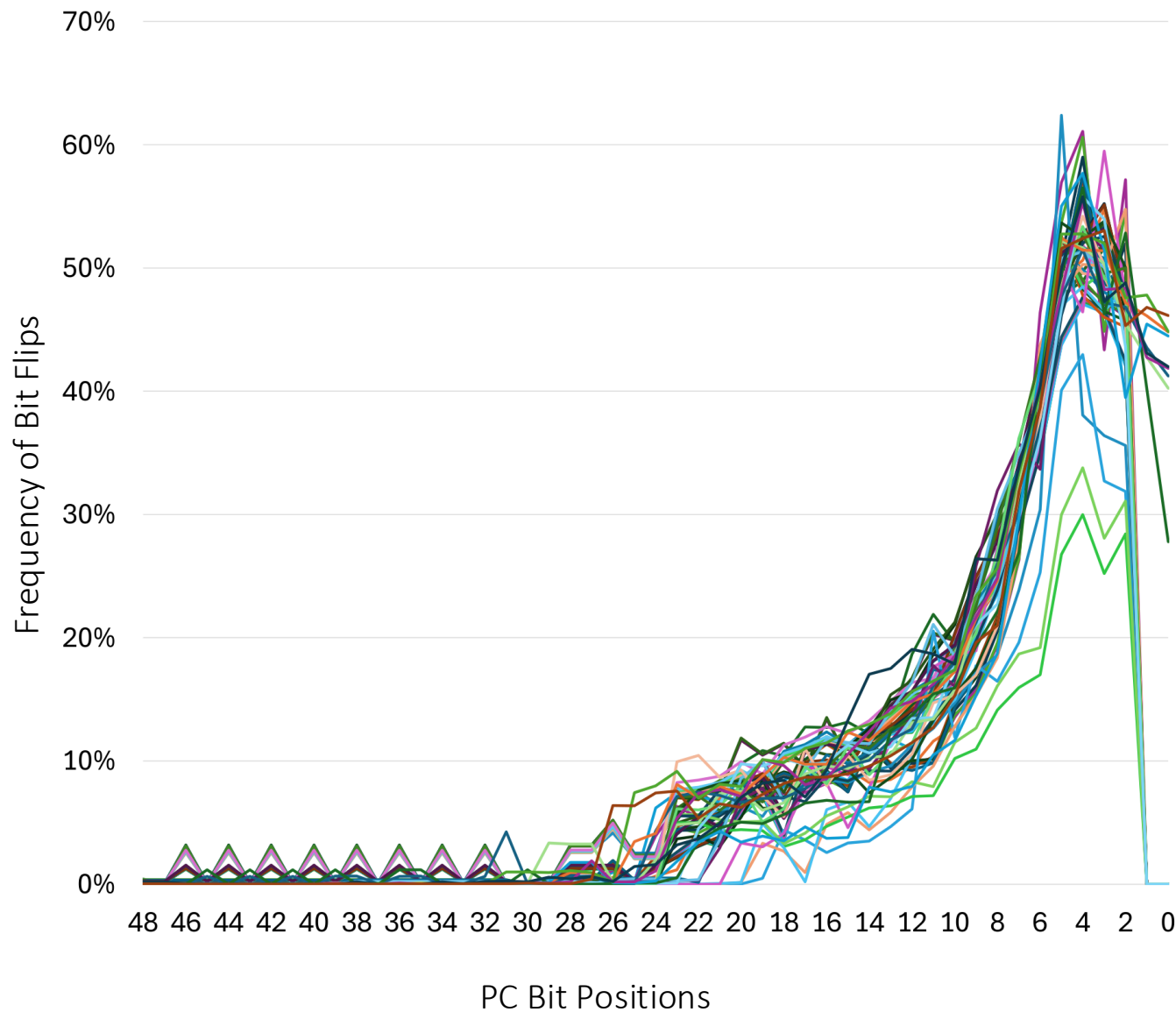
- Skylake: 8 bit
- Raptor/Golden Cove:
 - P Core: 9 bit
 - E core: 10b

How to size the BTB Tag

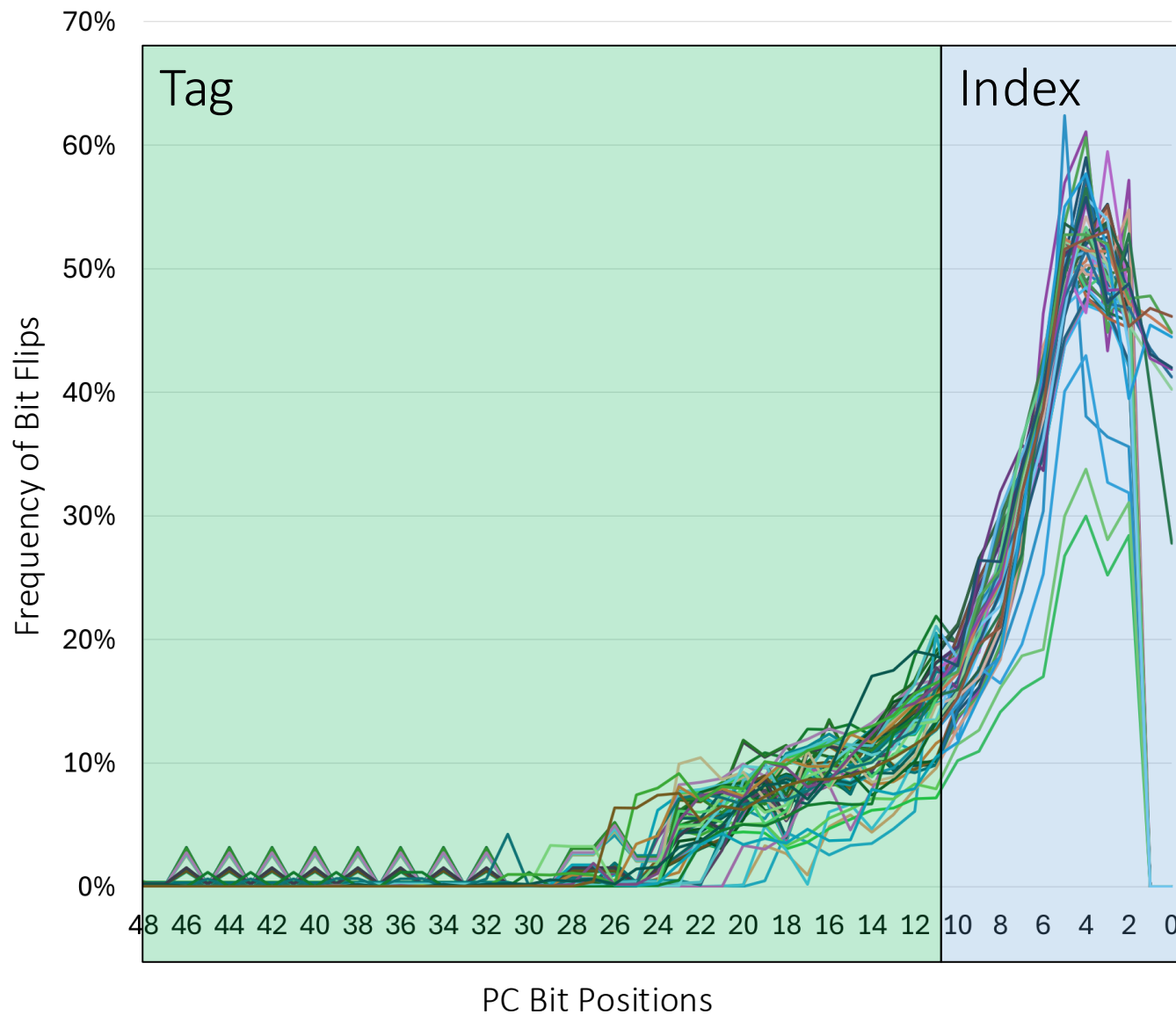
Tag is used to differentiate among branches

→ If a tag bit is the same for most branches it can be dropped with minimal performance loss (<1%)

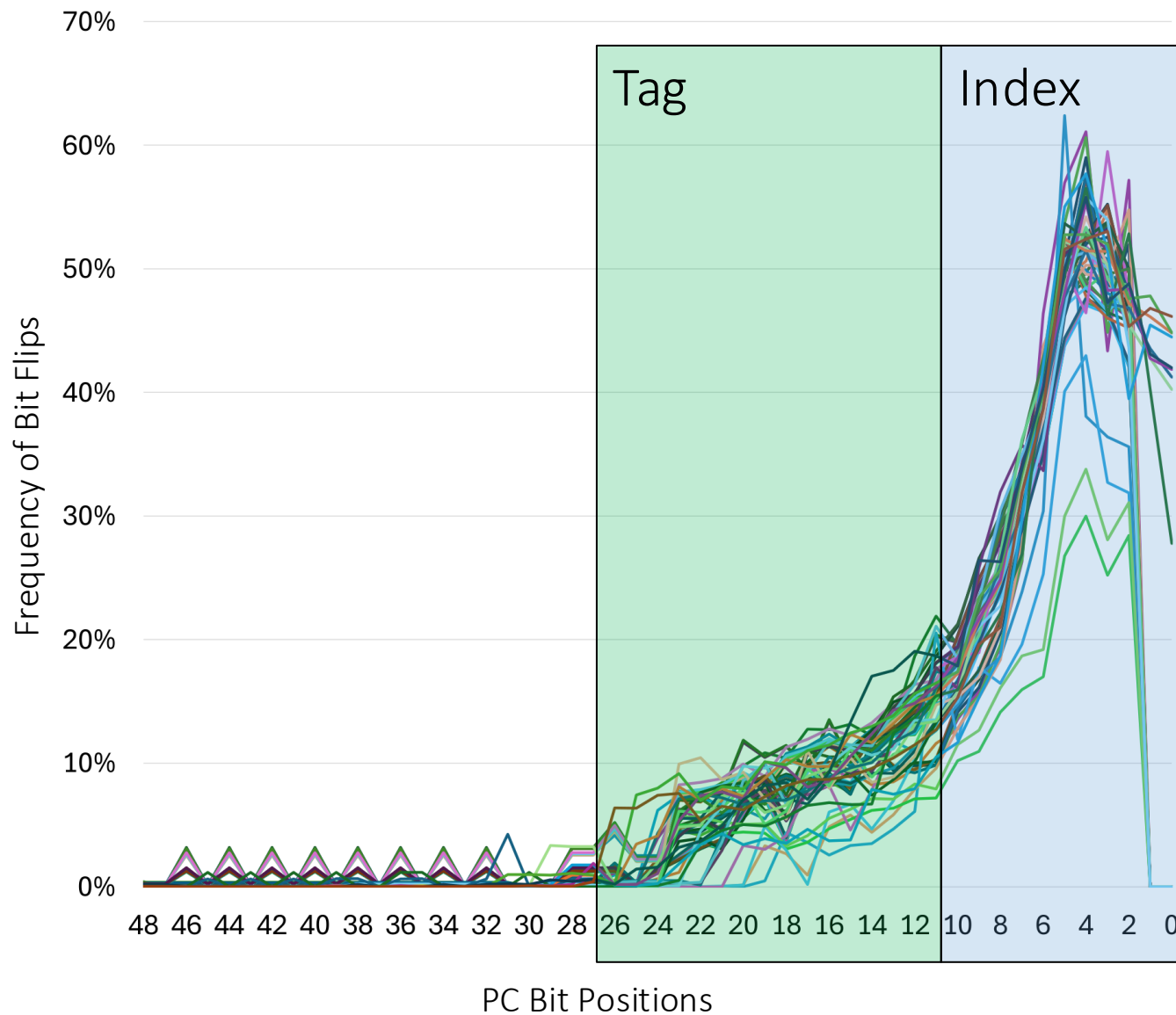
How to size the BTB Tag



How to size the BTB Tag



How to size the BTB Tag



Implications of truncated tag

Truncating higher order bits results in false hits



Implications of truncated tag

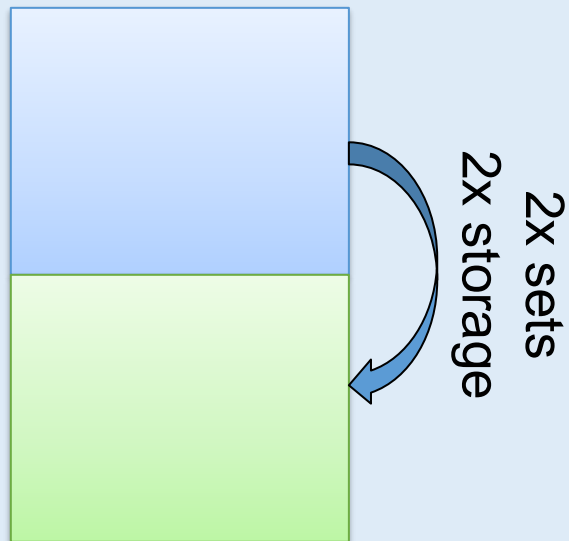
Truncating higher order bits results in false hits



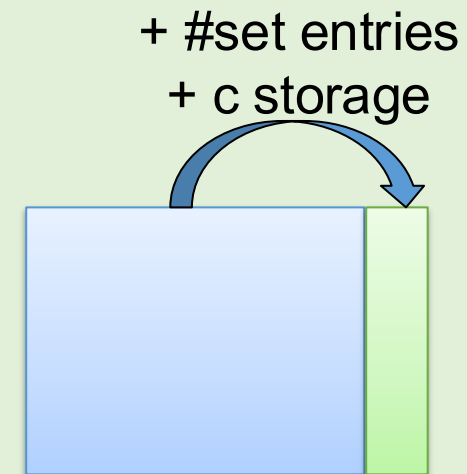
Goal

Understanding impact of tag sizes on performance across different BTB capacities and configurations

Set Scaling



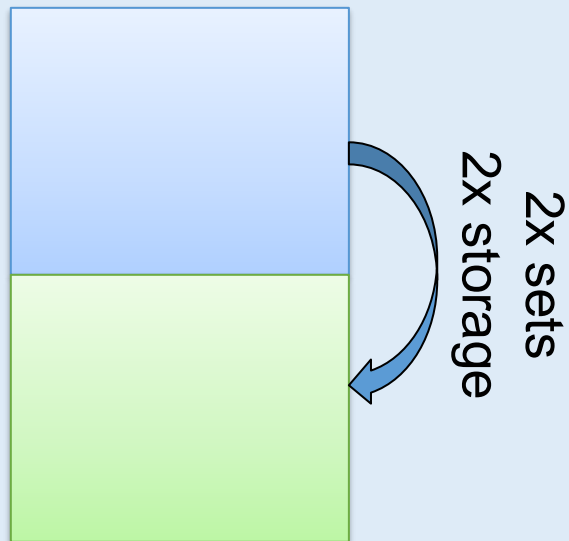
Way Scaling



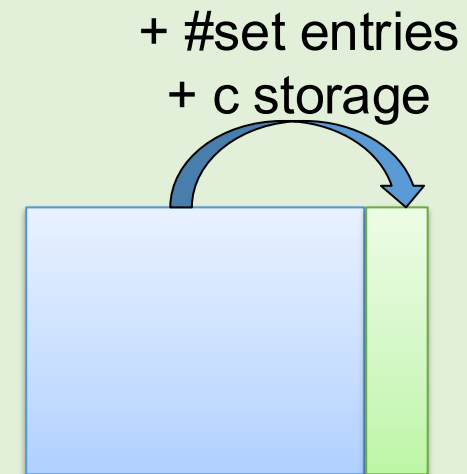
Goal

Understanding impact of tag sizes on performance across different BTB capacities and configurations

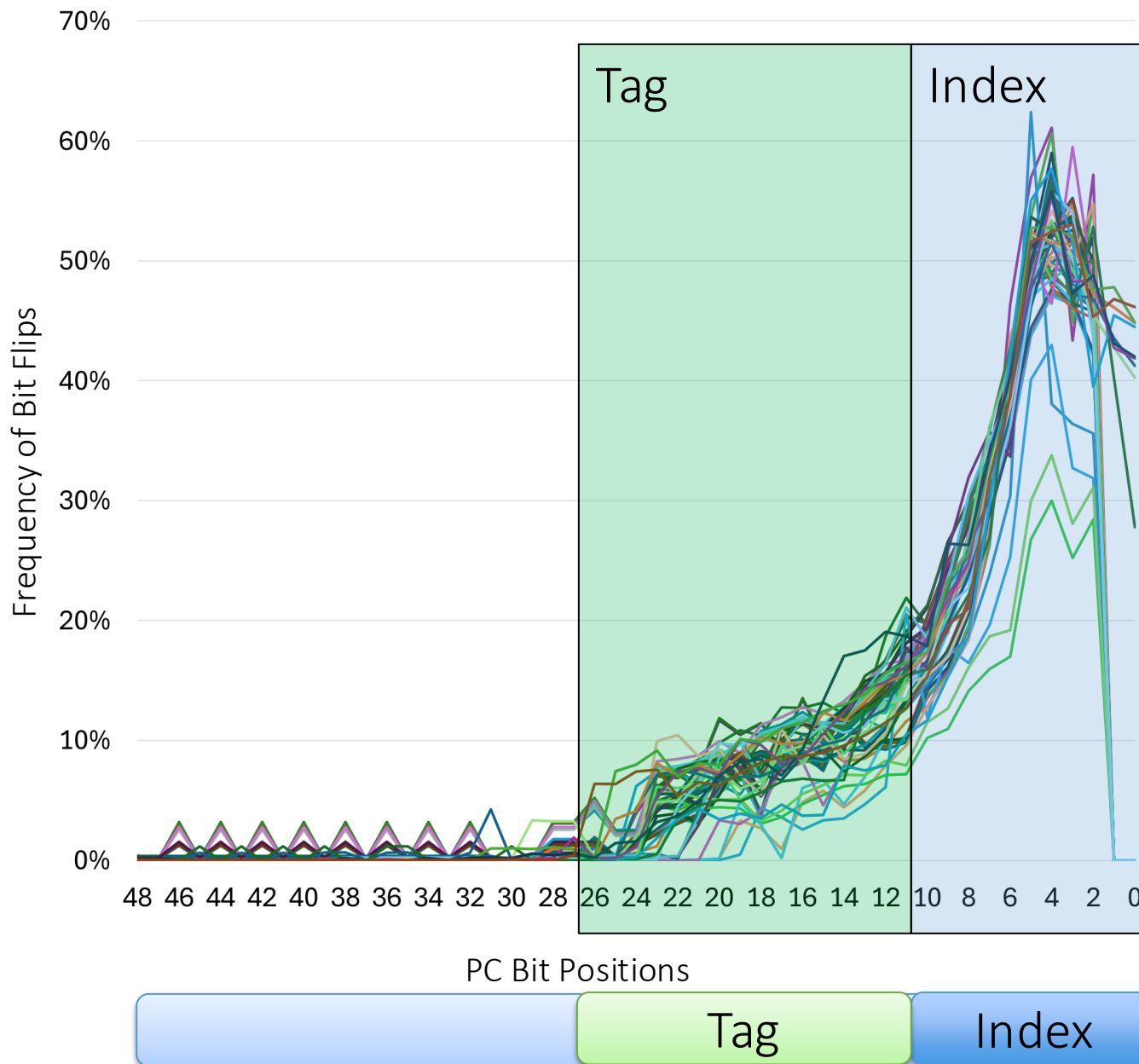
Set Scaling



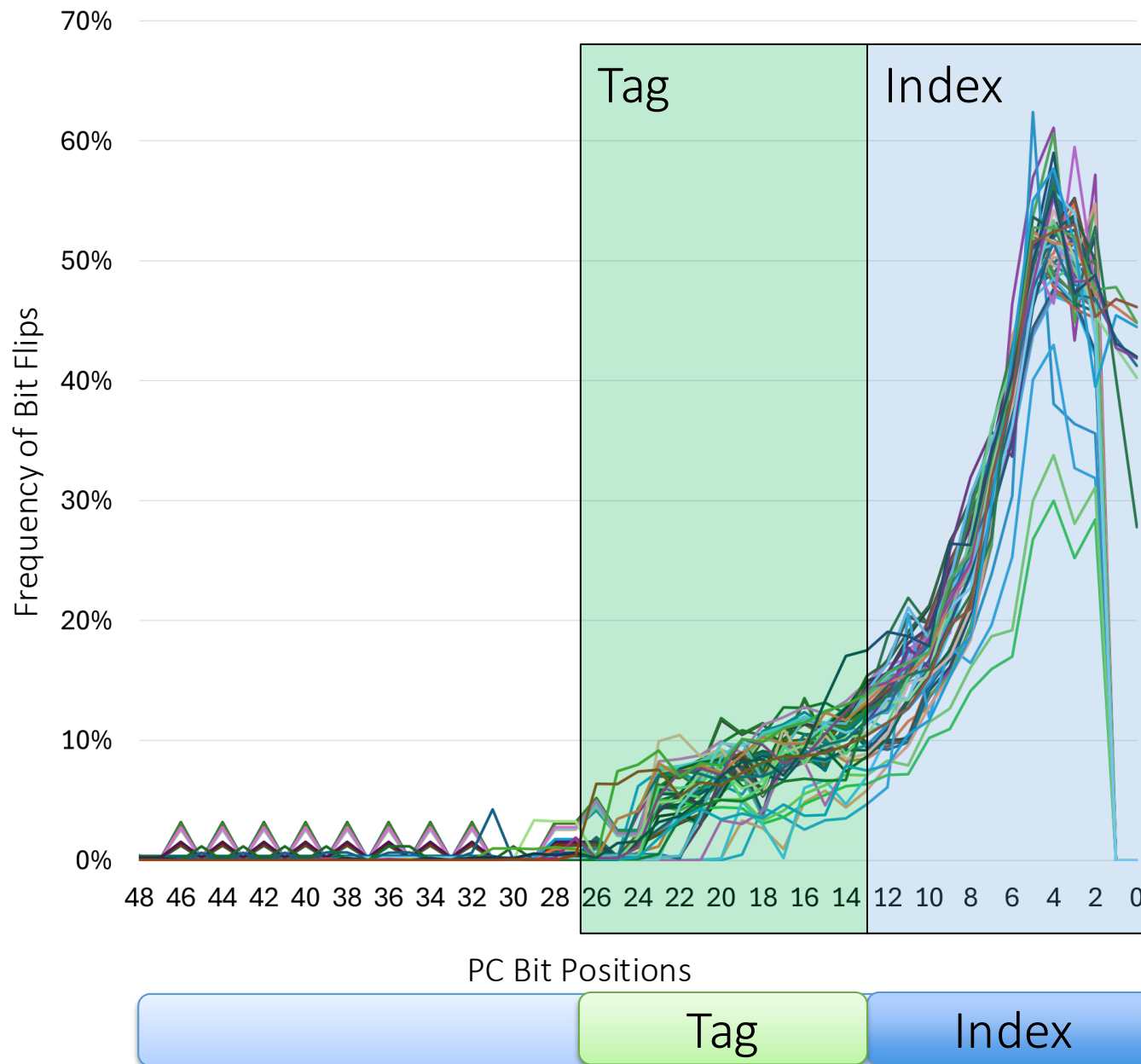
Way Scaling



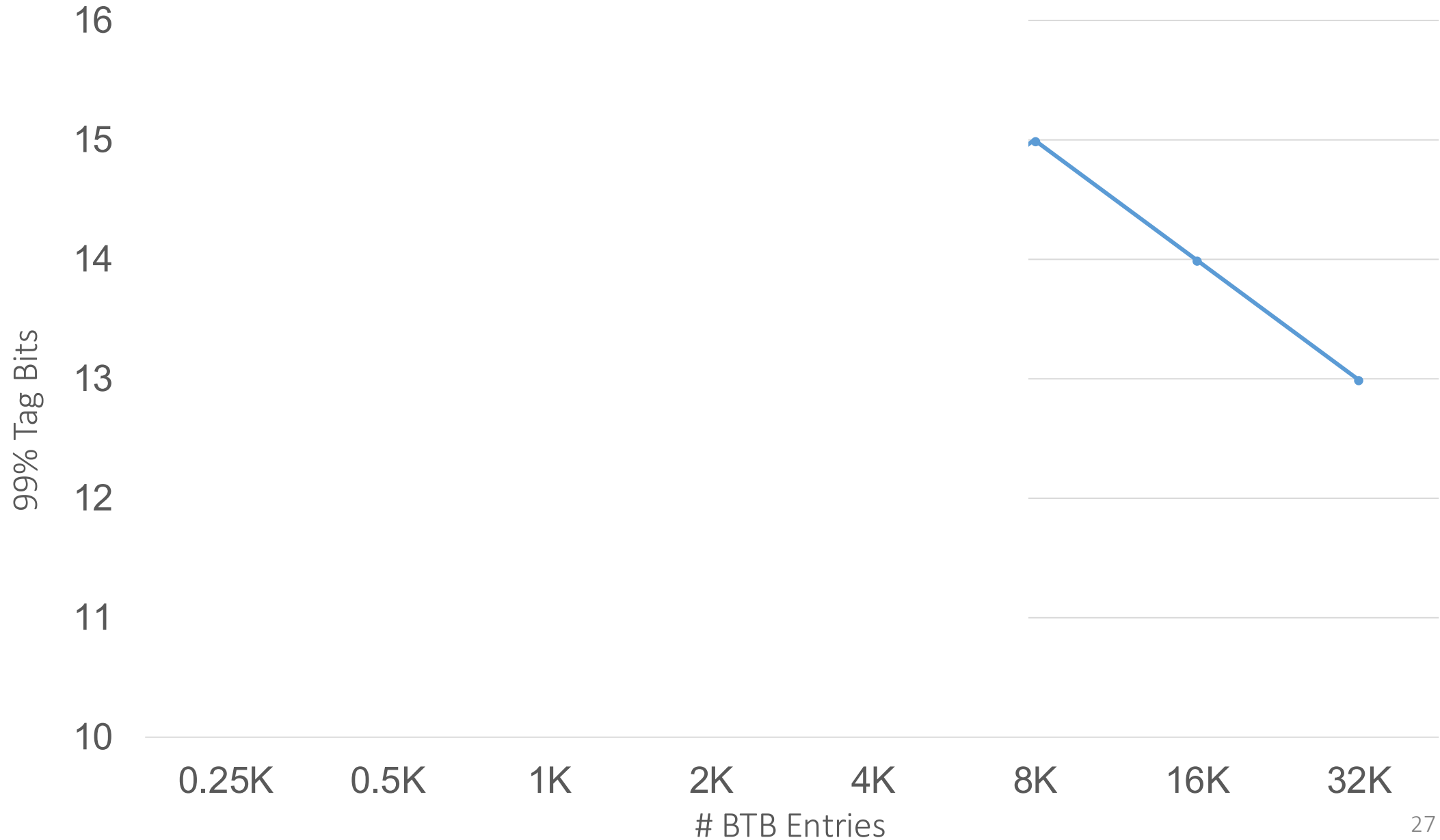
Impact of Scaling Sets on Index



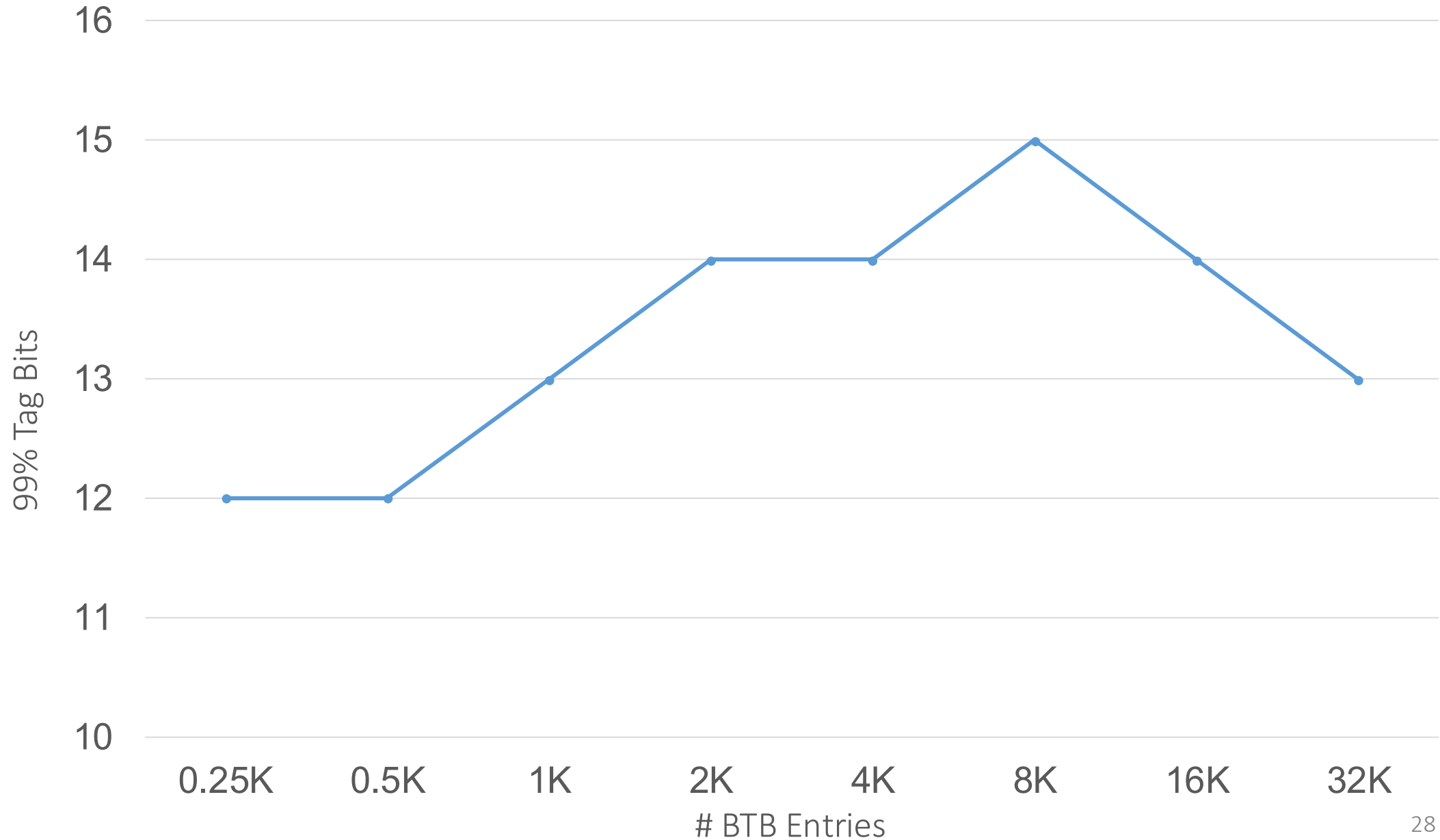
Impact of Scaling Sets on Index



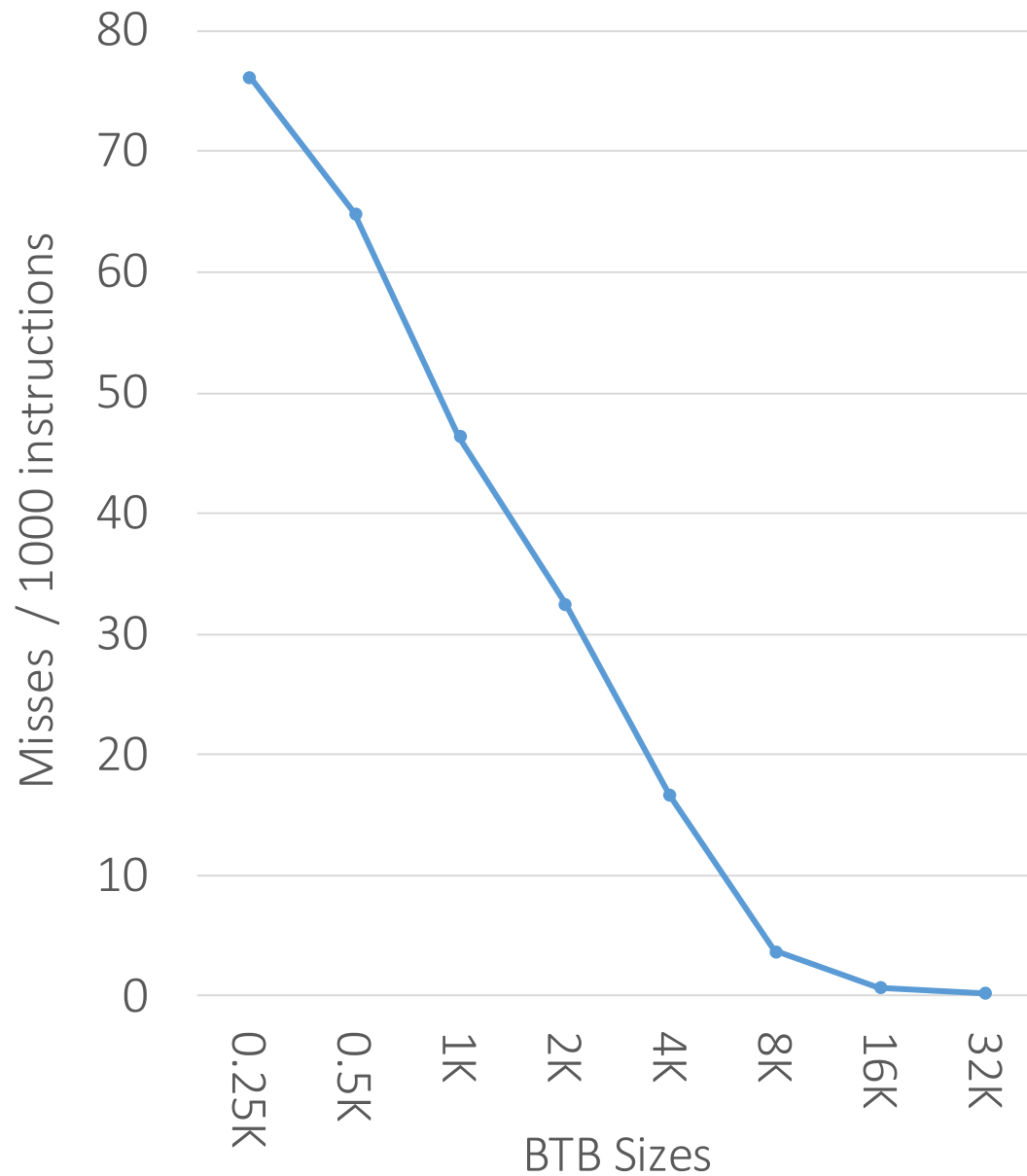
Tag Size Requirements



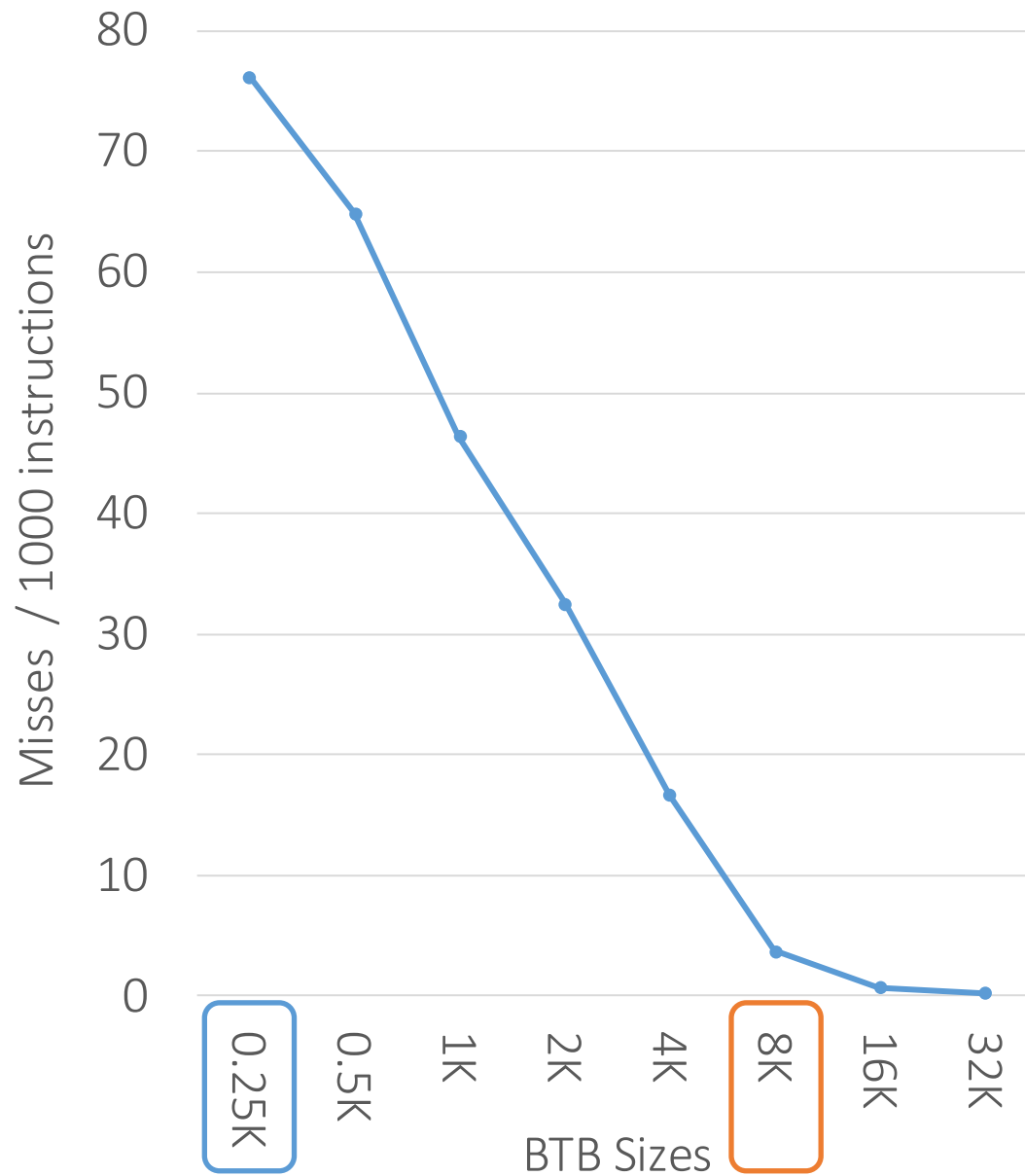
Tag Size Requirements



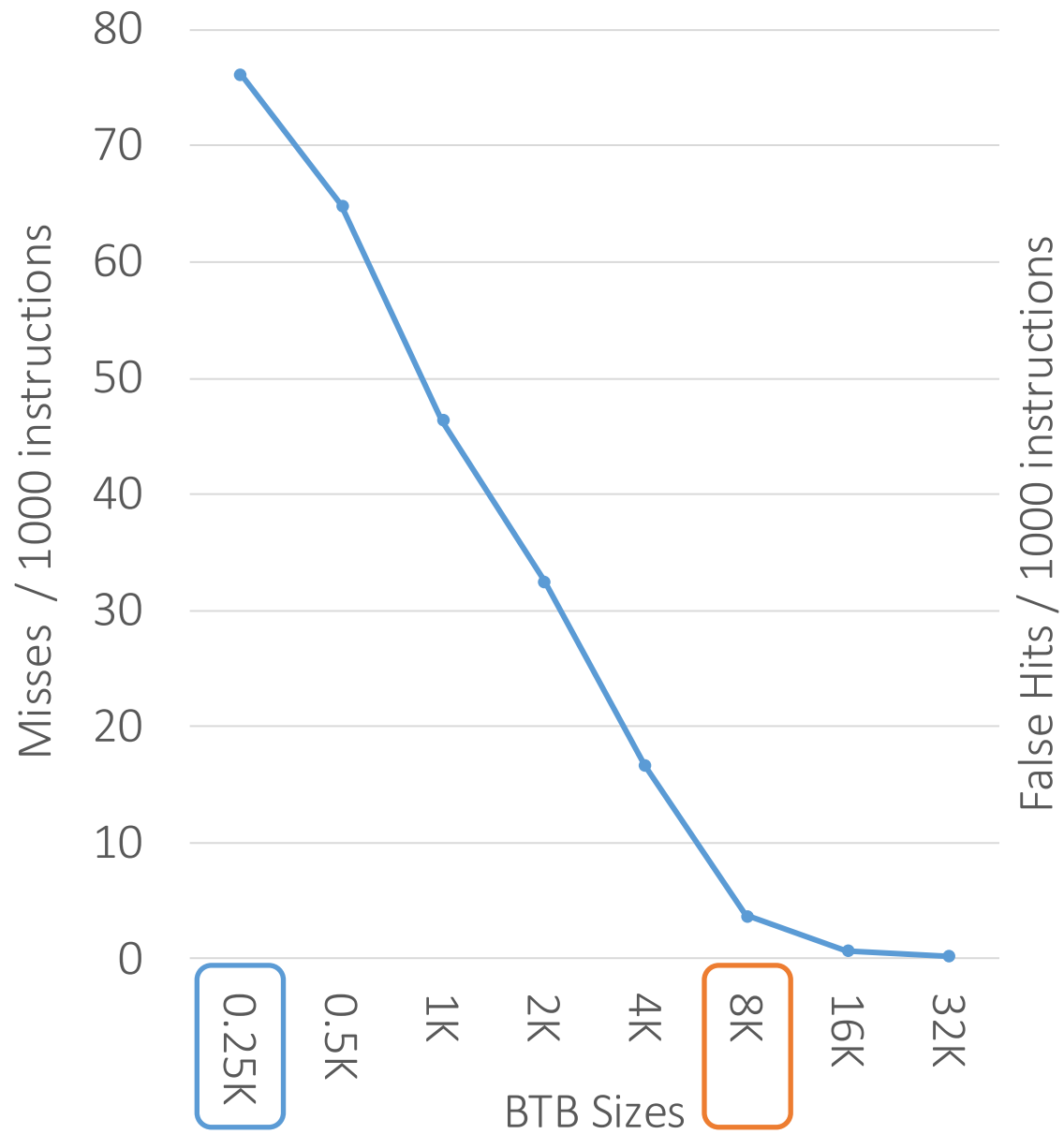
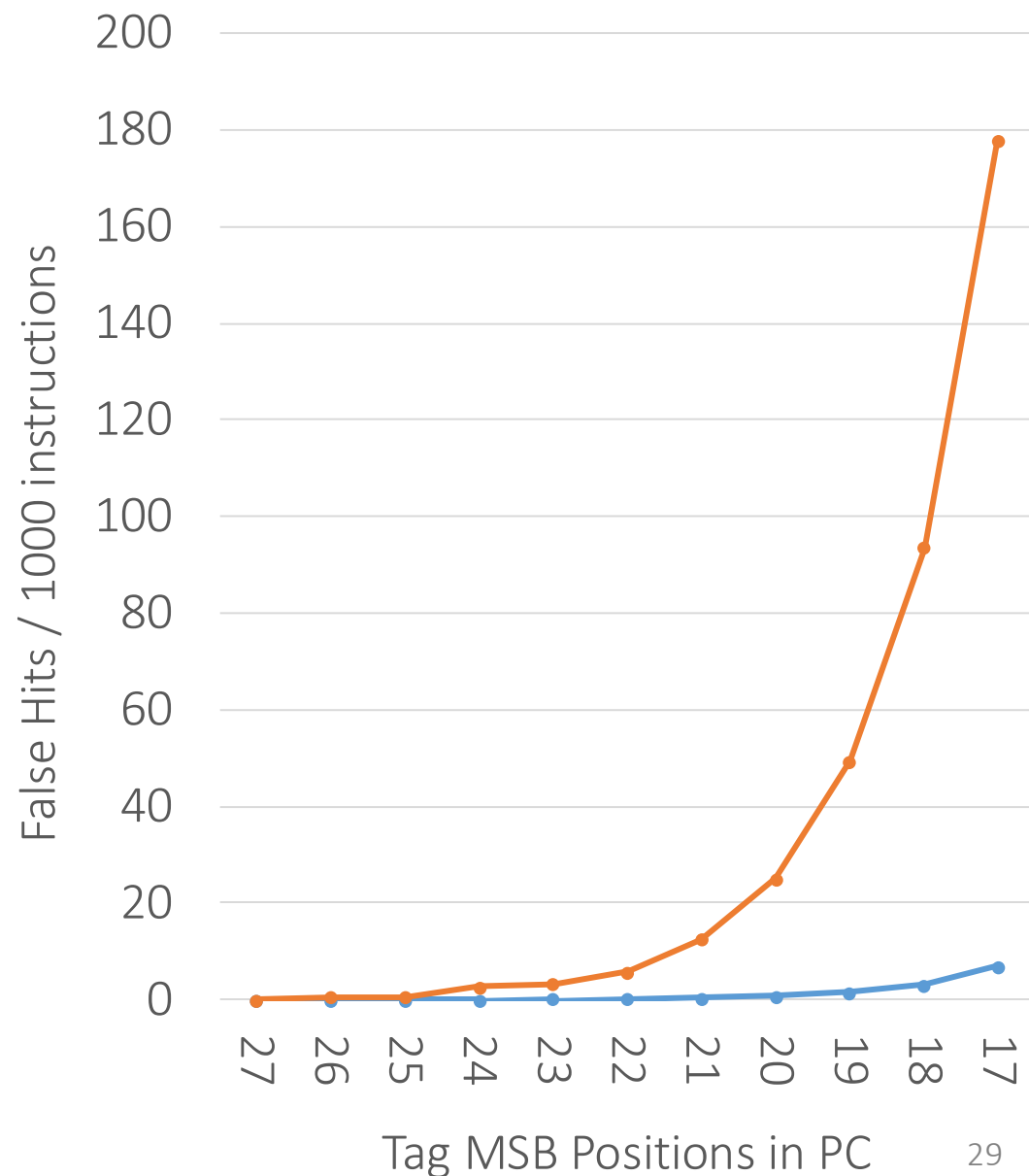
Tag Size Requirements



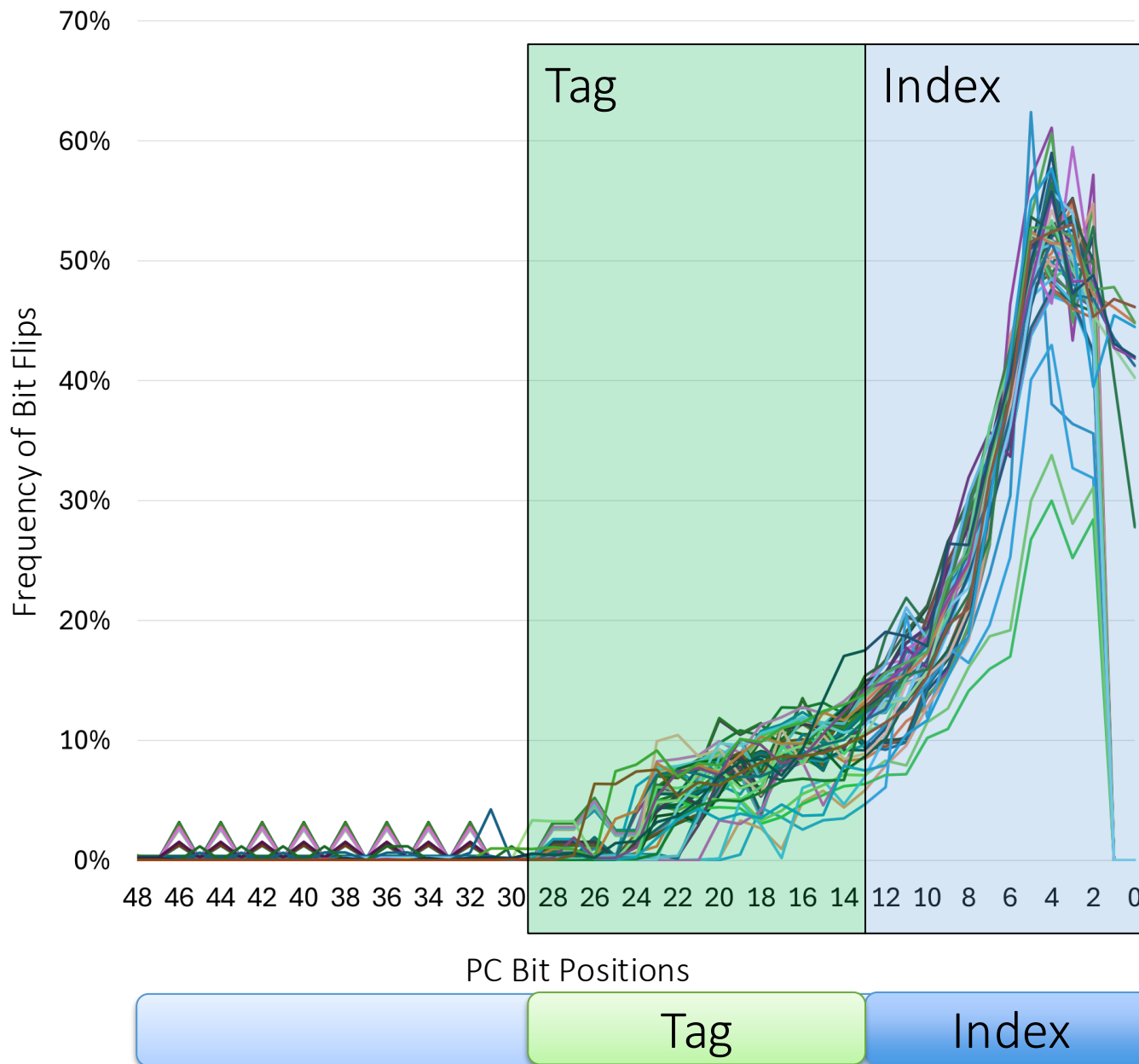
Tag Size Requirements



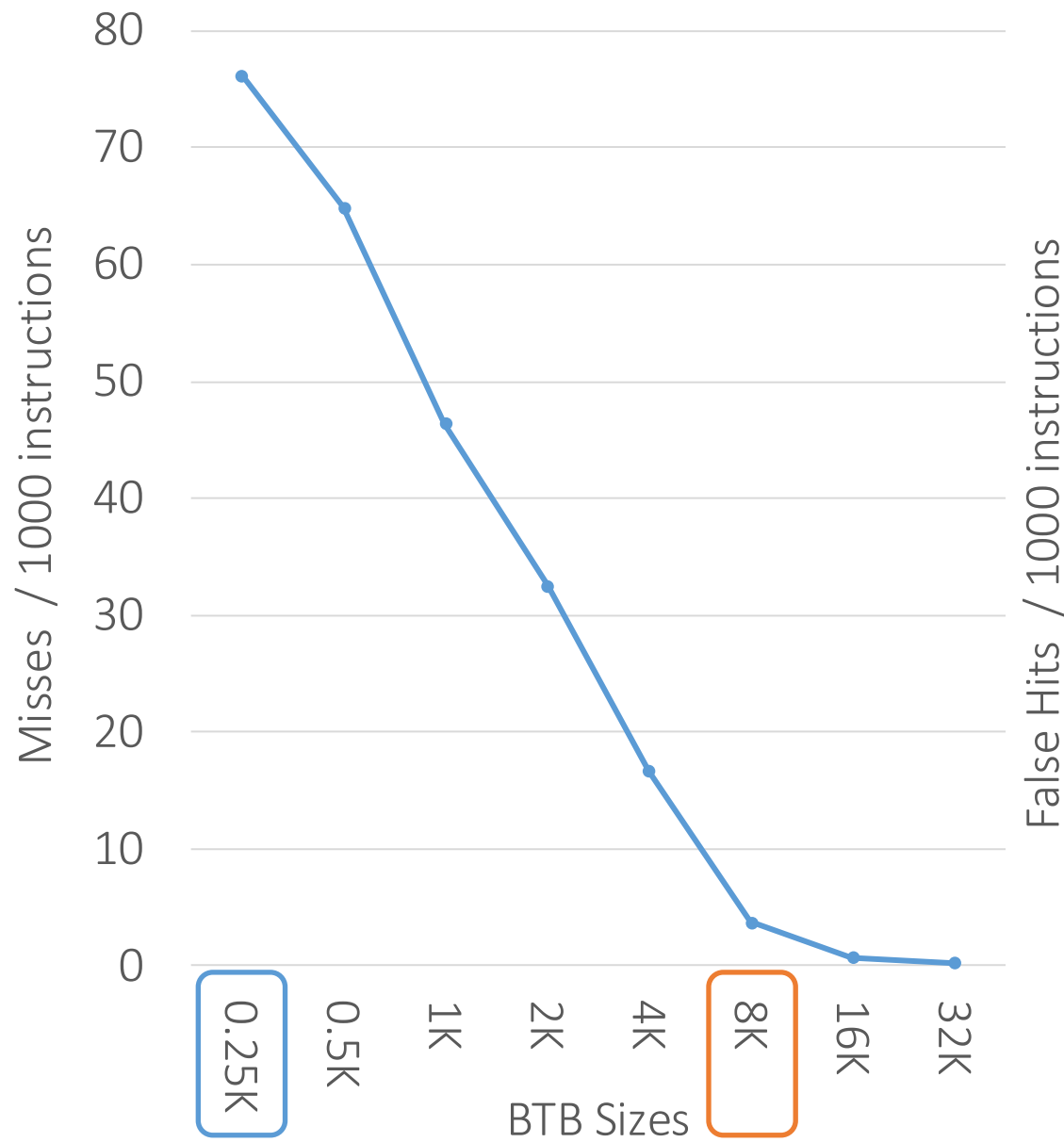
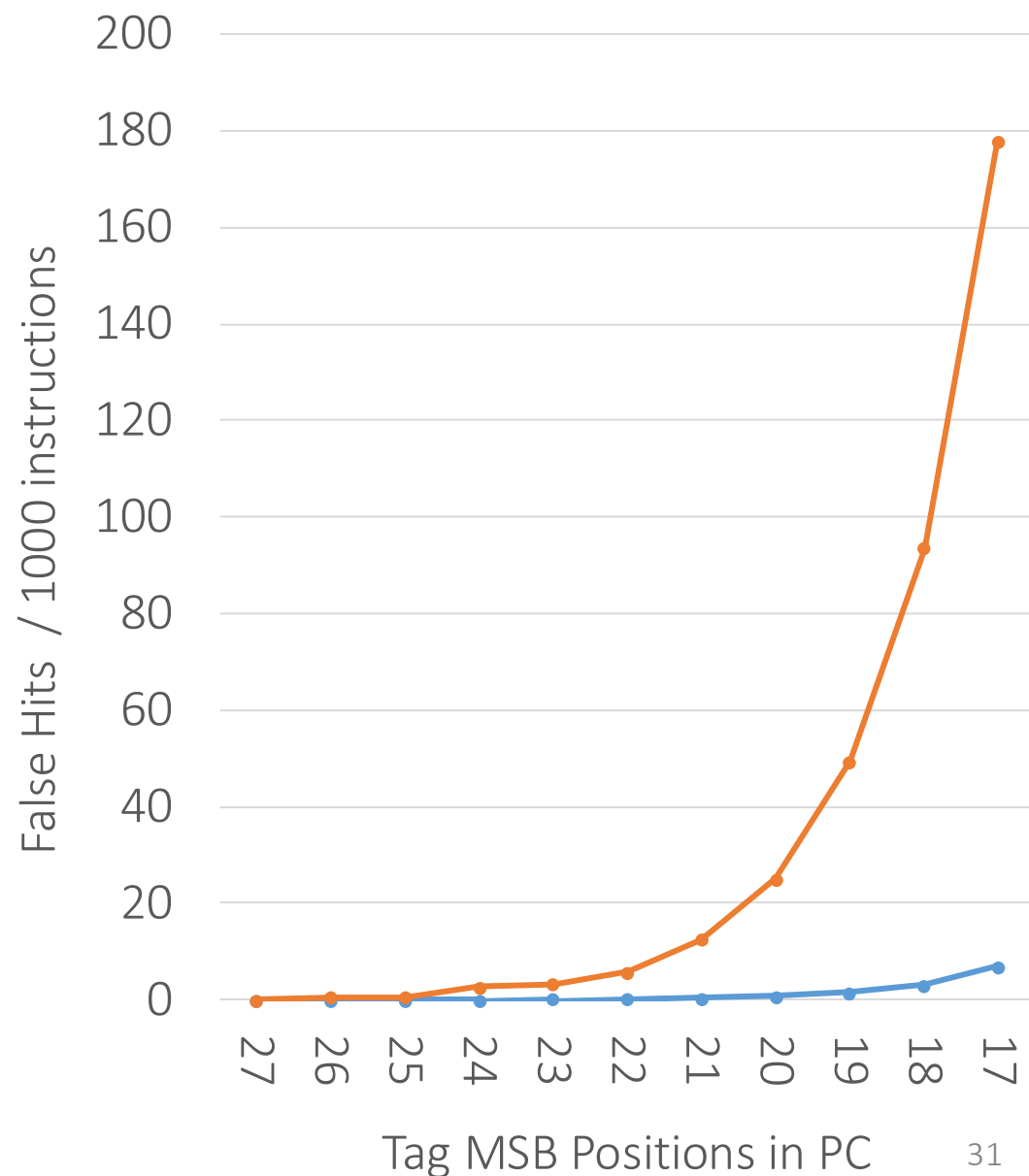
Tag Size Requirements



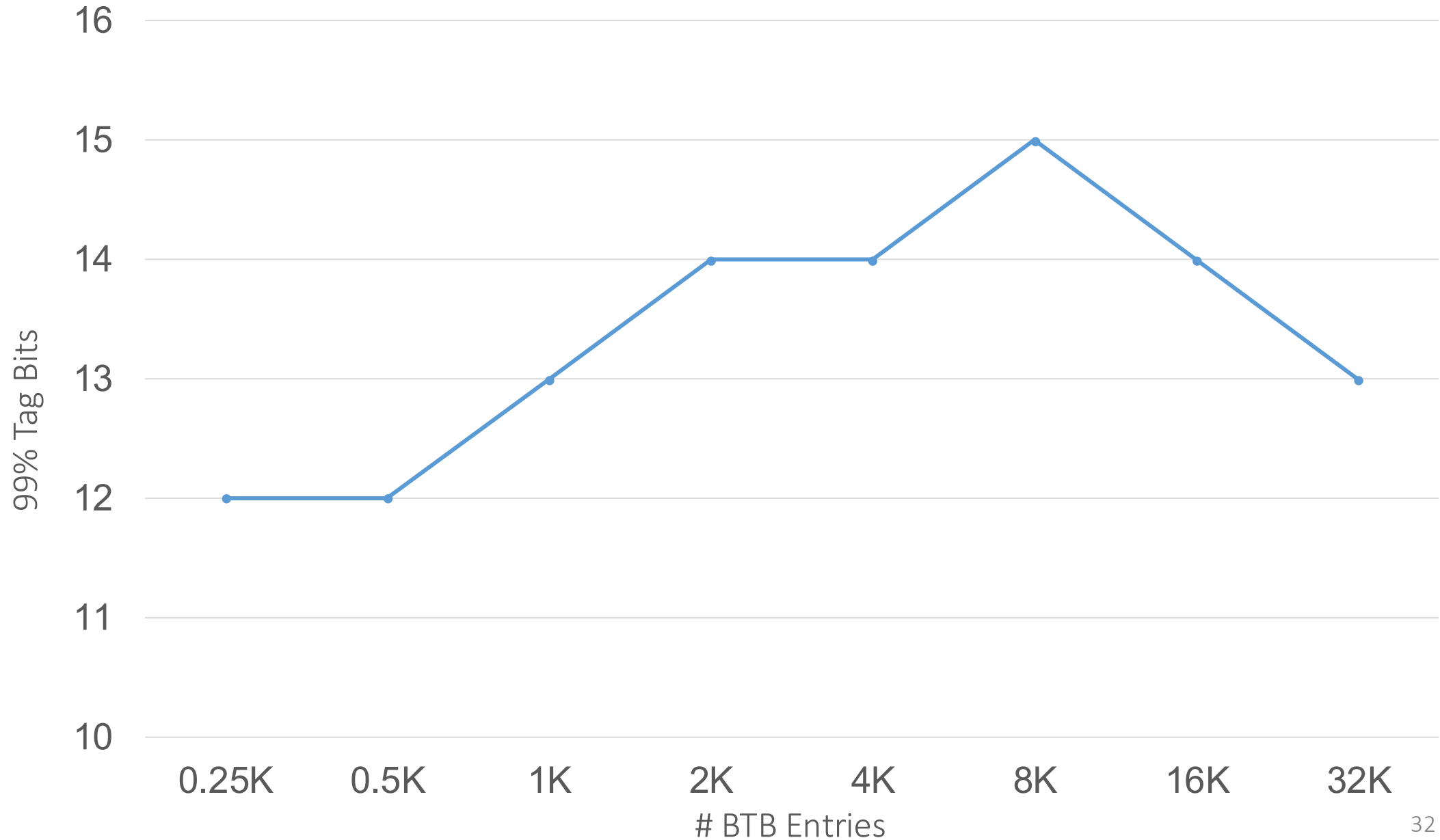
Impact of Scaling Sets on Index



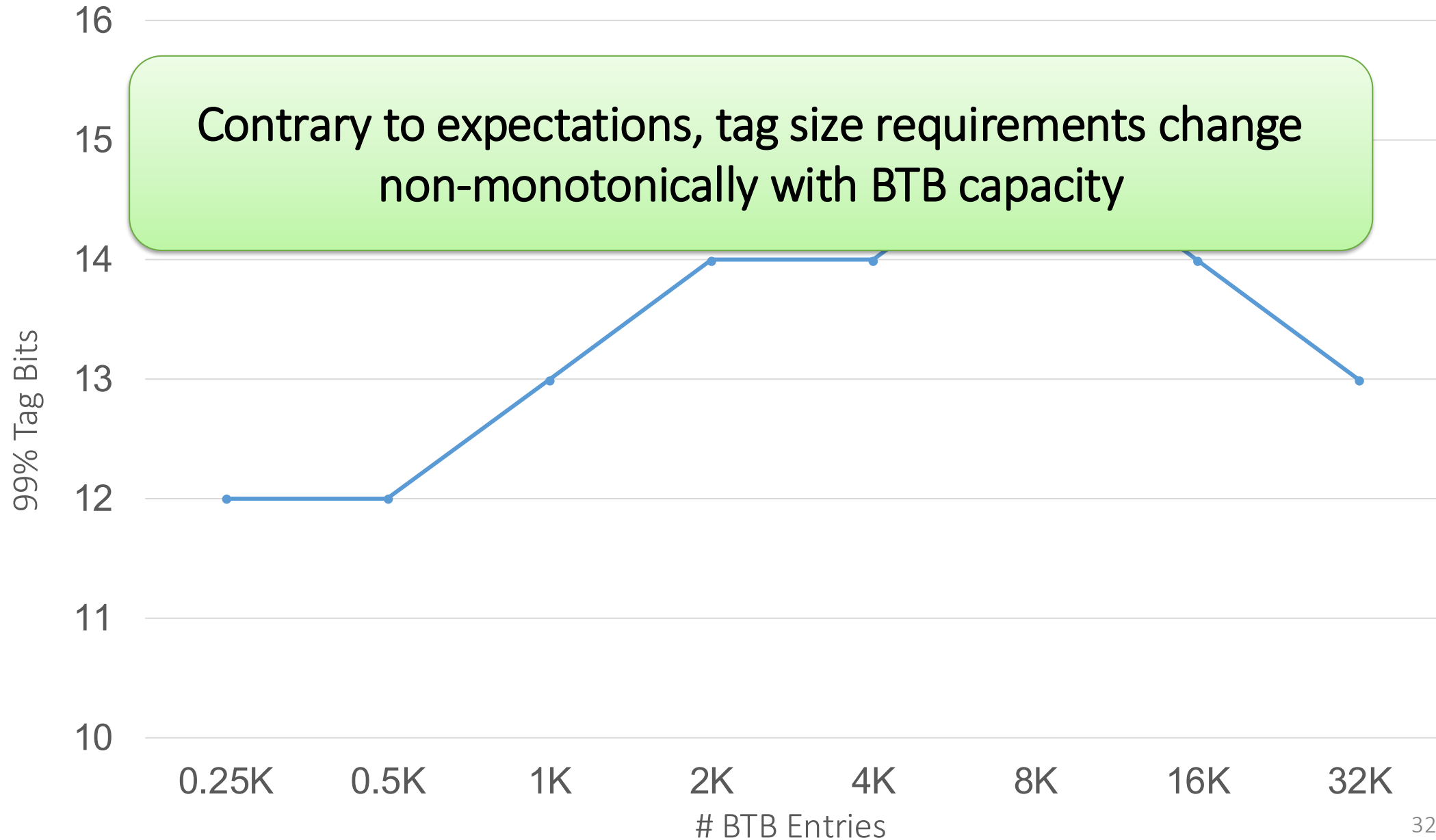
Tag Size Requirements



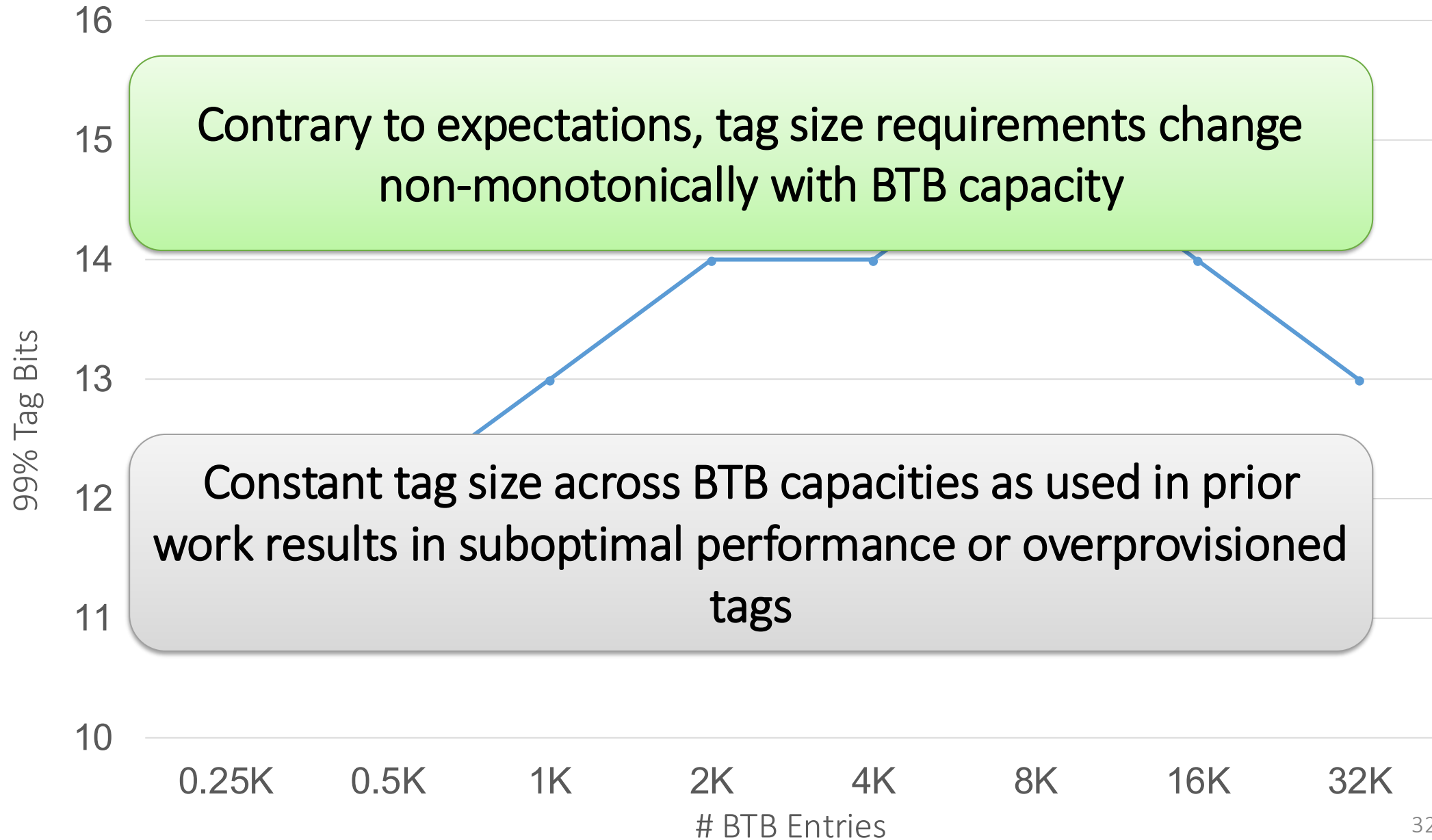
Tag Size Requirements



Tag Size Requirements



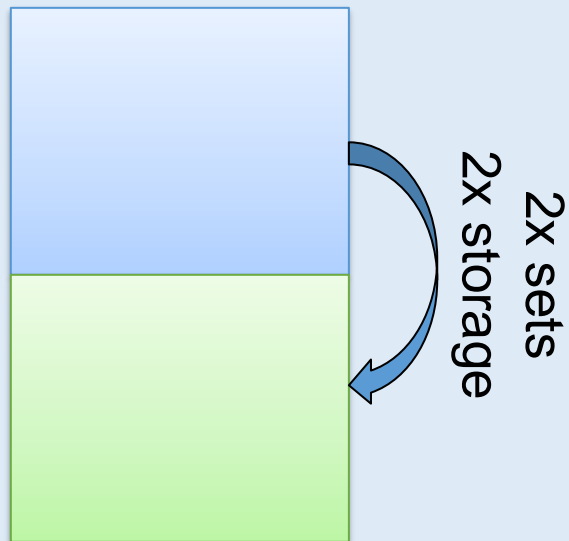
Tag Size Requirements



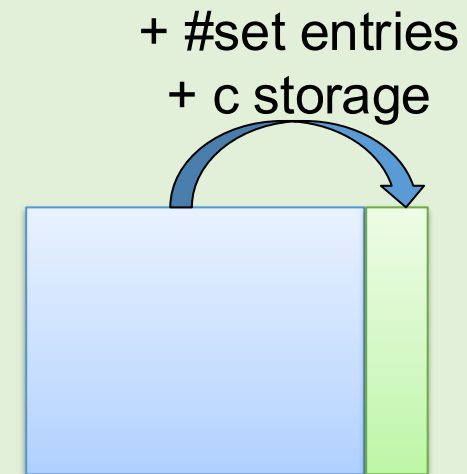
Goal

Understanding impact of tag sizes on performance across different BTB capacities and configurations

Set Scaling



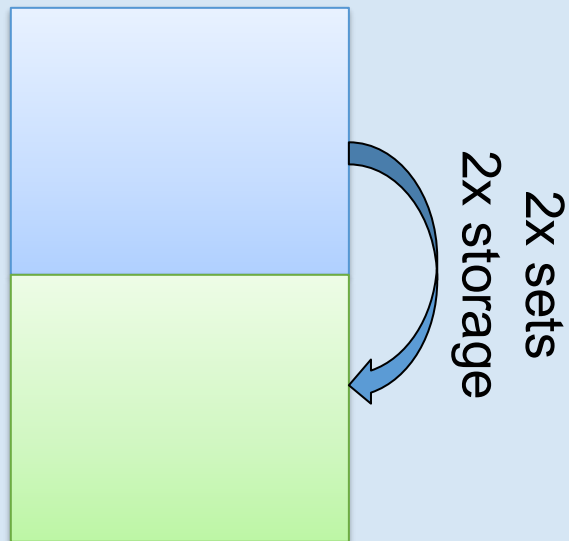
Way Scaling



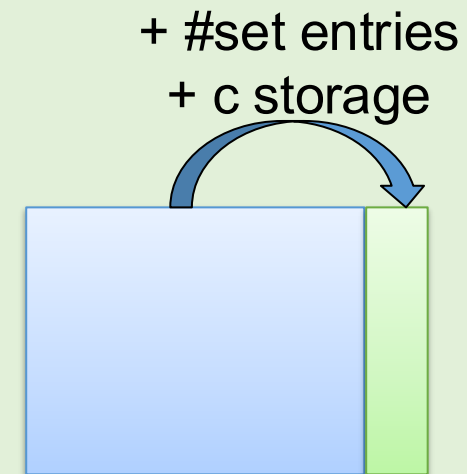
Goal

Understanding impact of tag sizes on performance across different BTB capacities and configurations

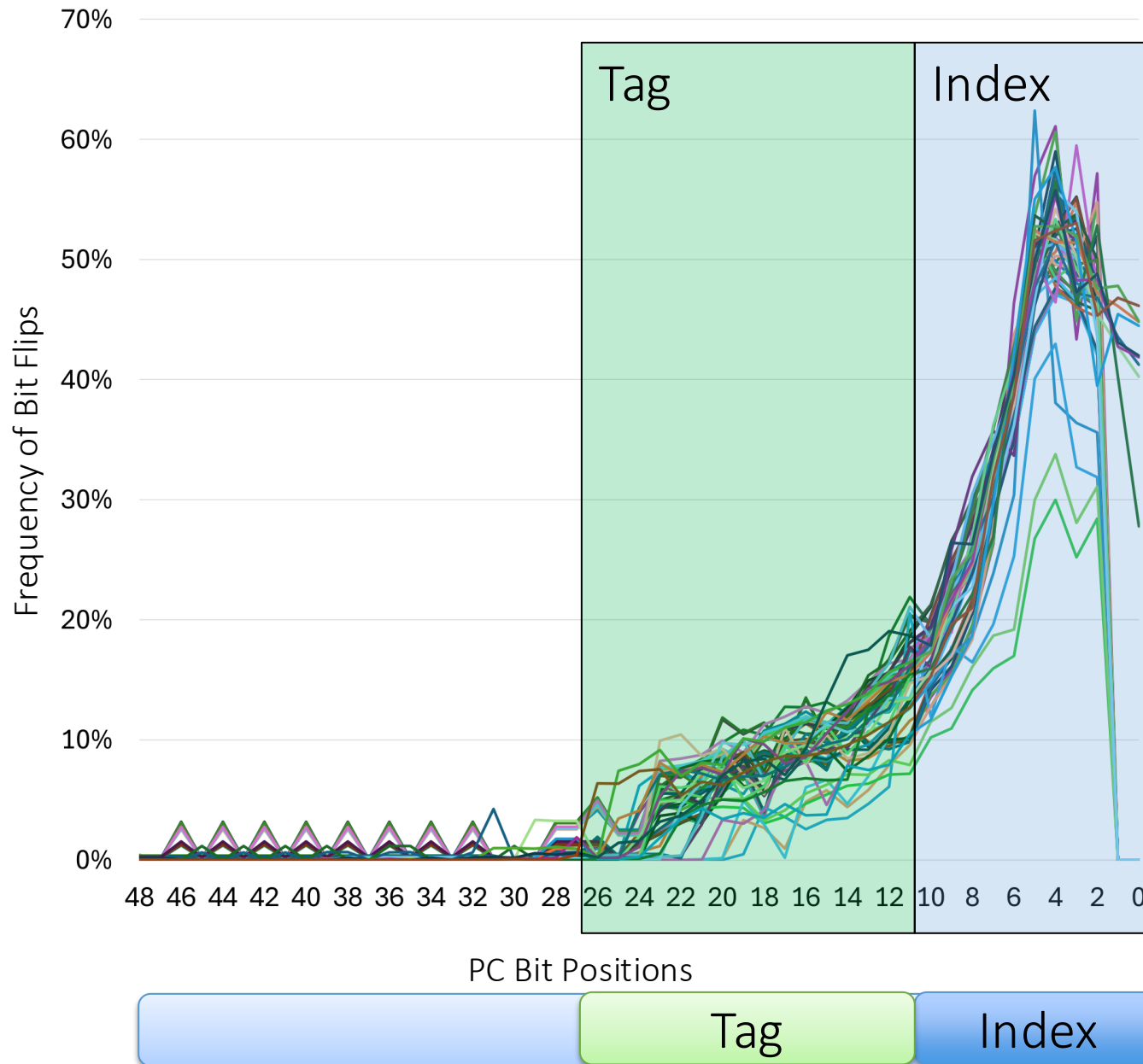
Set Scaling



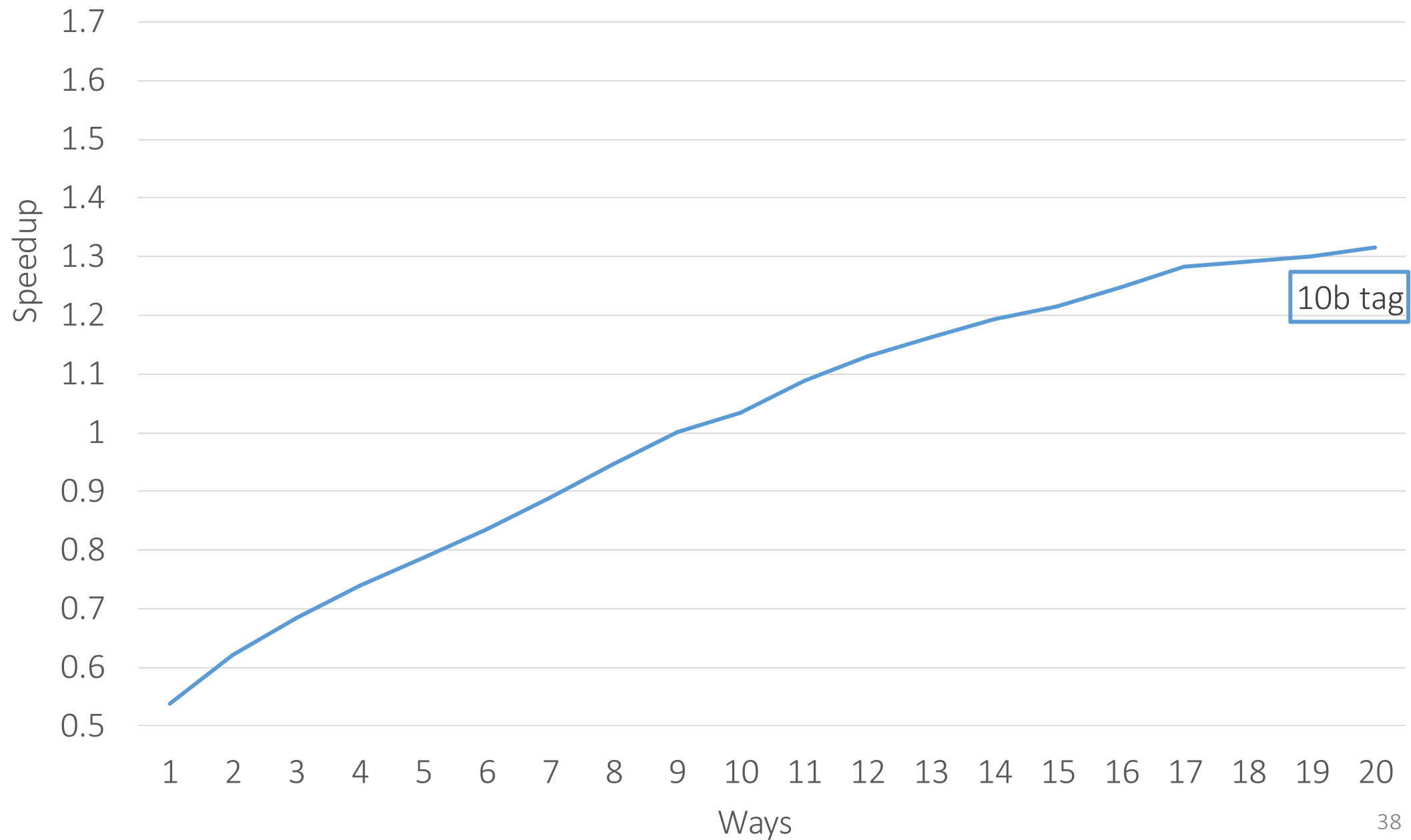
Way Scaling



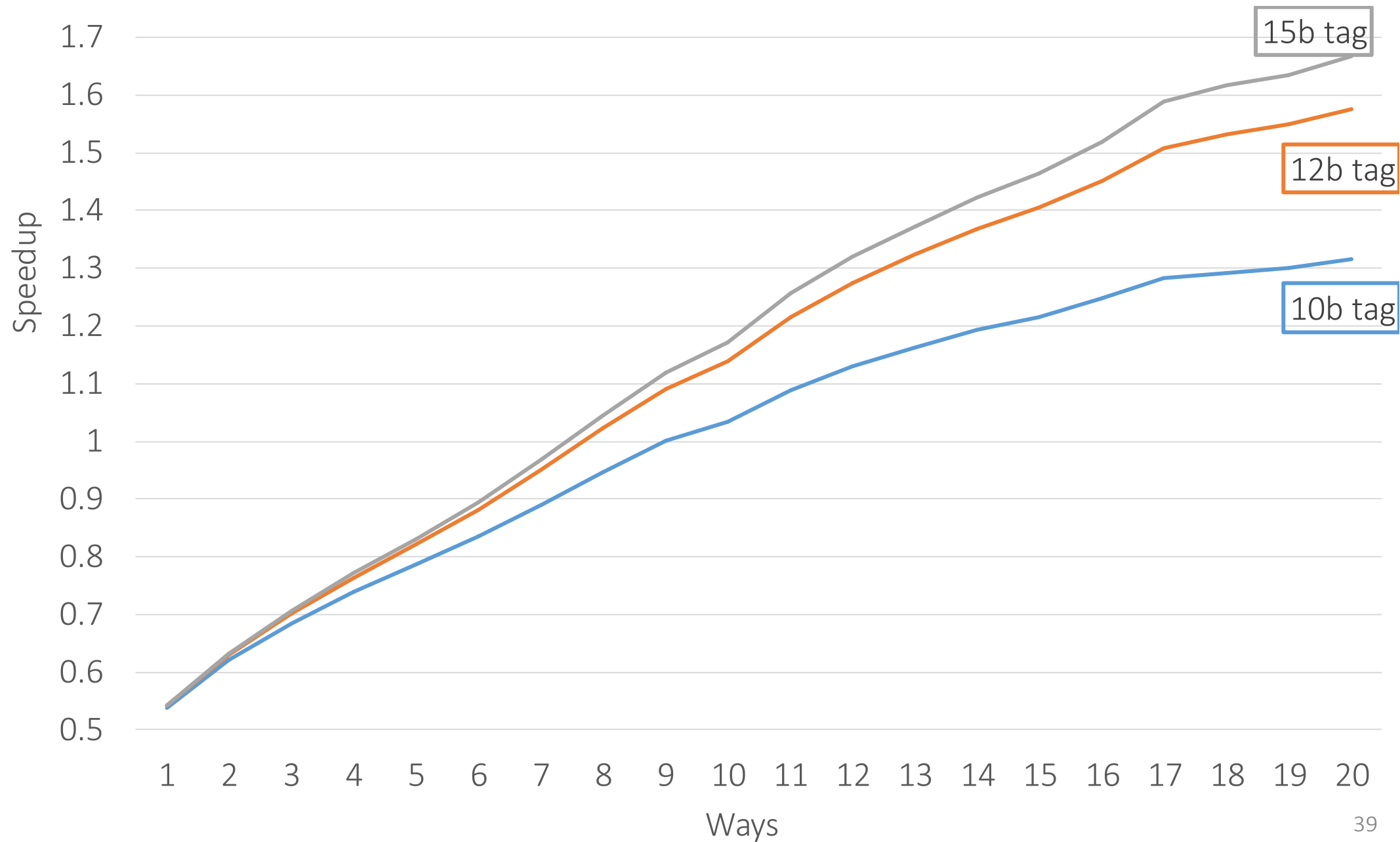
Impact of Way Scaling



Performance with way scaling



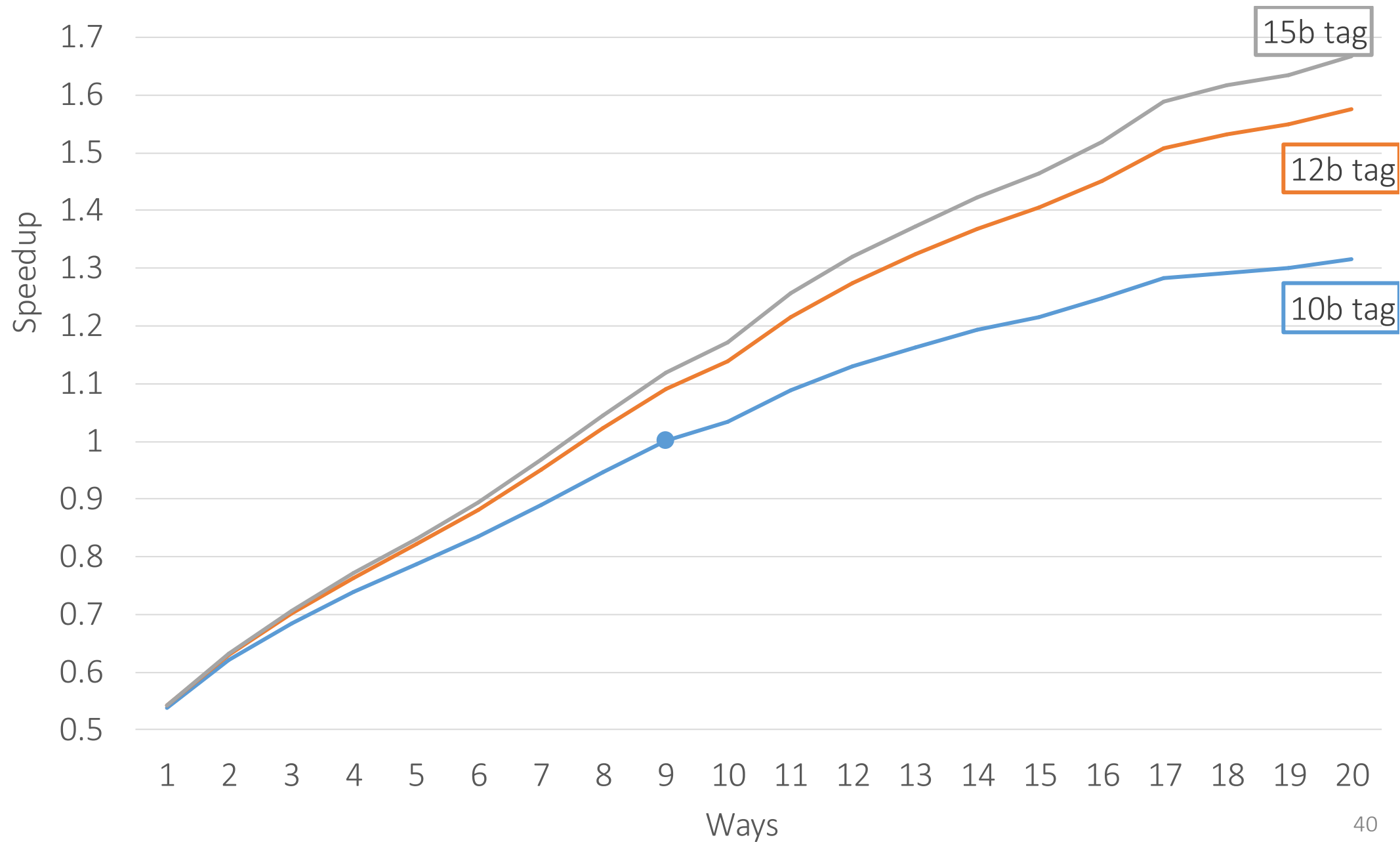
Performance with way scaling



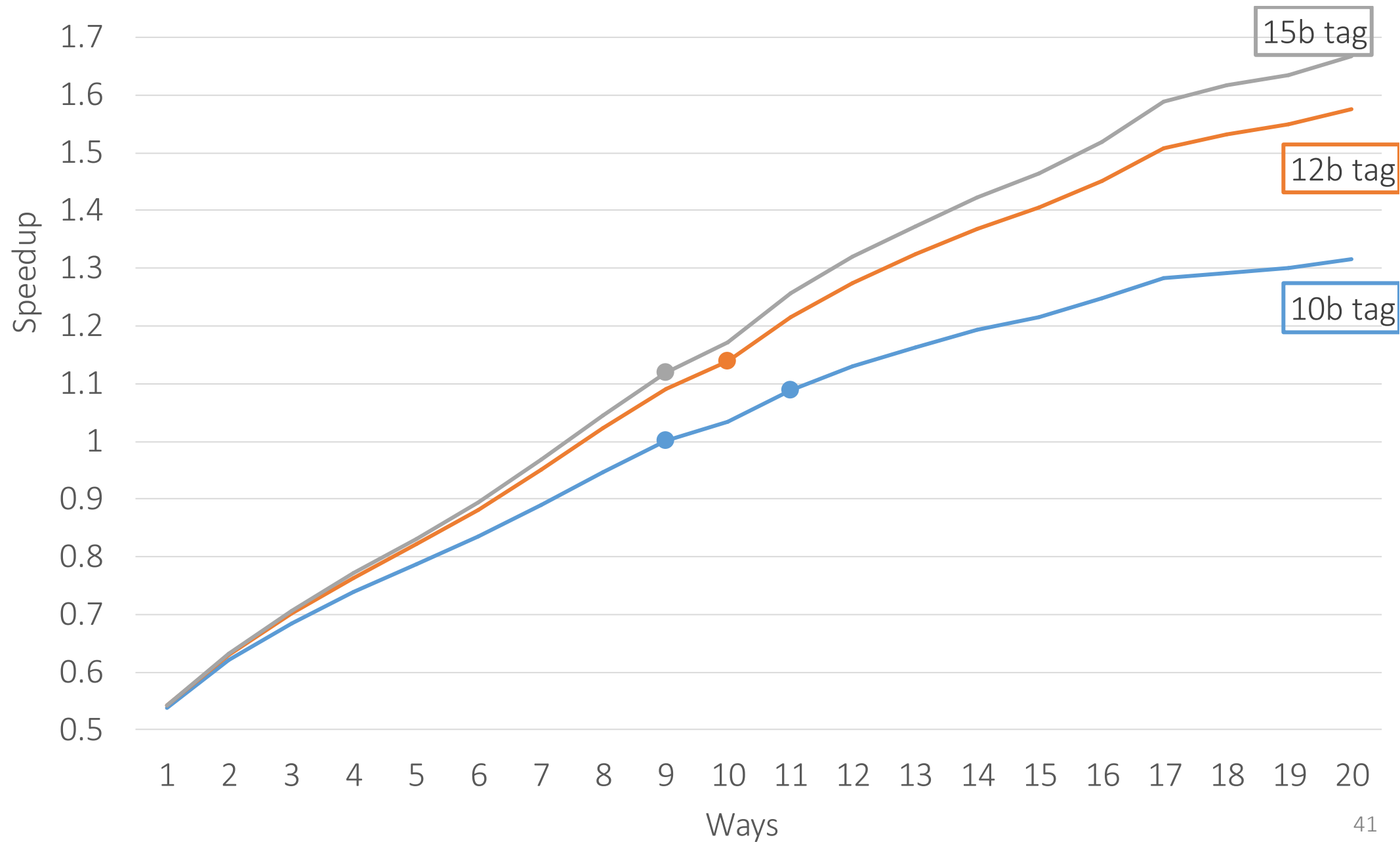
Performance with way scaling



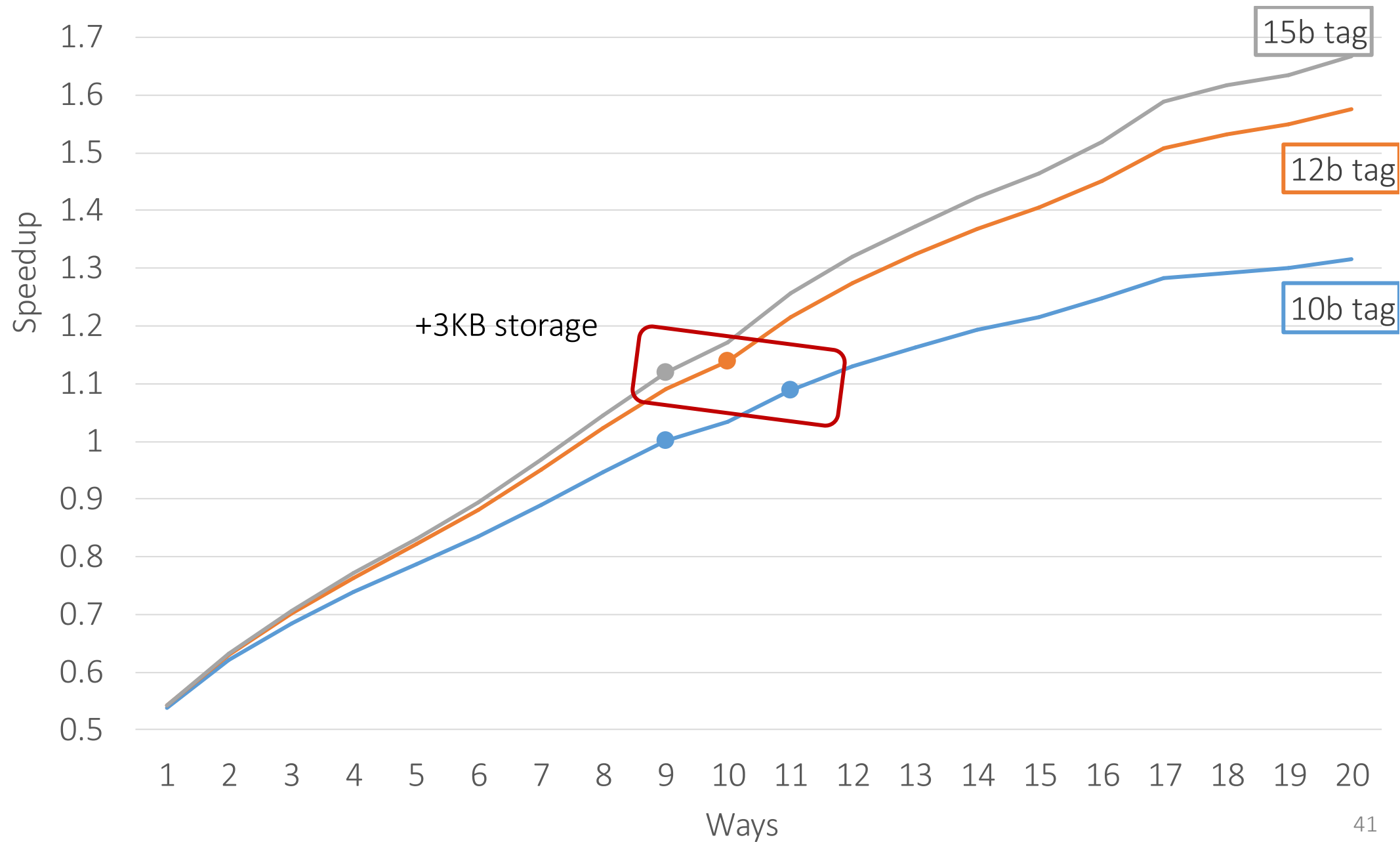
Performance with way scaling



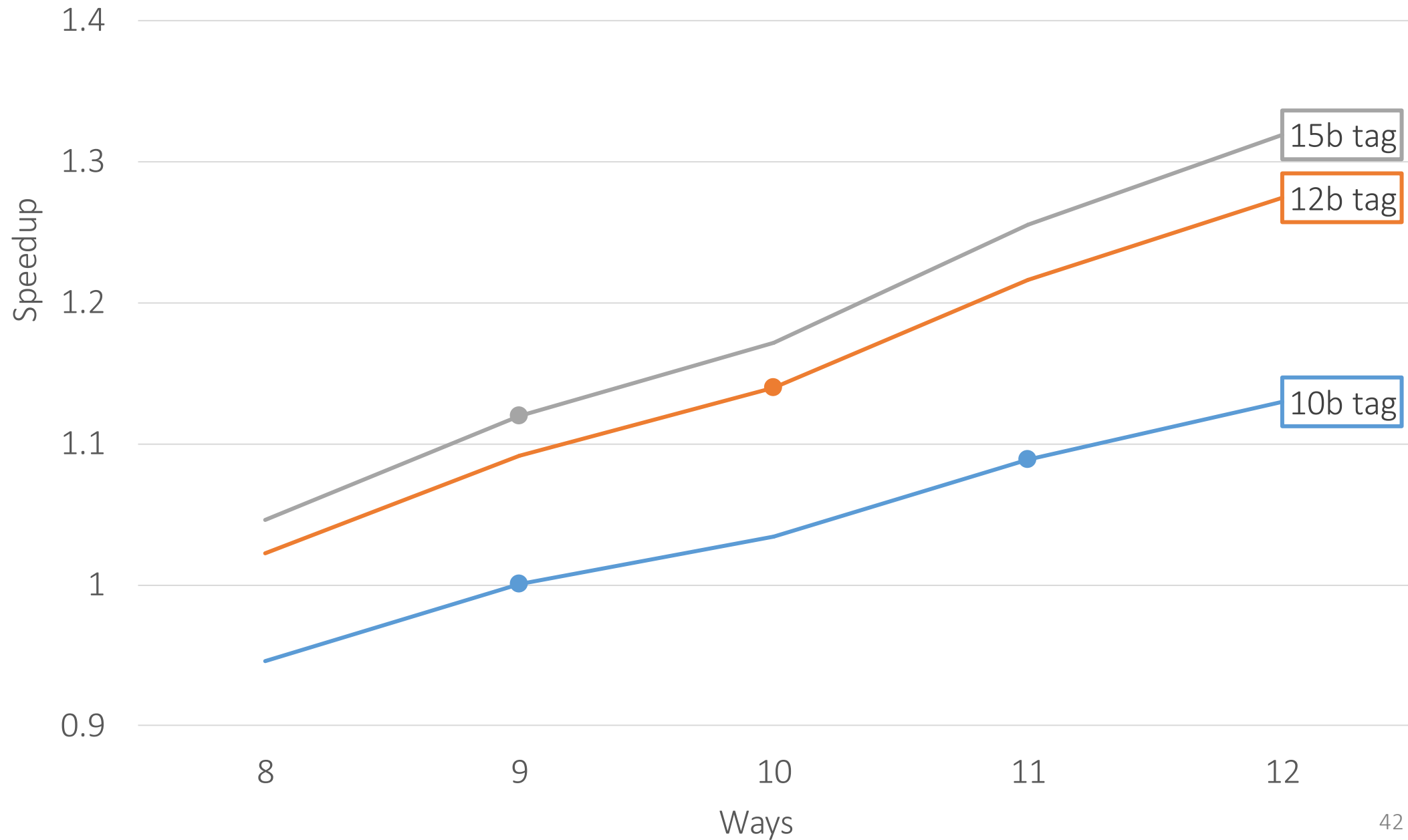
Performance with way scaling



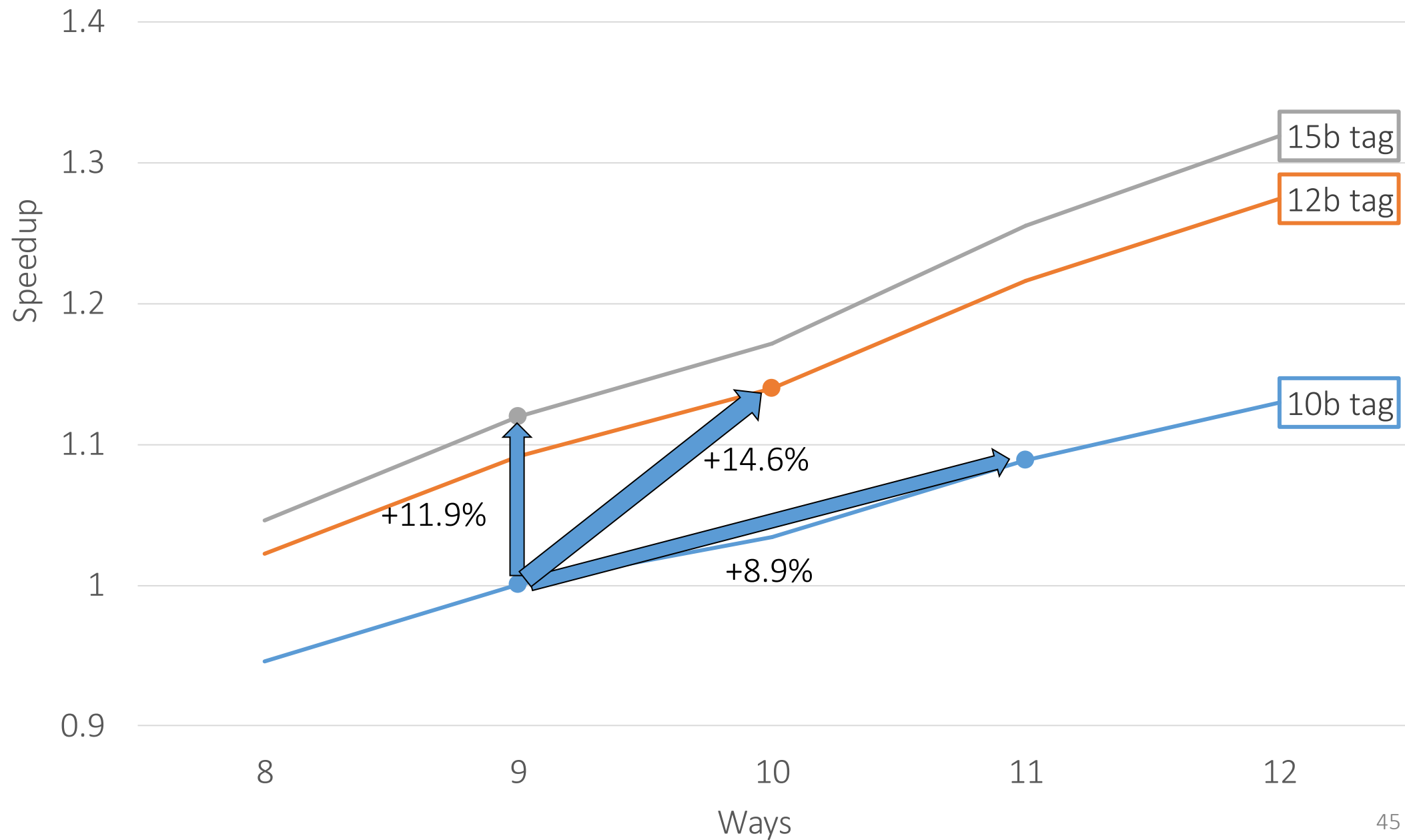
Performance with way scaling



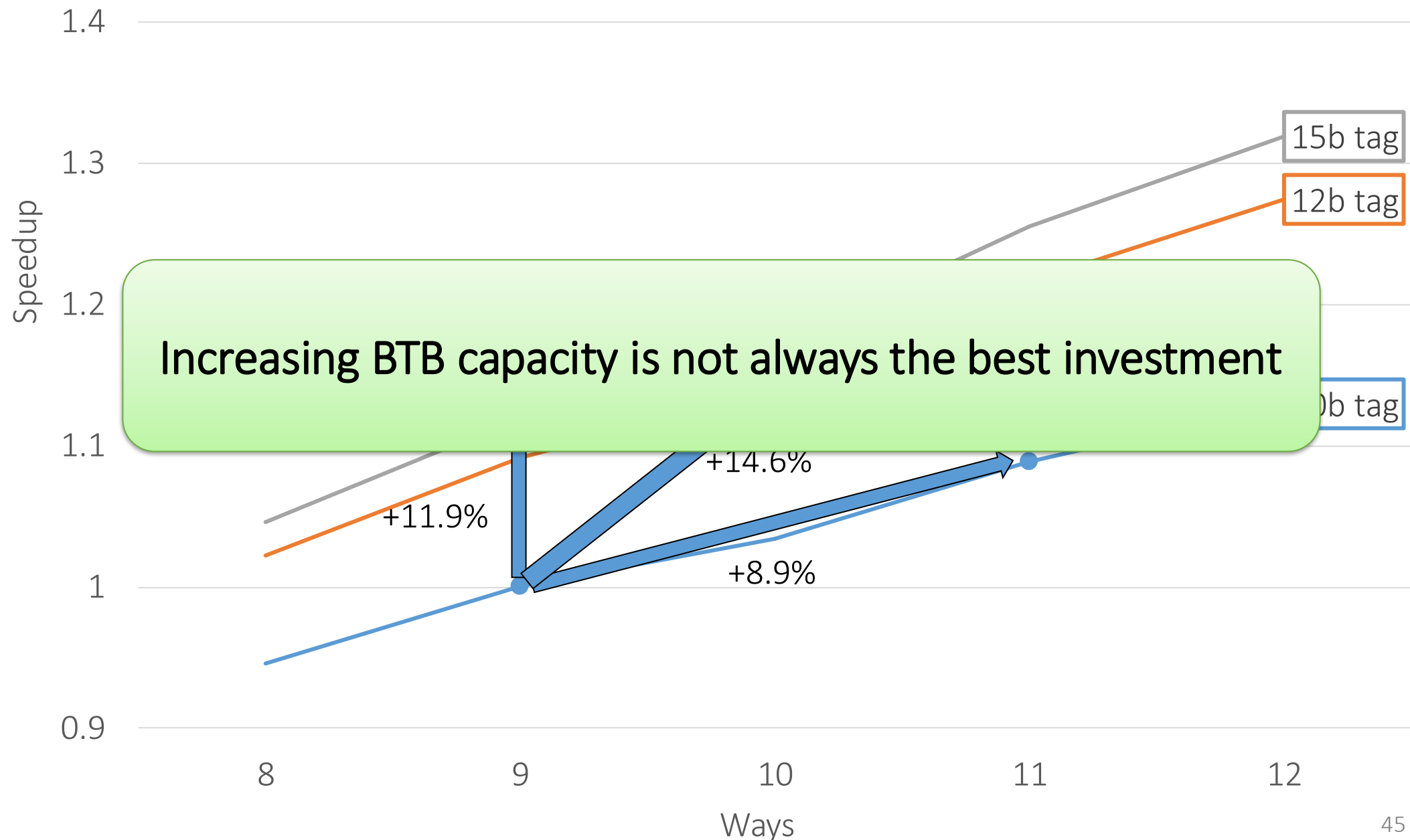
Performance with way scaling



Performance with way scaling



Performance with way scaling



Takeaways

- Tag size has a big impact on BTB storage and performance
- Tag size requirements change non-monotonically with BTB capacity
 - Constant tag size across BTB capacities, as used in prior work, results in suboptimal performance or overprovisioned tags
- Additional ways require additional tag bits to achieve optimal performance
- Increasing BTB capacity is not always the best investment of available storage resources

Future Research Directions

- BTB designs for minimizing tag storage
 - Balance tag and target storage in BTB designs
- Analytical models for tag sizing
 - Limiting design space exploration required
- Customizing BTB indexing functions
 - Limiting aliasing by balancing pressure on sets

Thanks!

Roman K. Brunner

roman.k.brunner@ntnu.no

NTNU

Rakesh Kumar

rakesh.kumar@ntnu.no

NTNU



Department of Computer Science
Computer Architecture Lab