CASPER

ISCA 2024

Maya Cache: A Storage-efficient and Secure Fully-associative Last-level Cache

Anubhav Bhatla, Navneet, Biswabandan Panda
Indian Institute of Technology Bombay
bhatlaanubhav2001@gmail.com | krnavneet13@gmail.com | biswa@cse.iitb.ac.in

Artifact

1. Introduction

Problem: Mirage proposed a last-level cache (LLC) design with complete security, only incurring a marginal performance overhead, and hence offers a sweet spot in terms of security and performance. However, it incurs an additional storage of 20% at the LLC with a static power overhead of 18% compared to a non-secure baseline.

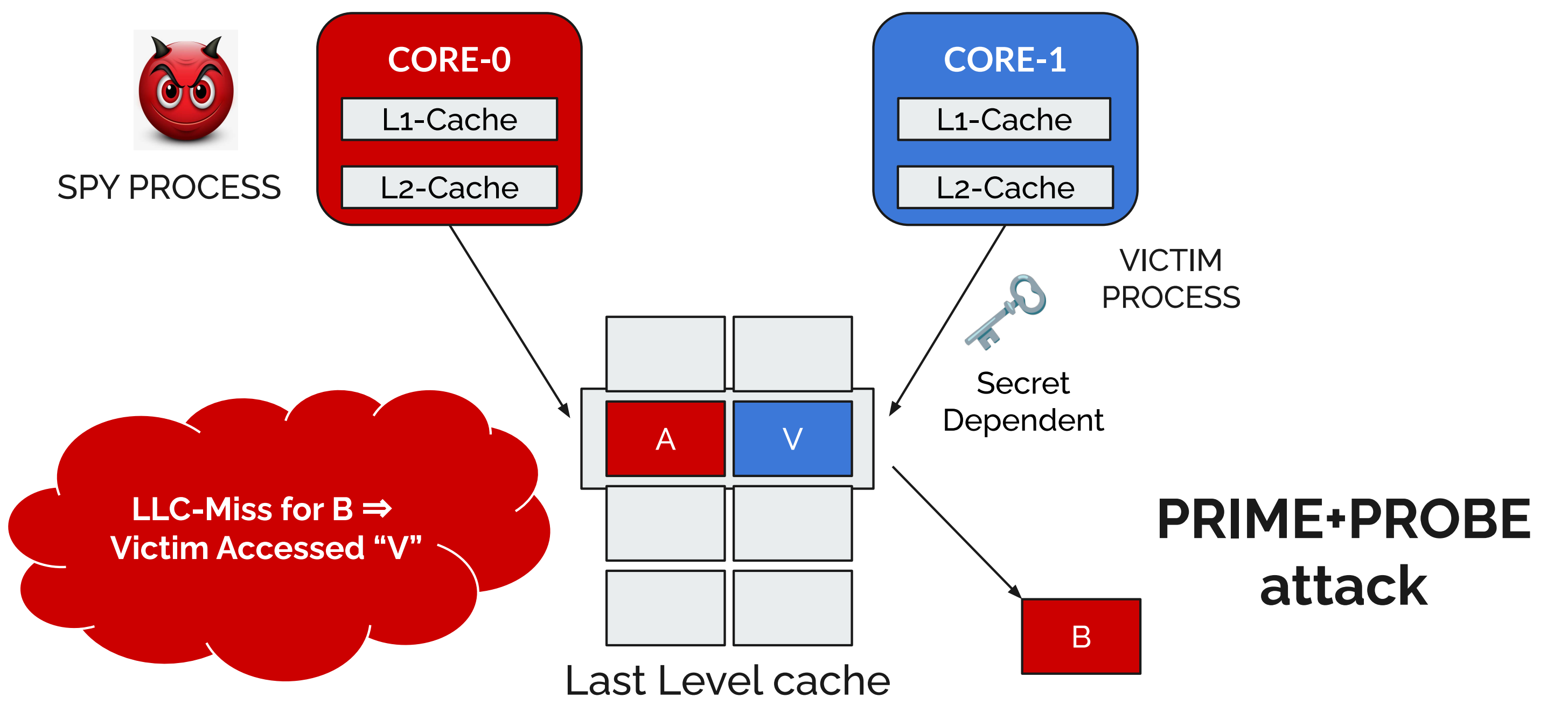
Goal: To propose an LLC design that can provide the illusion of a fully associative cache and, hence, the security guarantee without significant storage, power, and performance overhead.

Observations: A large fraction of the data store entries in the LLC are dead, and simply occupying space in the LLC.

Approach: We propose Maya, a secure and storage-efficient randomized LLC that provides the illusion of a fully associative LLC. Maya uses a smaller data store compared to the baseline, with additional tag entries for security and tracking reuse to avoid any performance overhead due to the reduced data store size.

2. Background

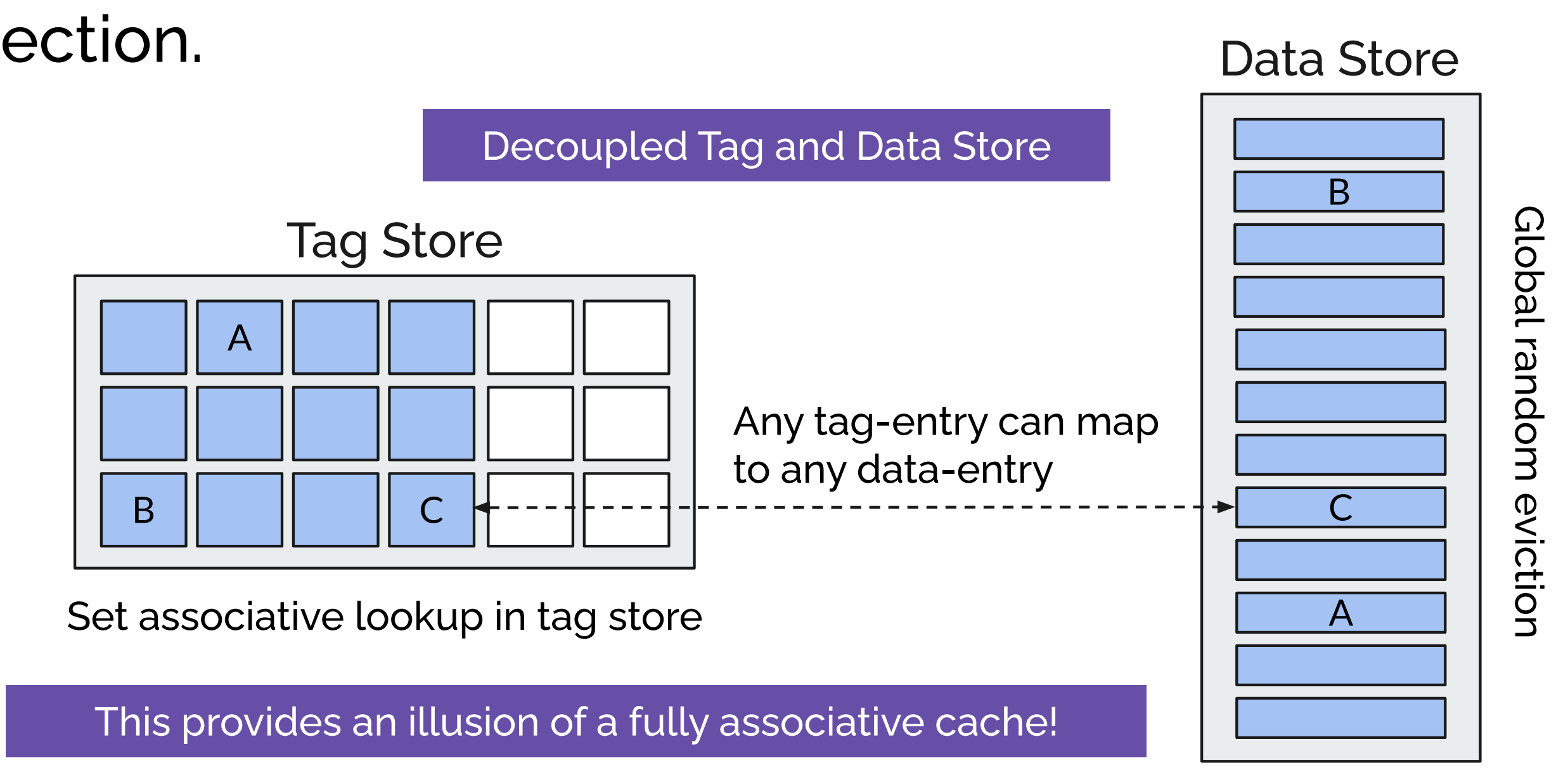
Eviction-based cache attacks: An attacker fills its data into an LLC that conflicts with the victim's data. The difference in access latency while probing reveals which lines have been accessed by the victim.
Shared memory-based attacks: The attacker flushes cache lines shared by the attacker and the victim. It then deduces the victim's access to these cache lines by observing memory access latency.



PRIME+PROBE attack

Mirage [1]: provides a proxy for a fully associative LLC with the help of a decoupled tag store and data store. It maintains a set associative tag lookup and global random eviction for data stores using pointer-based indirection.

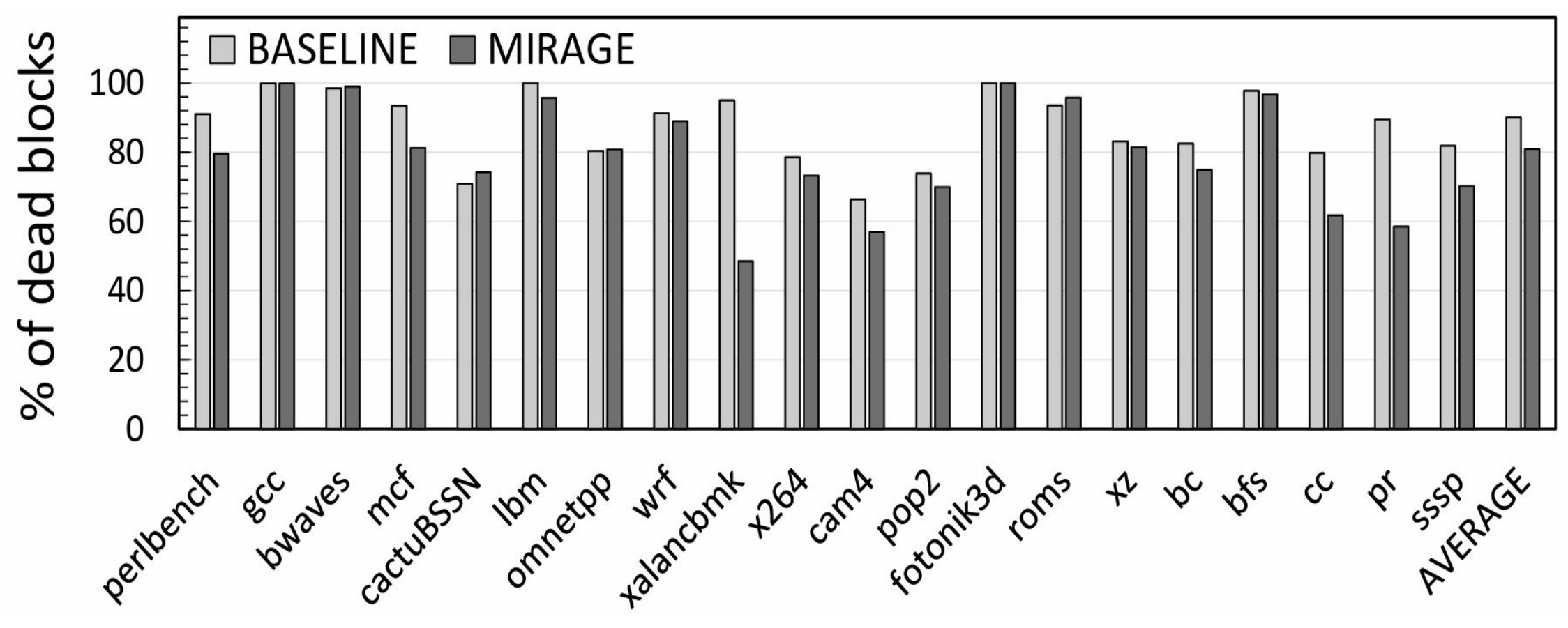
(+) Security
(+) Performance
(-) Storage/Area
(-) Power


Data Store
Tag Store
A
B
C
B
C
A
Any tag-entry can map to any data-entry
Global random eviction
Set associative lookup in tag store

Decoupled Tag and Data Store

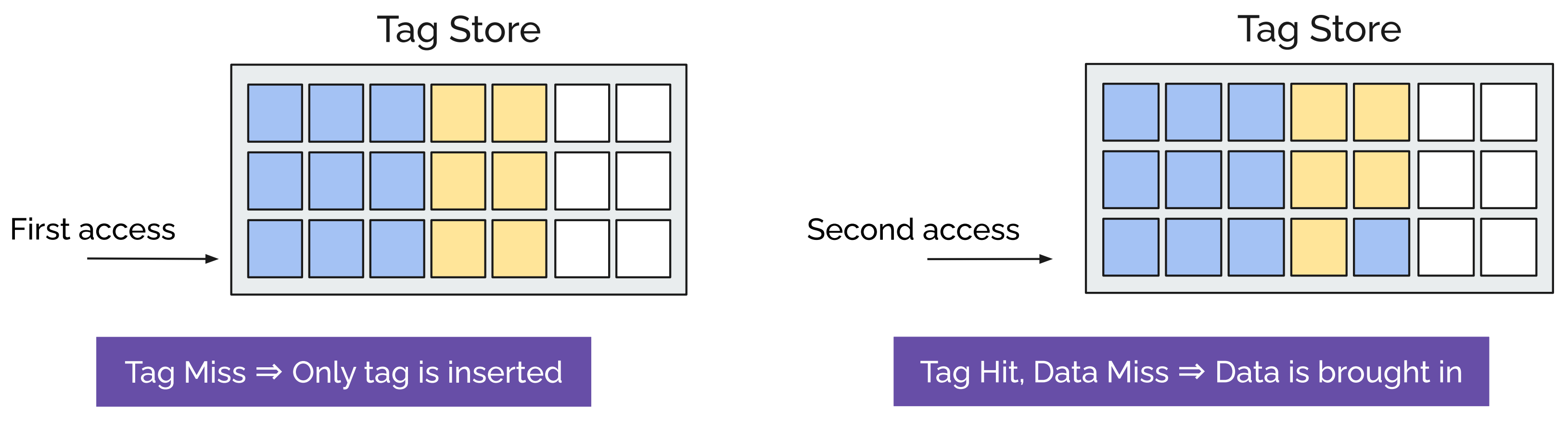
This provides an illusion of a fully associative cache!

3. Motivation



>80% dead blocks in the LLC

Naively reducing the data store size will lead to a large loss in performance ⇒ need a modification in the microarchitecture.
Reuse Cache [2]: uses a larger tag store with two different types of tag entries: ones with only tag and the others with both tag and data.

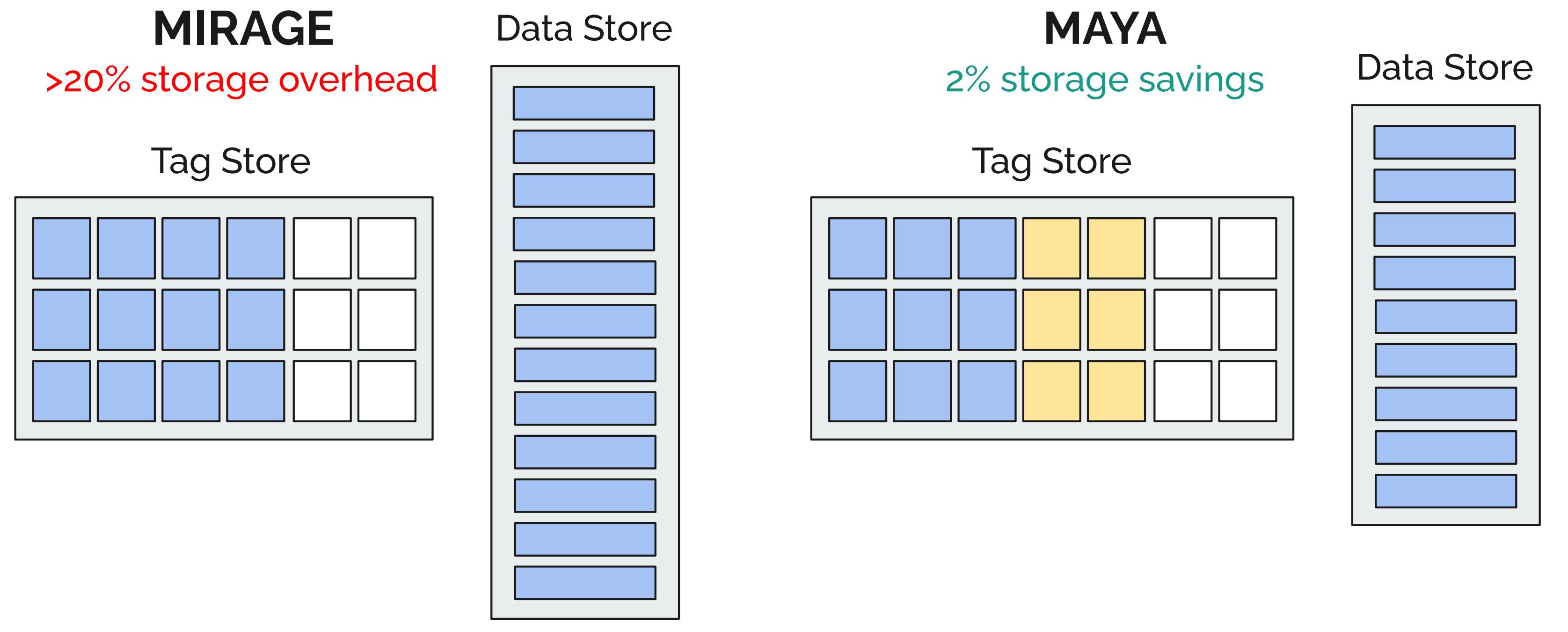


Tag Miss ⇒ Only tag is inserted

Tag Hit, Data Miss ⇒ Data is brought in

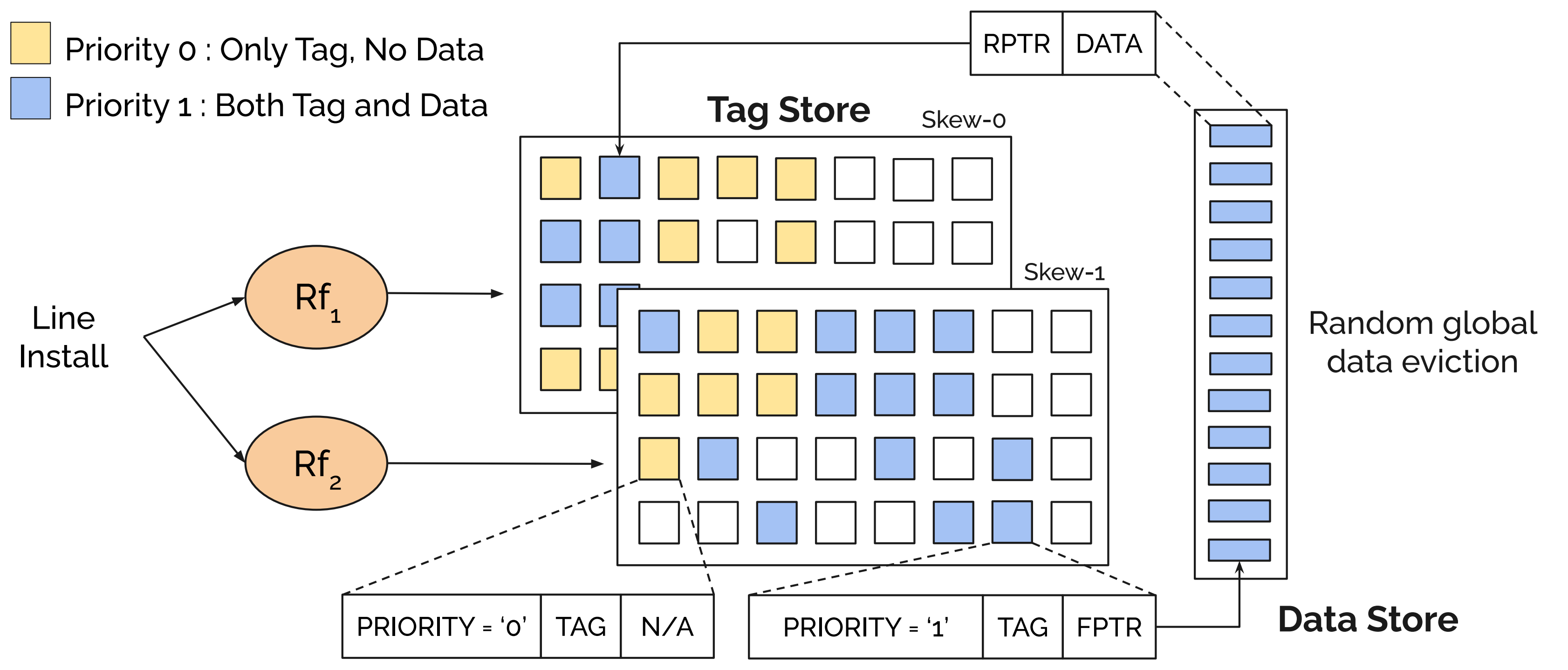
Data is stored only when reuse is detected

4. Maya Cache Design

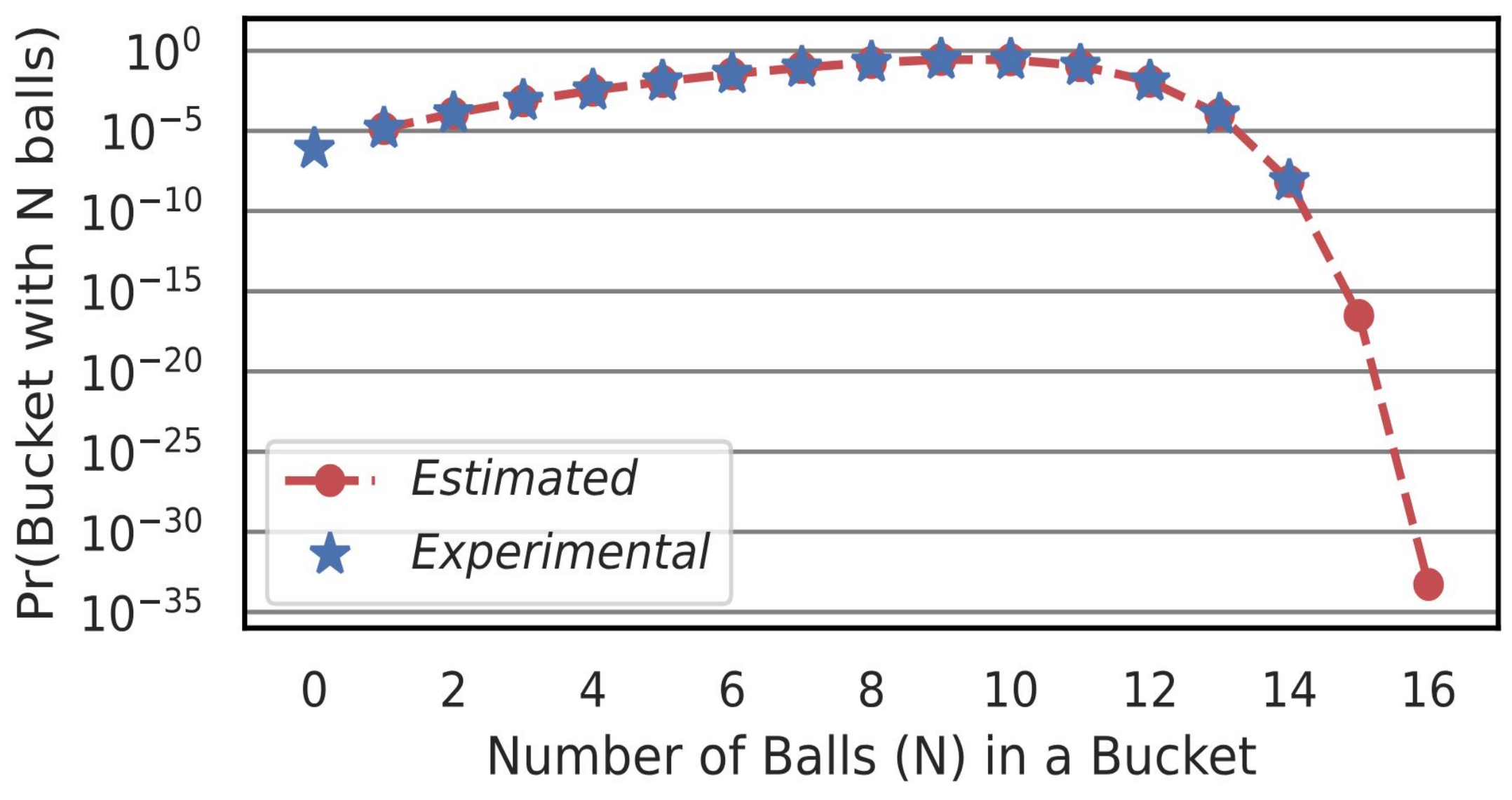


Data entry requires 8X the number of bits compared to a tag entry

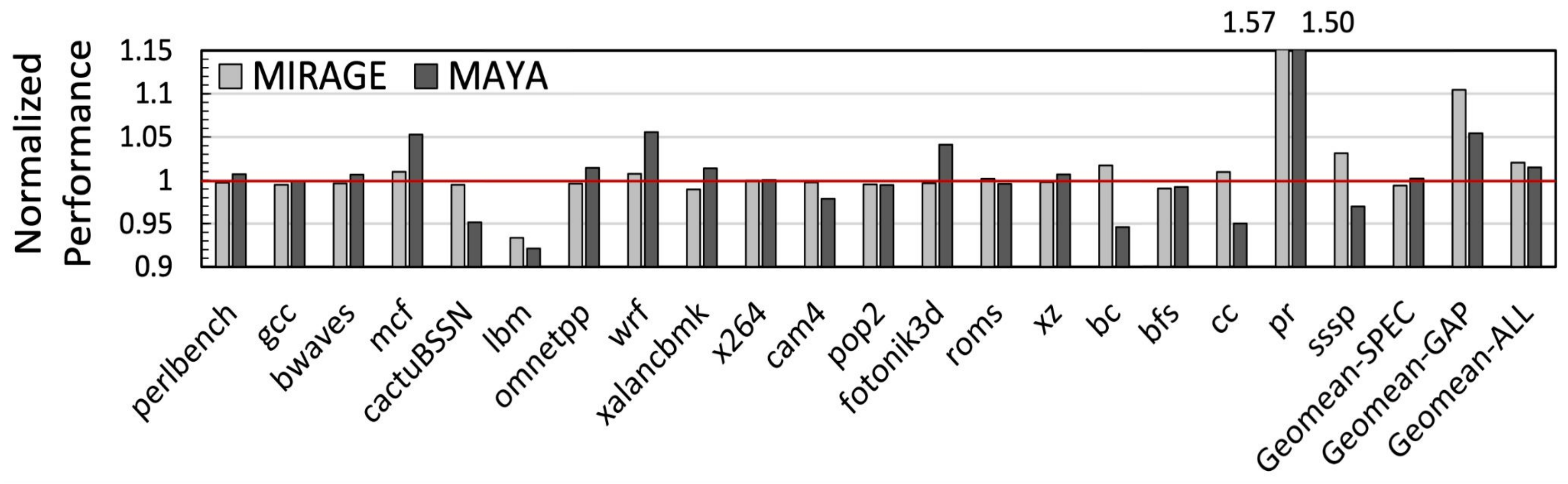
Maya introduces a priority bit in each tag entry to track reuse. It also provisions extra tag store entries and uses fewer data store entries.



5. Evaluation



No SAEs in 10^{16} years



Maya cache has a marginal performance improvement over a non-secure baseline.

6. Conclusion

Cache Design	Security (Installs per SAE)	Storage	Performance
Maya	10^{32} (10^{16} years)	-2%	+0.20%
Mirage	10^{34} (10^{17} years)	+20%	-0.55%
Mirage-Lite	10^{21} (22,000 years)	+17%	-0.55%
Maya ISO	10^{30} (10^{14} years)	+26%	+1.84%

The Maya cache design provides an optimal balance between security, storage, and performance.

References

[1] G. Saileshwar and M. Qureshi, "MIRAGE: Mitigating conflict-based cache attacks with a practical fully-associative design," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021.
[2] J. Albericio, P. Iba ñ ẽz, V. Vin ãls, and J. M. Llaber ía, "The reuse cache: Downsizing the shared last-level cache," in *2013 46th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2013, pp. 310– 321.

Supported by TrustLab, IIT Bombay