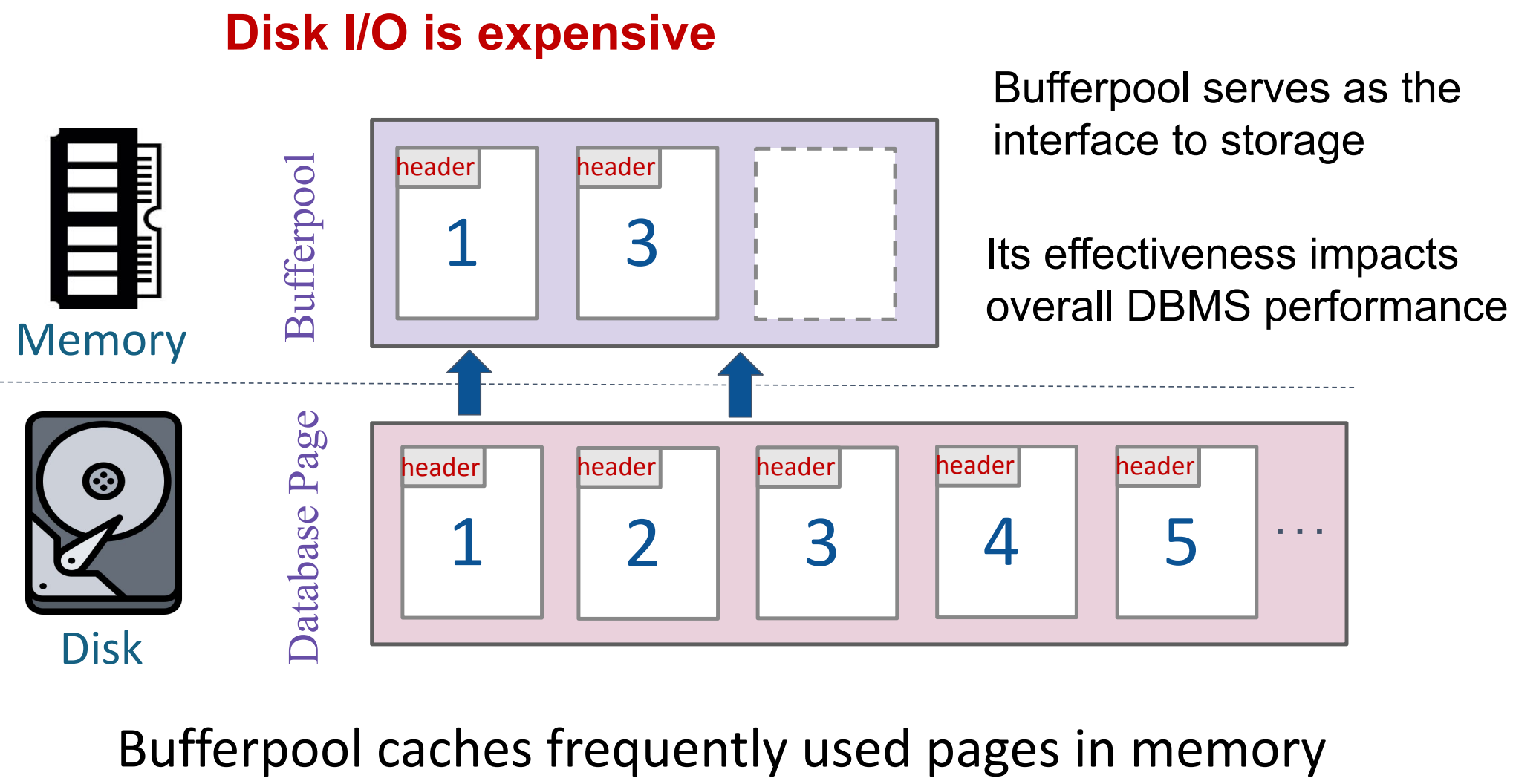
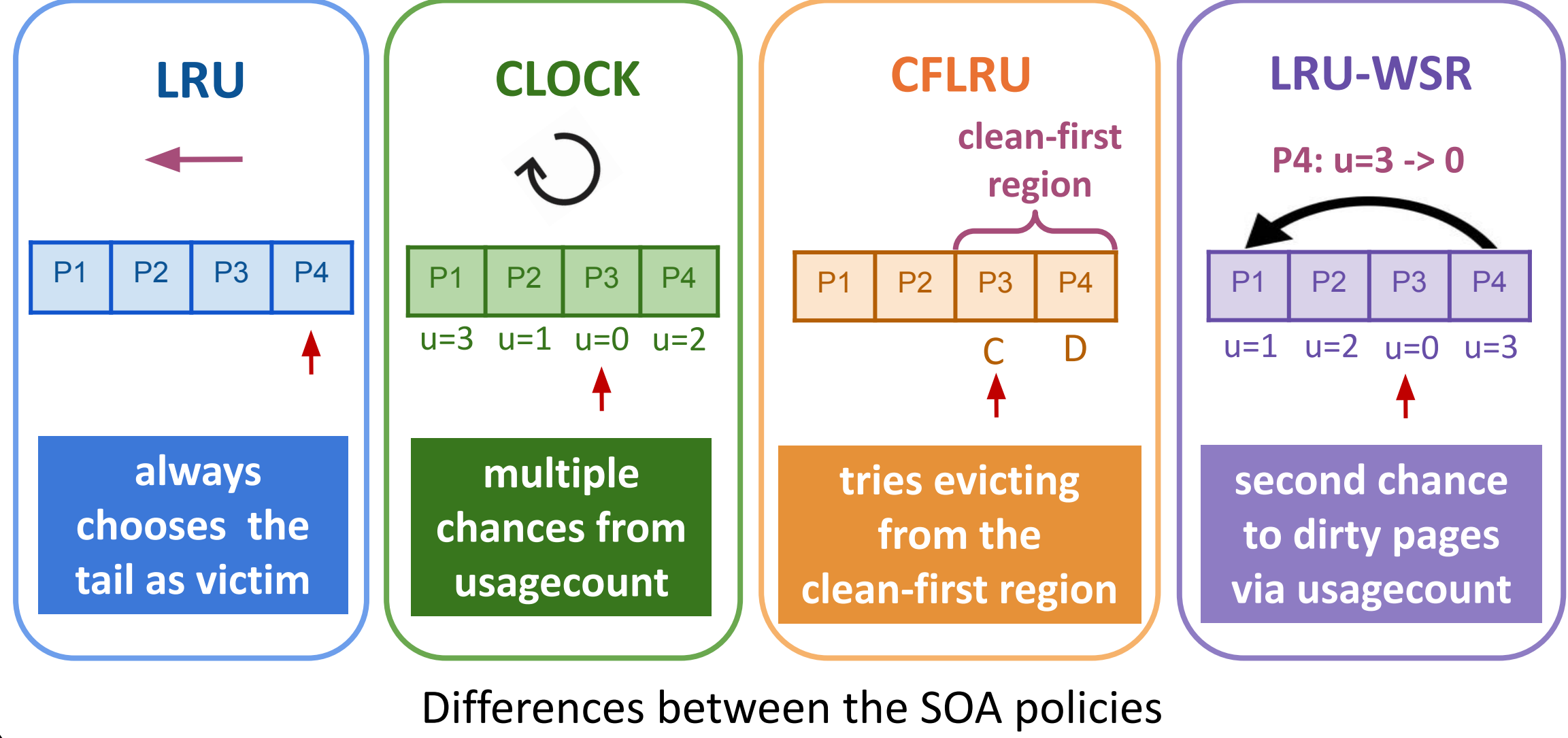


Database Bufferpool

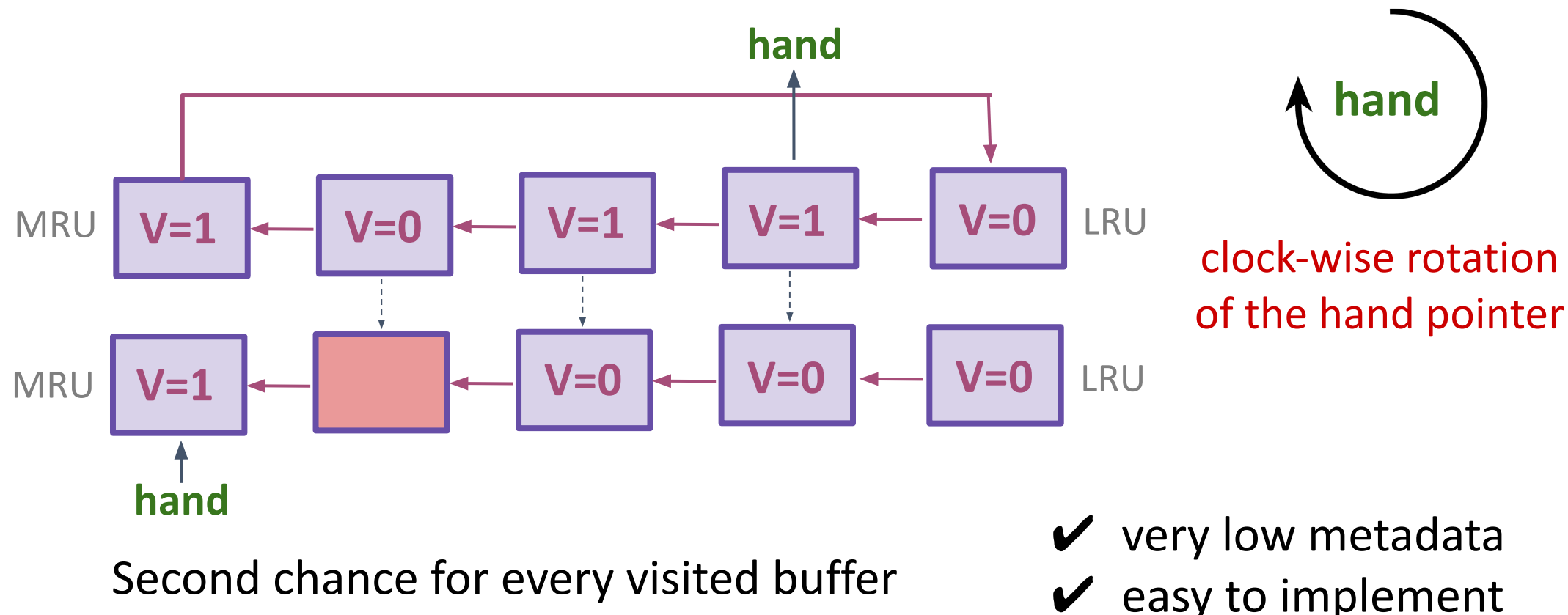


Page Replacement Policies



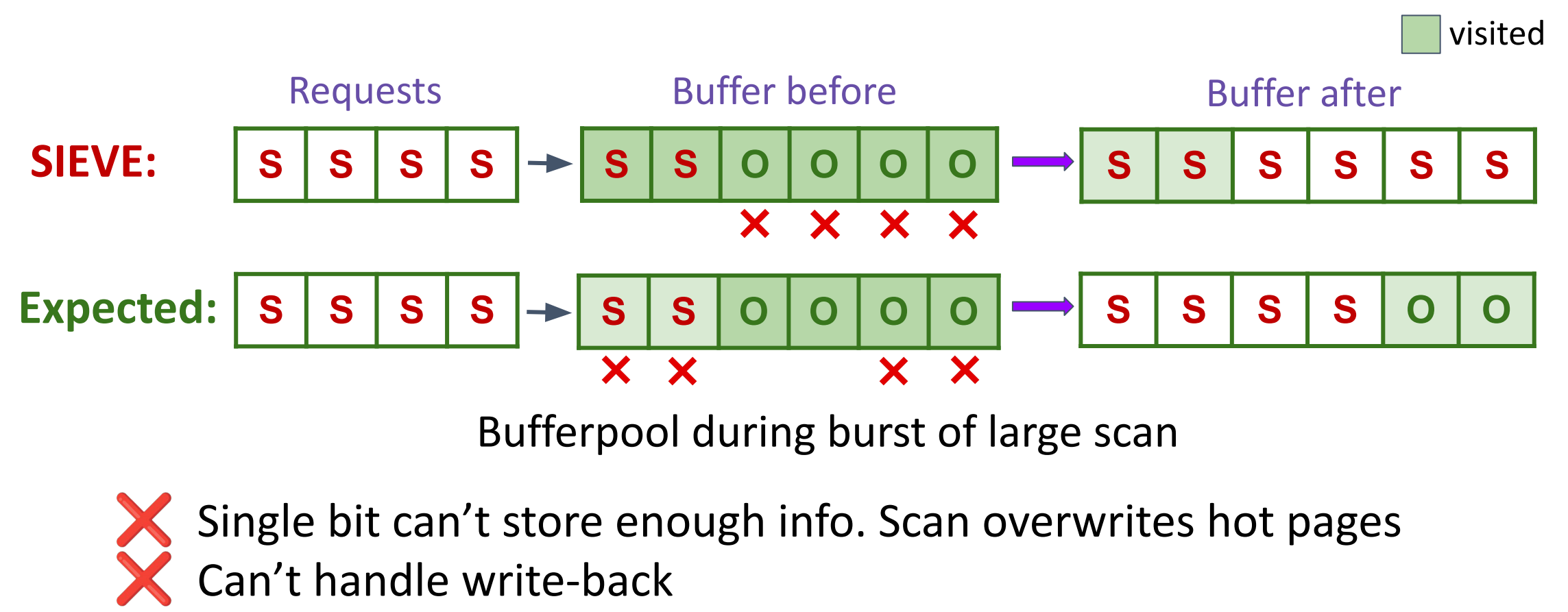
SIEVE

SIEVE is simpler than LRU. It is good for web caches
It achieves lazy promotion and quick demotion



Challenges of SIEVE

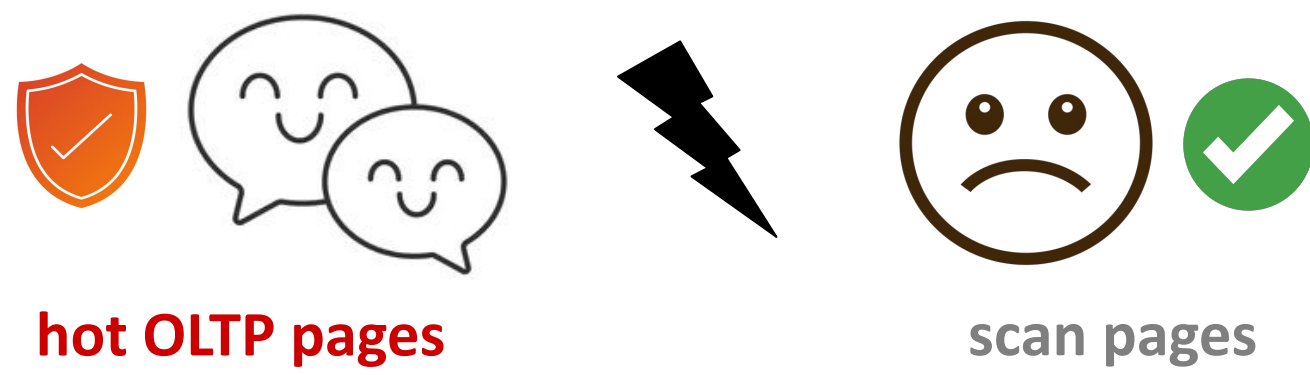
SIEVE can't differentiate between hot OLTP pages and scan pages



SIEVE-DB

We introduce

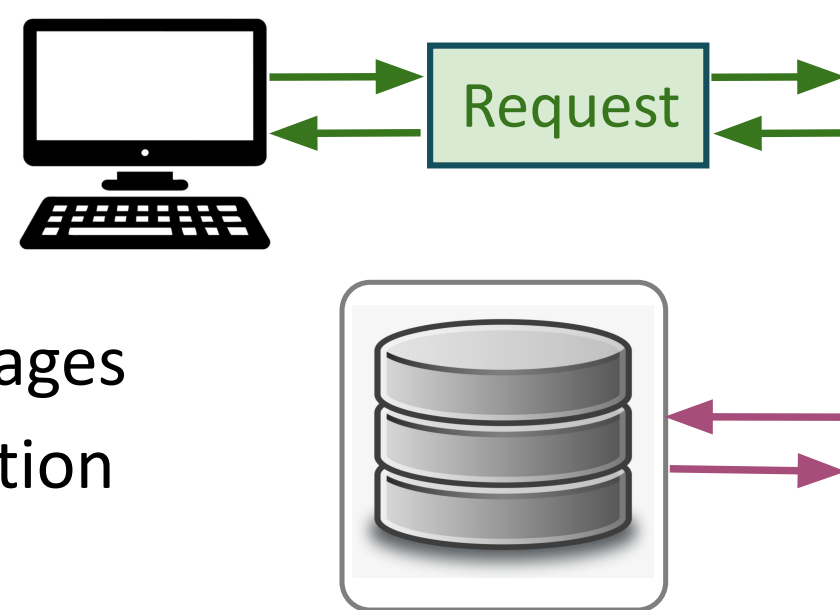
- ✓ scan-resistance using an extra protected bit per buffer



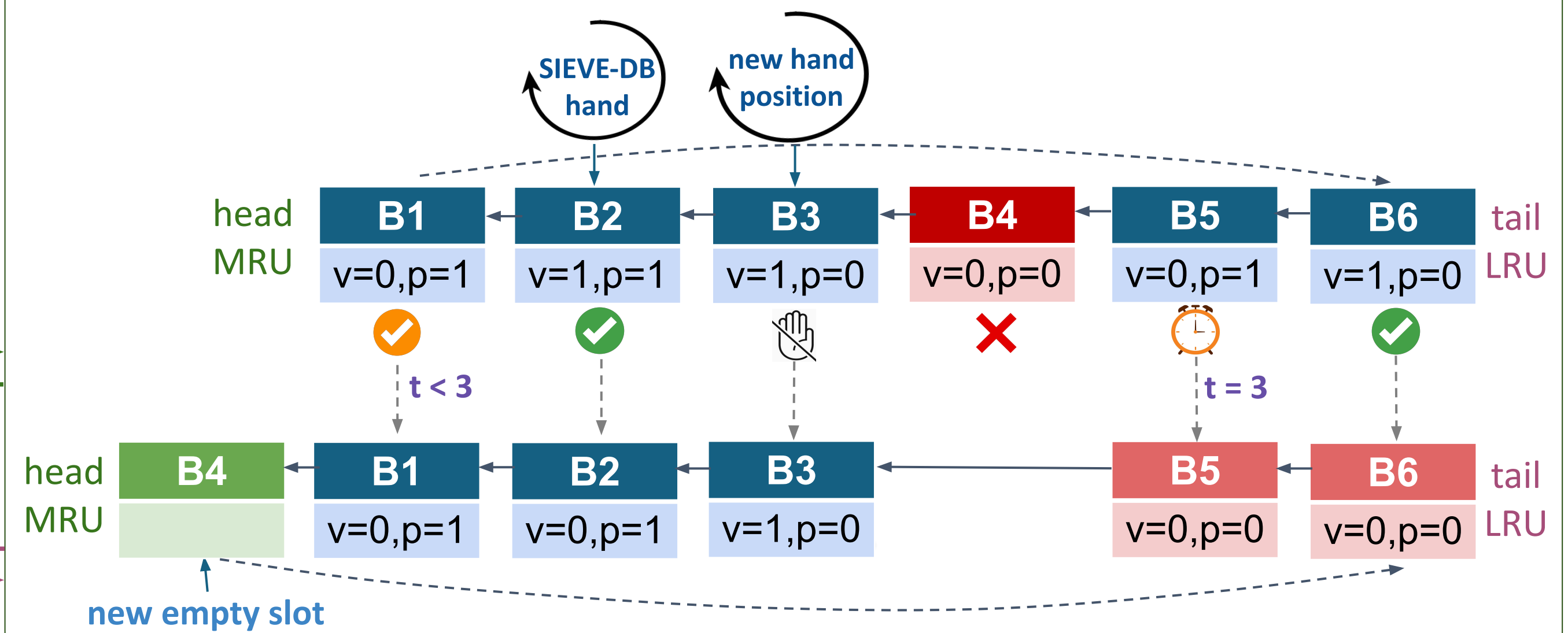
- ✓ ring buffer to store scan pages

This allows

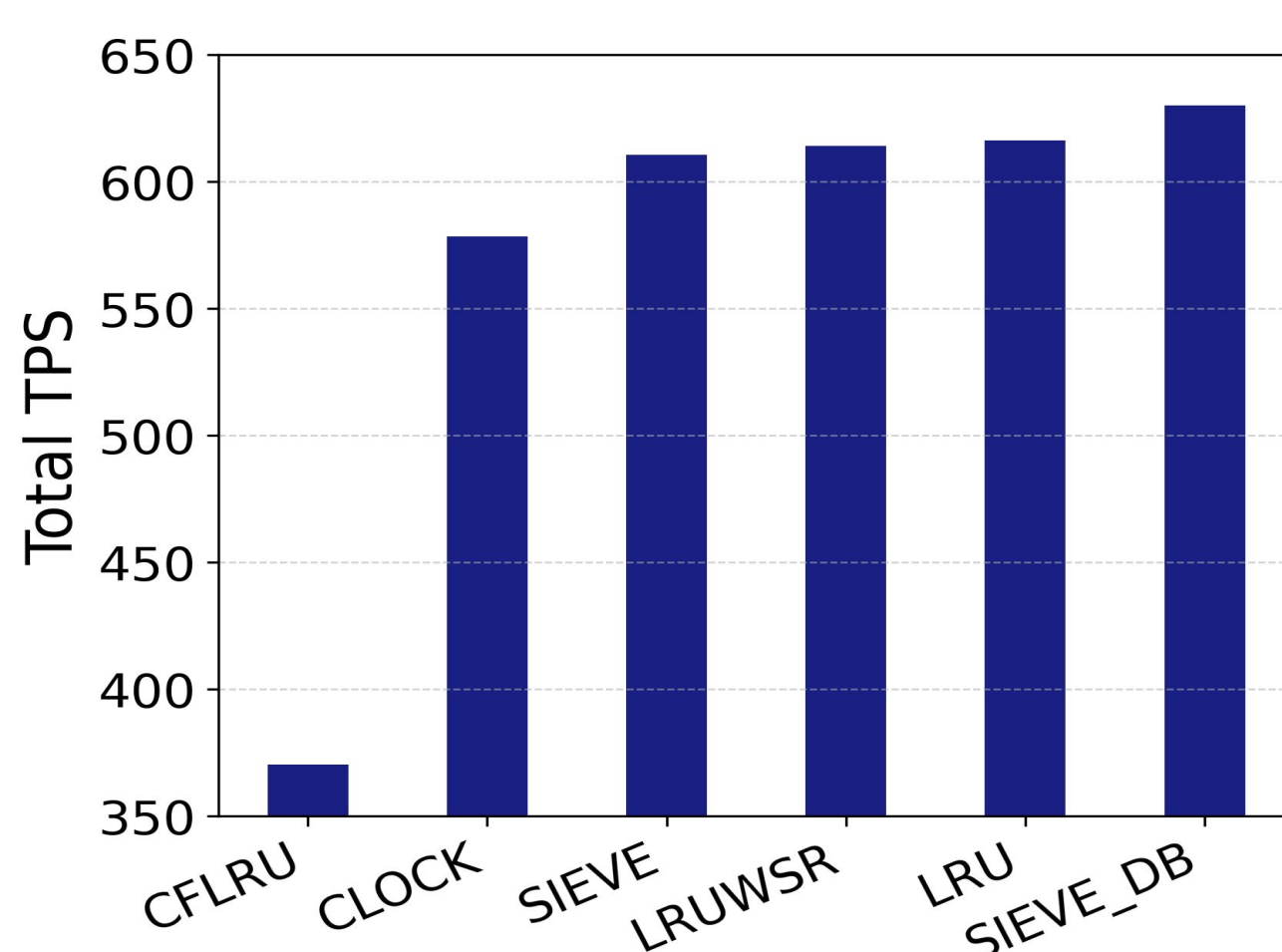
- ✓ to retain reuse-worthy pages
- ✓ light-weight implementation



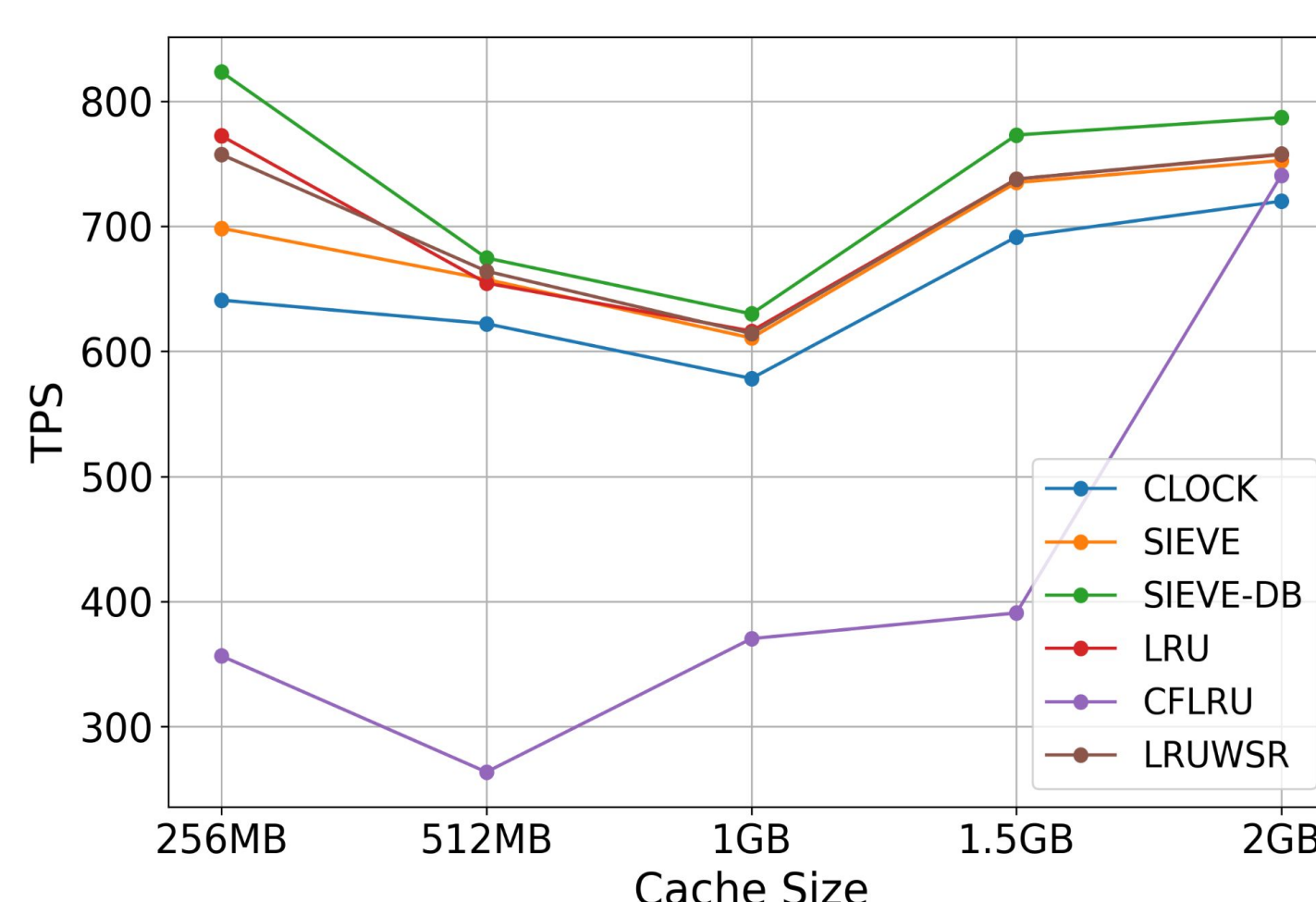
- ⌚ For $t = 3$ passes if $v = 0$ and $p = 1$: $p \rightarrow 0$ aging out protection
- ✓ passed by setting $v = 1 \rightarrow v = 0$
- ✓ passed, $t < 3$
- ✋ untouched



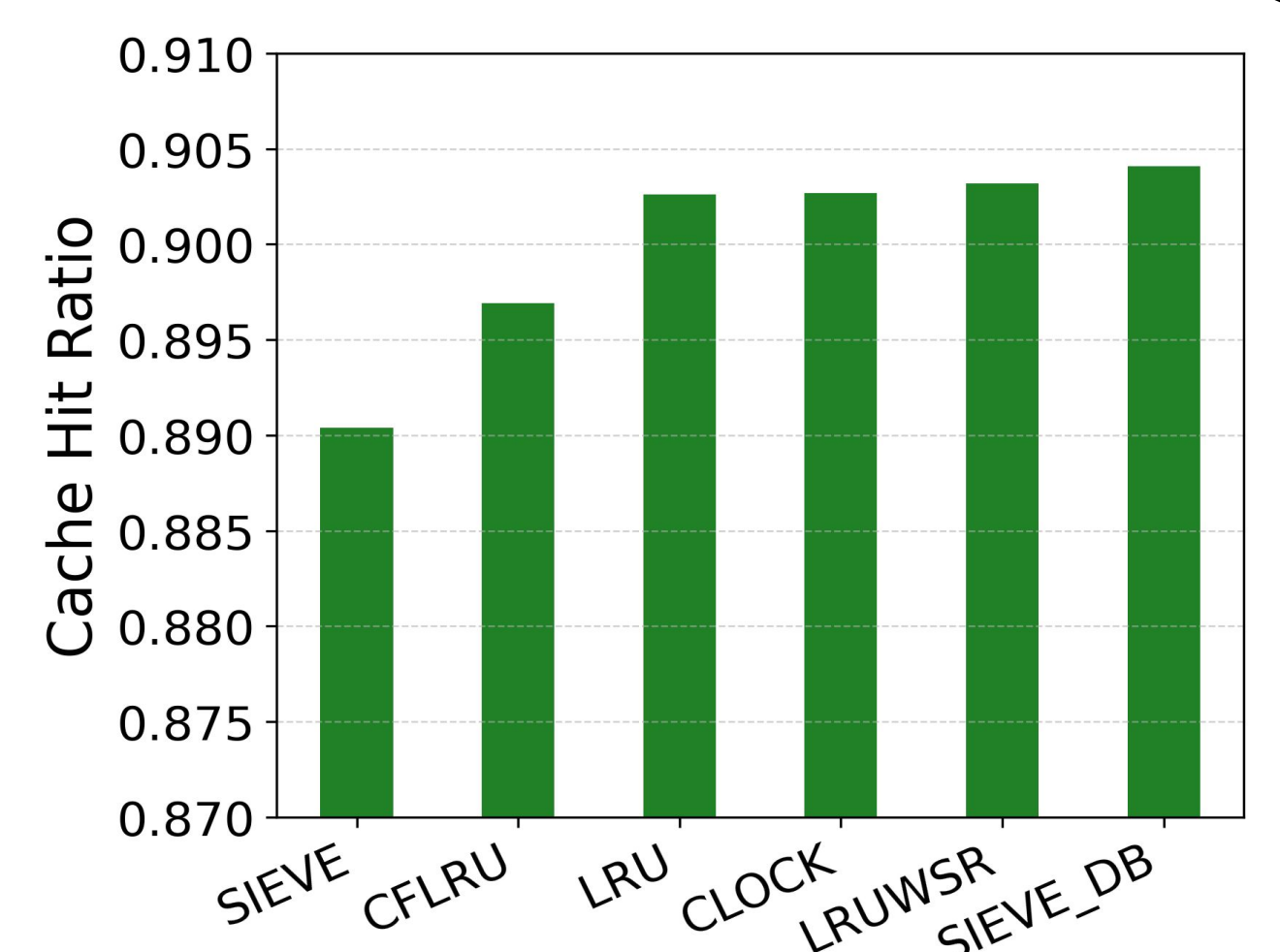
Preliminary Results in PostgreSQL 18



SIEVE-DB improves throughput



It outperforms SOA for all cache sizes



It provides comparable hit rate