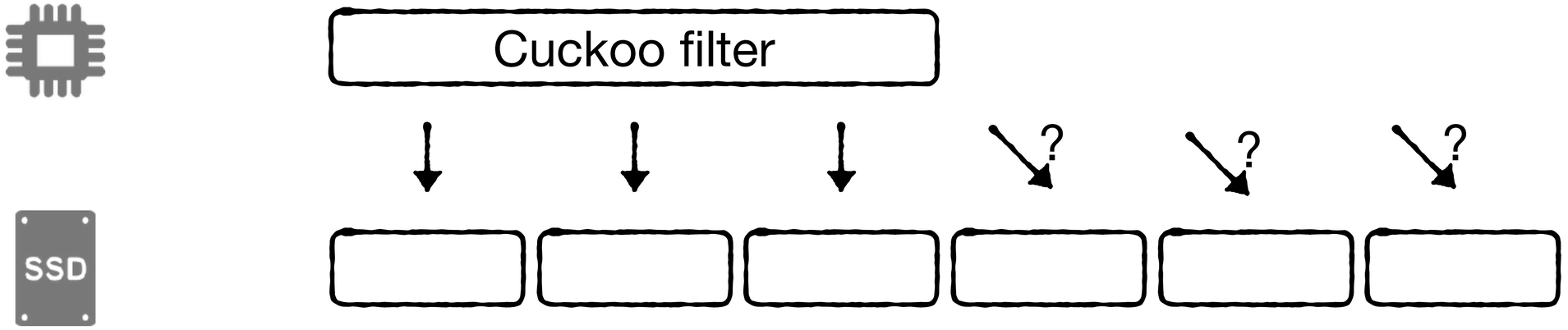
Tutorial on Circular Logs & Cuckoo Filters

Database System Technology

Niv Dayan

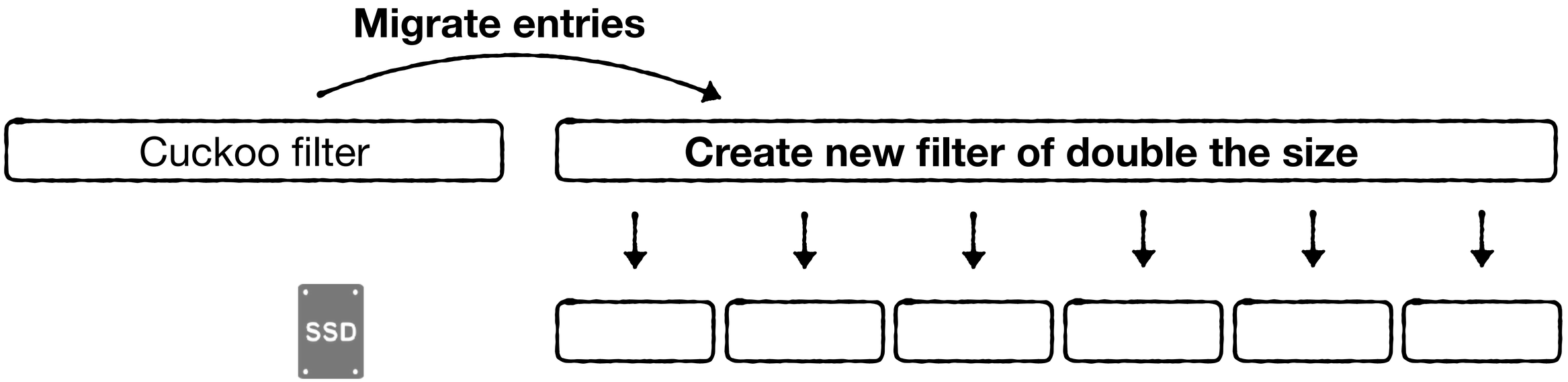
Question 1 - Expanding Cuckoo Filters

A Cuckoo filter is allocated with a fixed capacity. As a circular log grows, we must expand its cuckoo filter to map more data. Devise a Cuckoo filter expansion algorithm. Ideally, this algorithm should maintain constant time performance and not have to read any data from storage. Comment on any trade-offs or downsides.



Question 1 - Expanding Cuckoo Filters

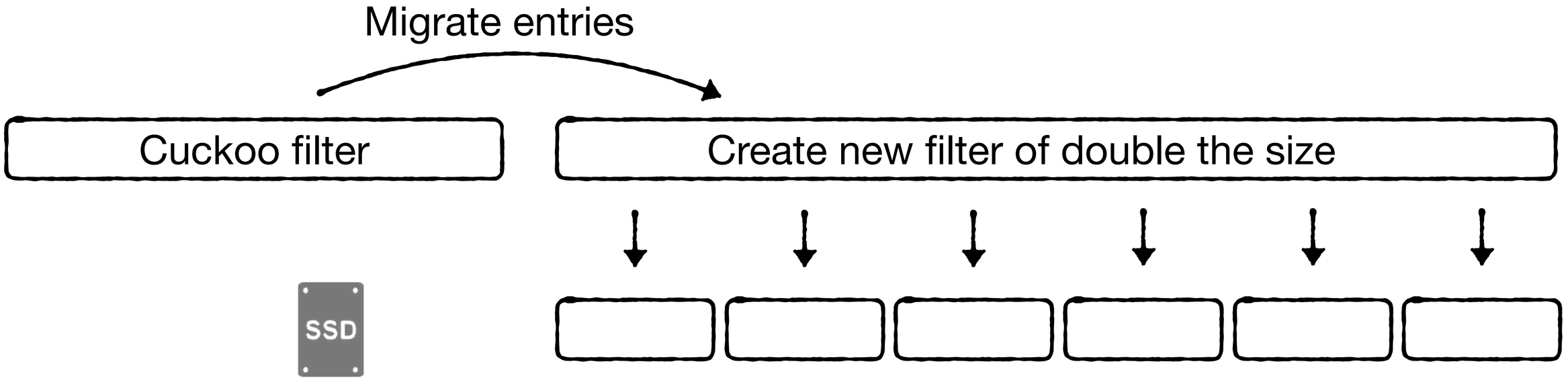
A Cuckoo filter is allocated with a fixed capacity. As a circular log grows, we must expand its cuckoo filter to map more data. Devise a Cuckoo filter expansion algorithm. Ideally, this algorithm should maintain constant time performance and not have to read any data from storage. Comment on any trade-offs or downsides.



Question 1 - Expanding Cuckoo Filters

A Cuckoo filter is allocated with a fixed capacity. As a circular log grows, we must expand its cuckoo filter to map more data. Devise a Cuckoo filter expansion algorithm. Ideally, this algorithm should maintain constant time performance and not have to read any data from storage. Comment on any trade-offs or downsides.

Challenge: we do not have the full keys to rehash. We only have fingerprints.

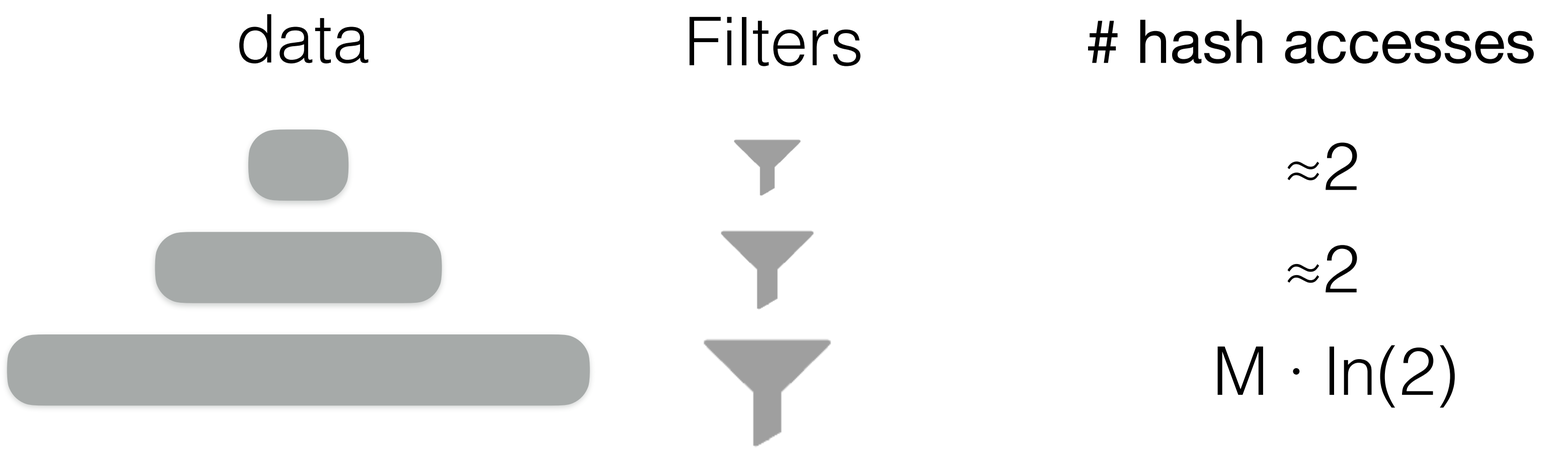


Question 2 - Cuckoo Filter for LSM-tree

As we have seen, the expected worst case query cost over Bloom filters for a basic LSM-tree is O(L + M), where L is the number of levels and M is the number of bits per entry. (Assume only unique entries in the tree).

(A) How can we employ a cuckoo filter to achieve constant time?

(B) What are the implications on the false positive rate and memory footprint? Any downsides compared to plain Bloom filters?



Question 3 - Hot/Cold Data Separation

Consider a circular log where the physical capacity consists of 70% static data (never updated), 10% hot data, and 20% over-provisioning.

(A) Estimate a lower bound and upper bound for write-amplification assuming no hot/cold data separation.

(B) Estimate write-amplification assuming perfect hot/cold data separation.