

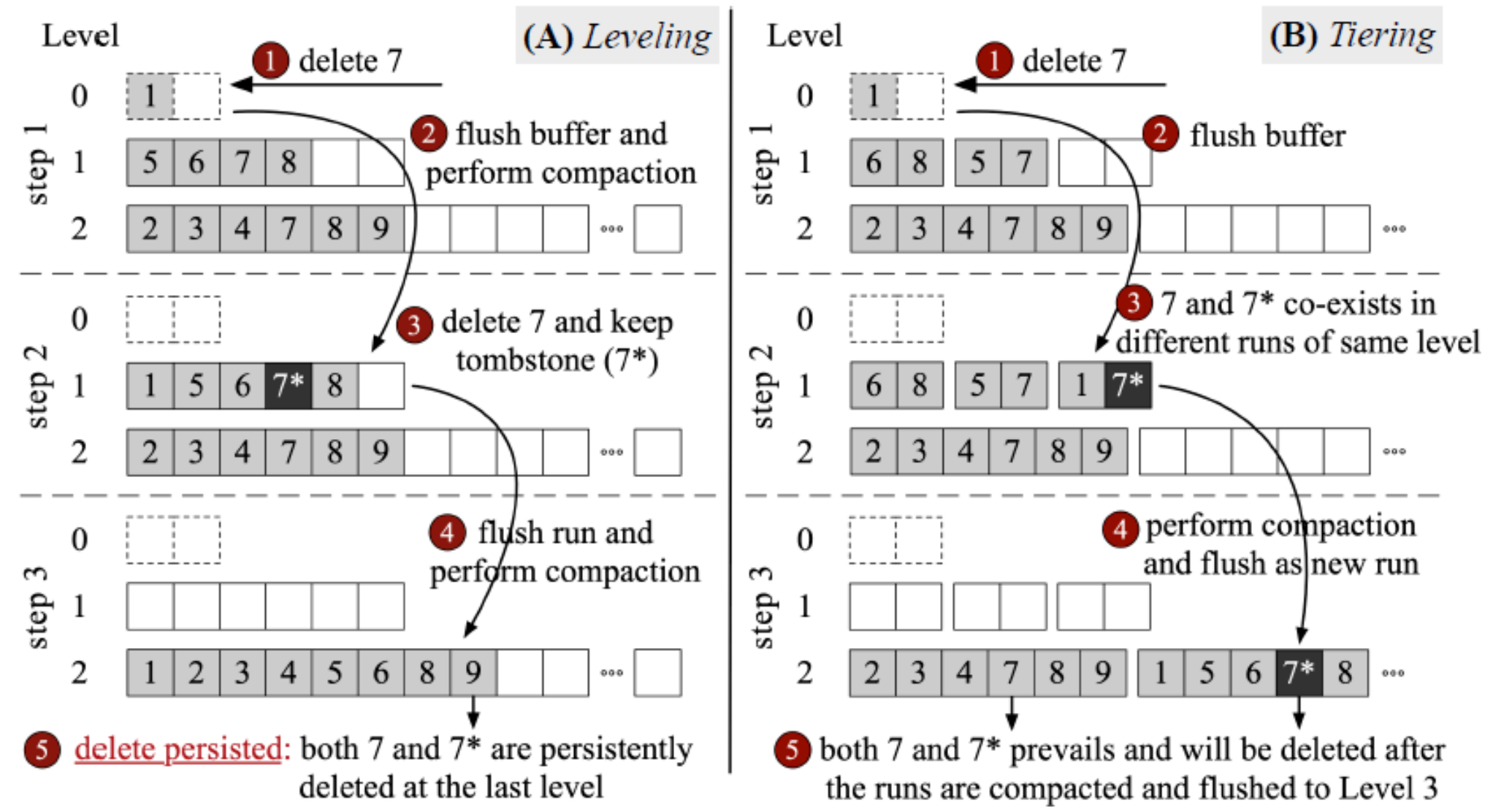
LSM-Trees are everywhere



Efficient ingestion
Competitive reads
Good space utilization

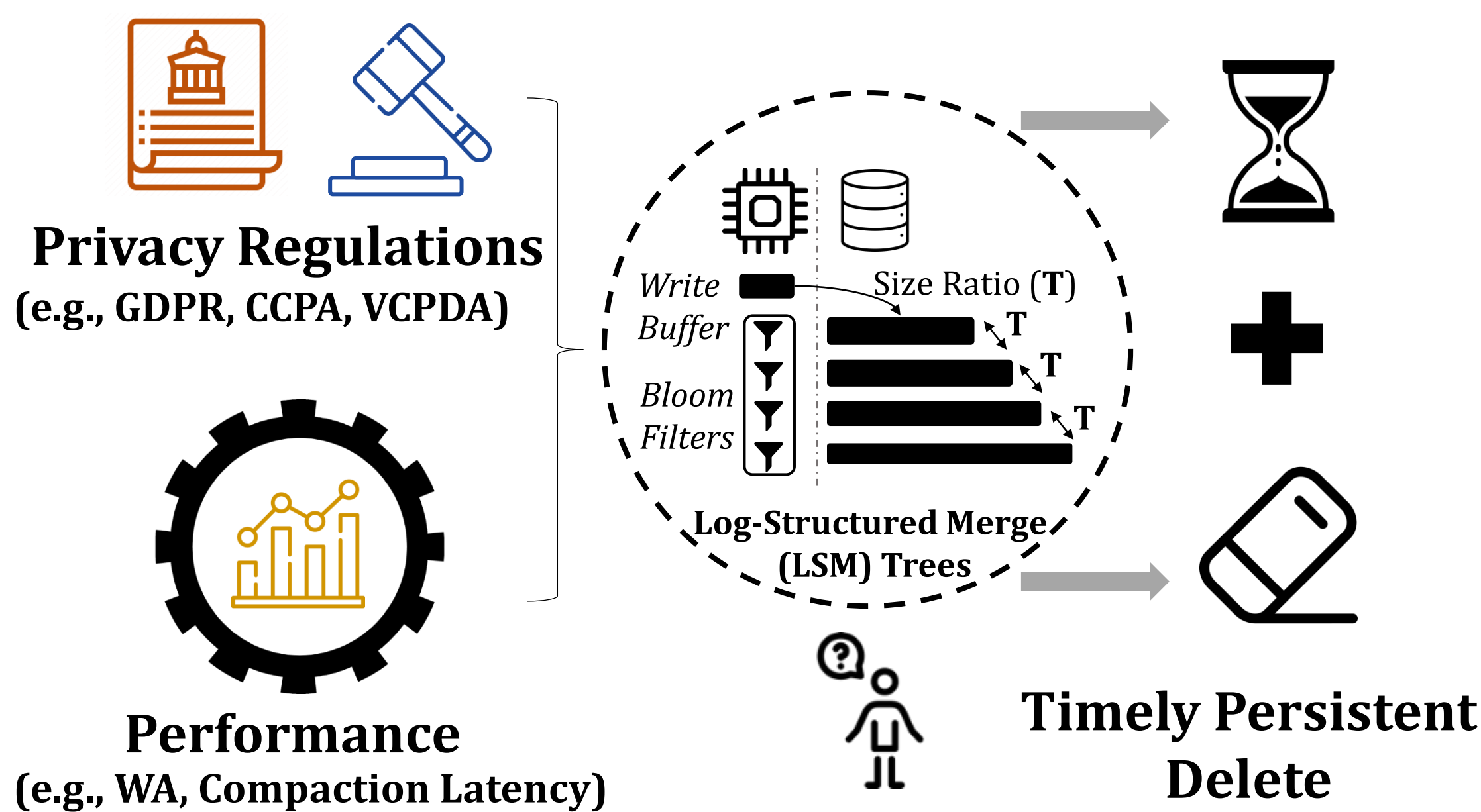
Out-of-place
Paradigm

Out-of-place Deletes By Tombstones

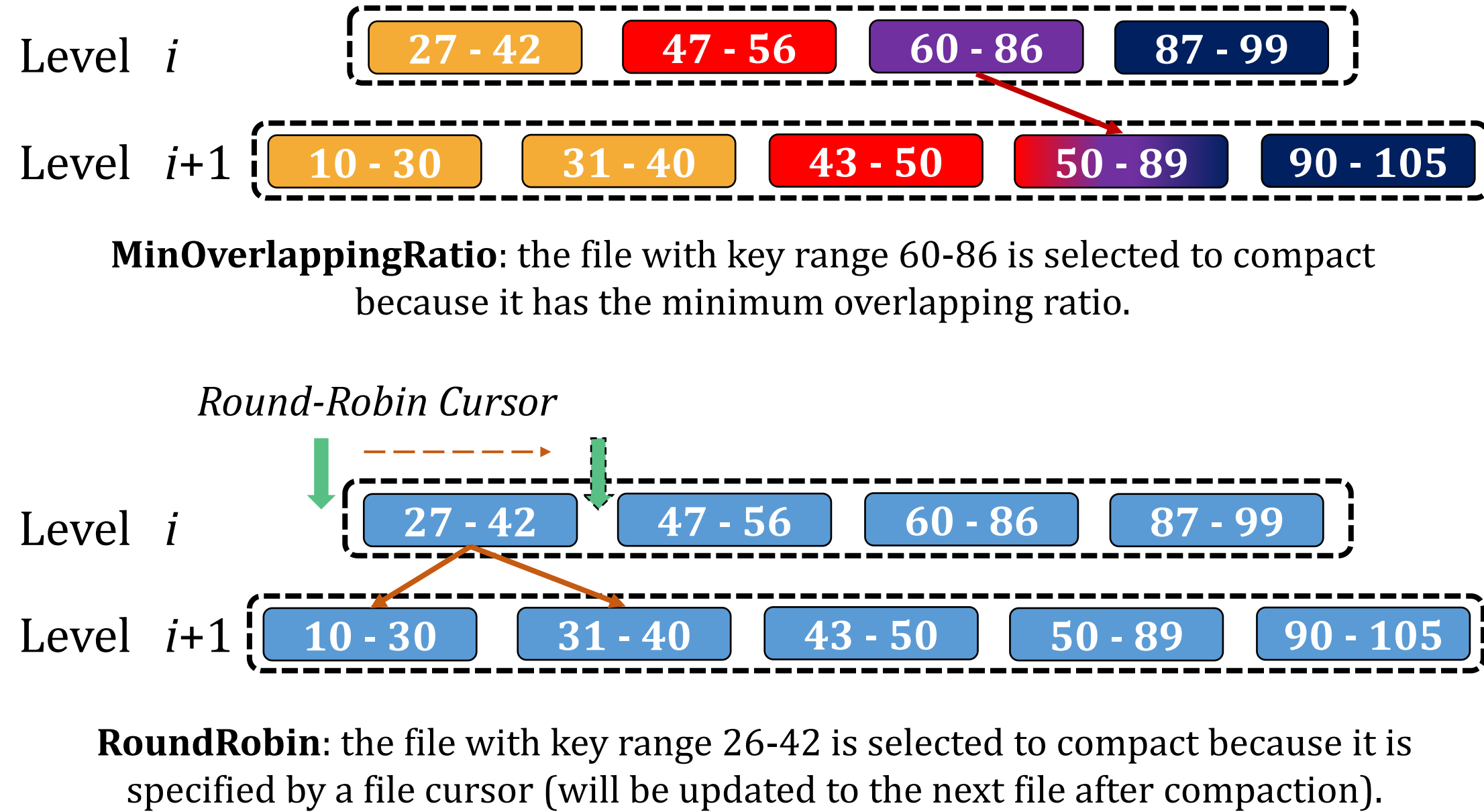


In an LSM tree, for every tombstone, there can be (A) one matching entry per level for leveling or (B) one matching entry per tier per level.

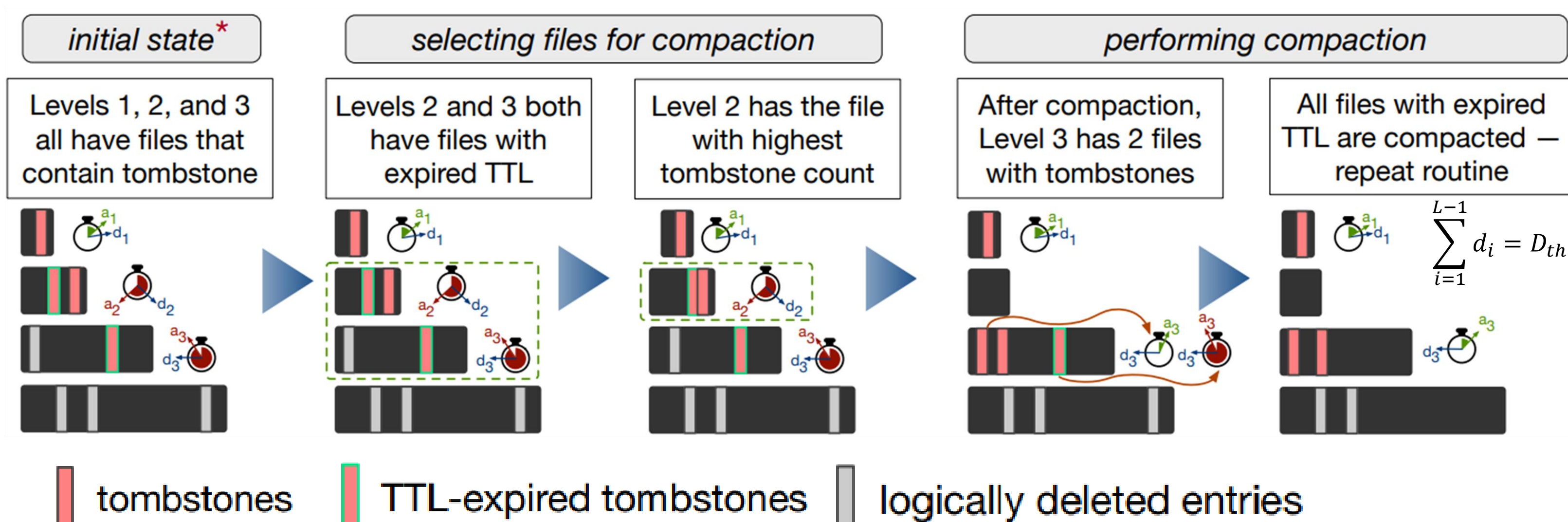
New Challenges for LSM-Trees



Partial Compaction Policies



FADE (FAST DELETE): Enforcing a Finite Bound for Delete Persistence Latency



Setting Environment

after every flush, perform the following check

```
// Update d[i] if a new level is added to the tree
// T = size ratio; level_count = levels in tree
// D_th = delete persistence threshold
double x = D_th * T / pow(T, level_count - 1);
if (new_level_added()) {
    for (int i = 0; i < level_count; ++i) {
        d[i] = x * pow(T, i);
        if (i > 0)
            d[i] += d[i - 1];
    }
}
```

Acheron System Overview



Evaluation

