RocksDB

Challenges of LSM-Trees in Practice

Siying Dong

Software Engineer, Database Engineering Team @ Facebook

Sep 29, 2015

What is RocksDB?

What is RocksDB?

- Key-Value persistent store
- Point / range lookup
- Optimized for flash and memory
- C++ library
- Other language bindings
- Fork of LevelDB

RocksDB As Storage Engine of Data Management Systems

mongoDB

RocksDB | Mmap | WiredTiger

MySQL

BerkeleyDB ...... InnoDB | RocksDB

Yahoo Sherpa

RocksDB

And many more ...

RocksDB As Storage in Applications

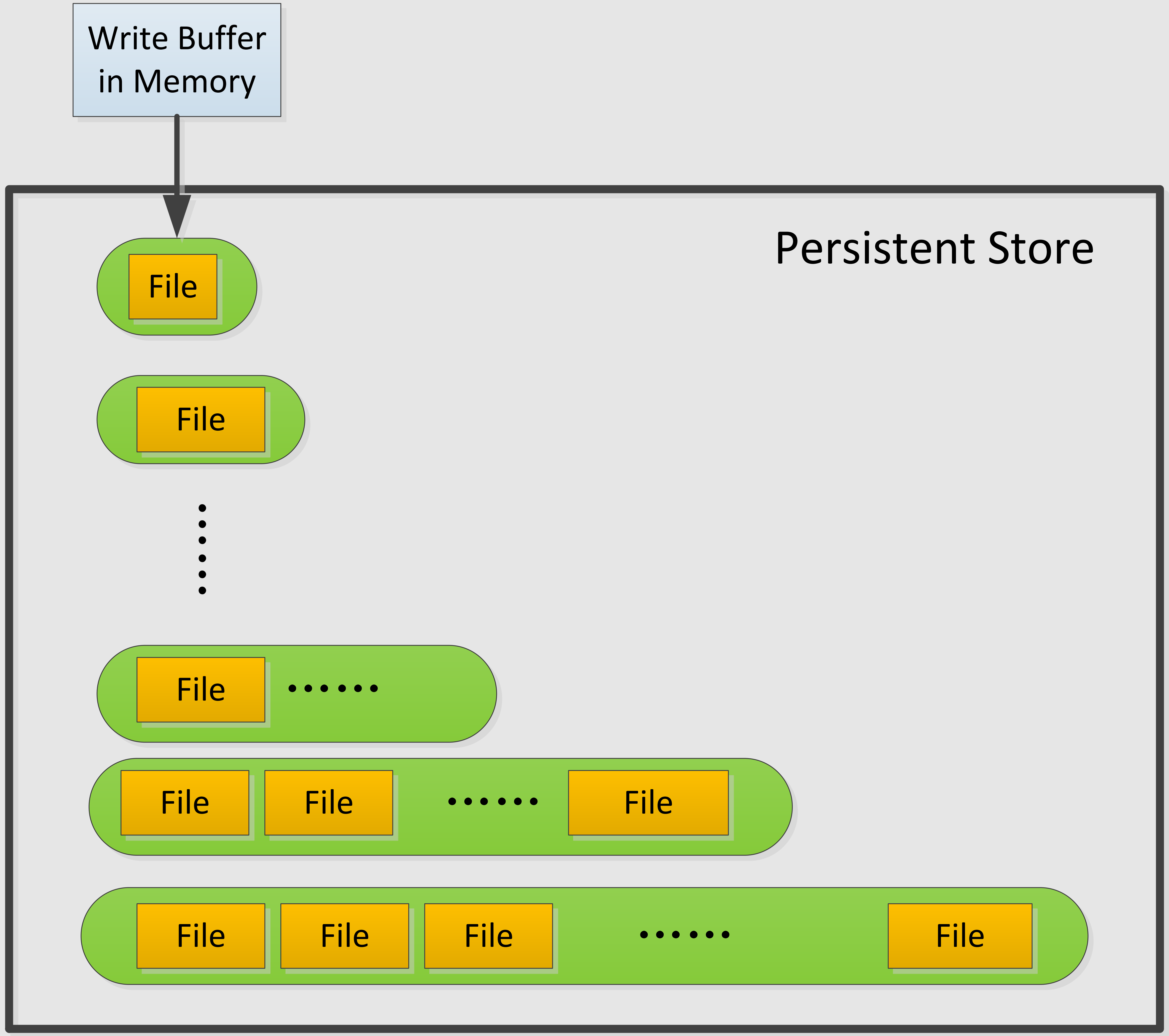
- Facebook: many backend services
- LinkedIn’s FollowFeed
- Apache Samza
- Iron.io
- Tango Me
- And more…

Agenda

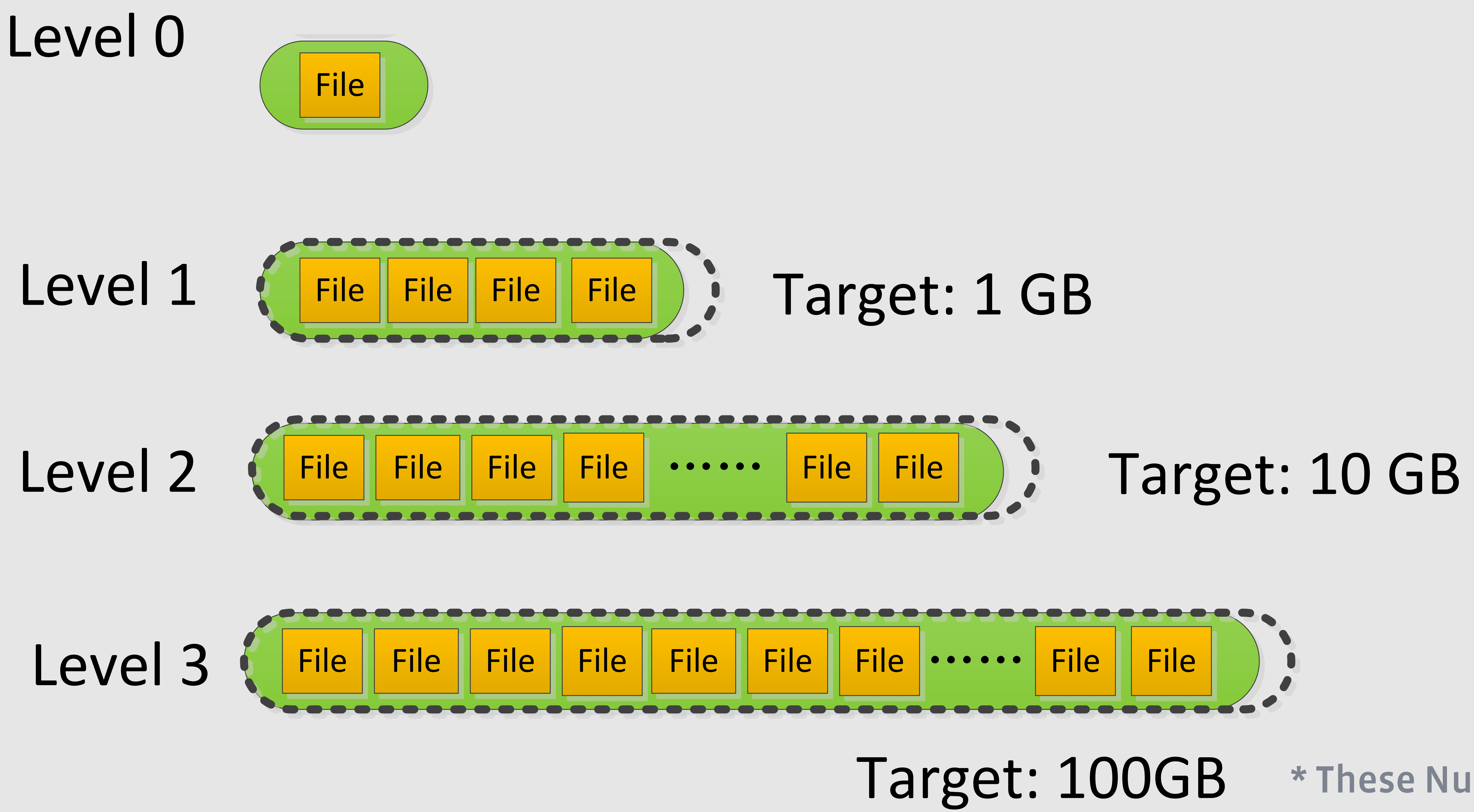
1	RocksDB Architecture
2	Challenges

RocksDB Architecture

Log Structure Merge Tree

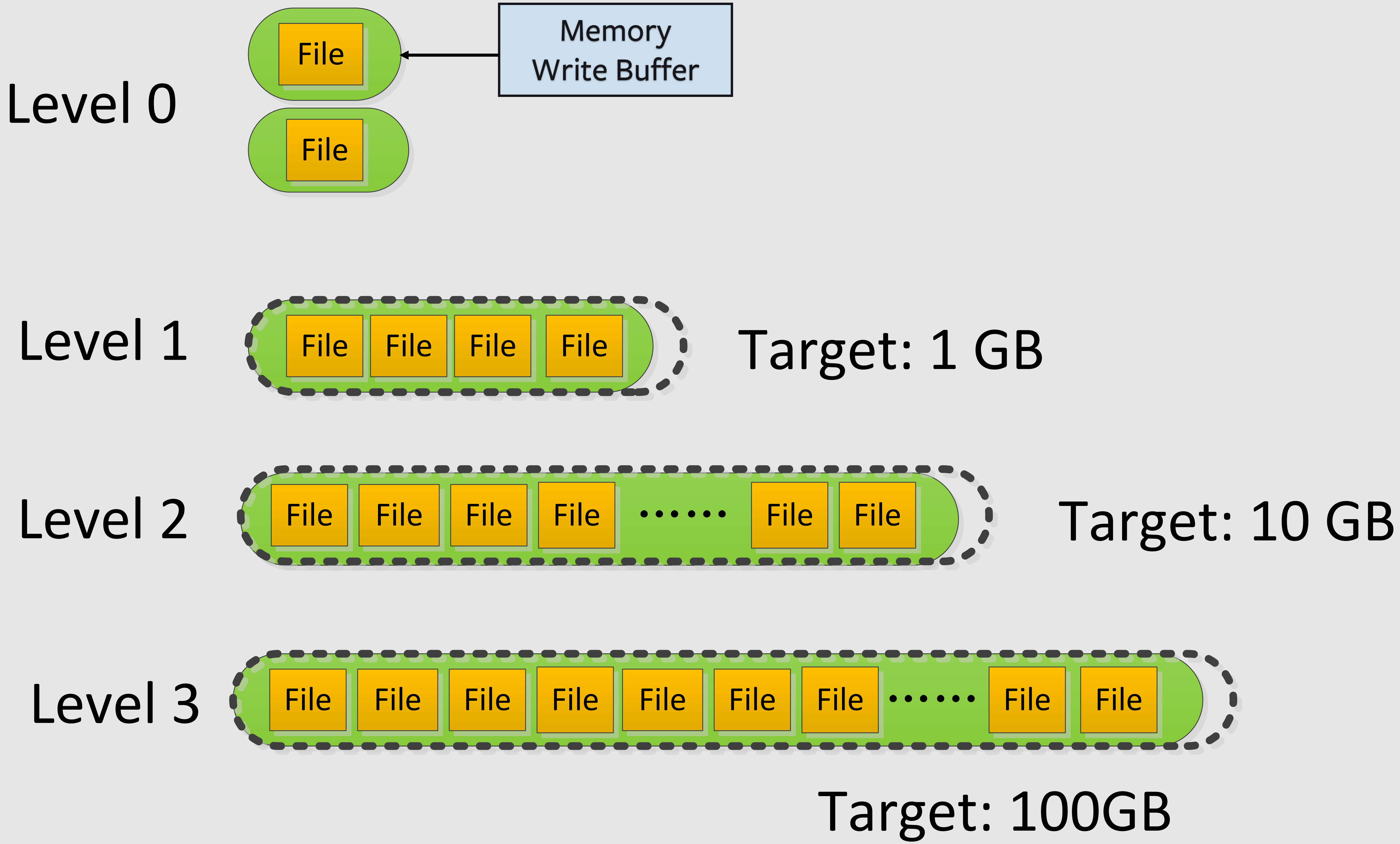


Level-Based Compaction

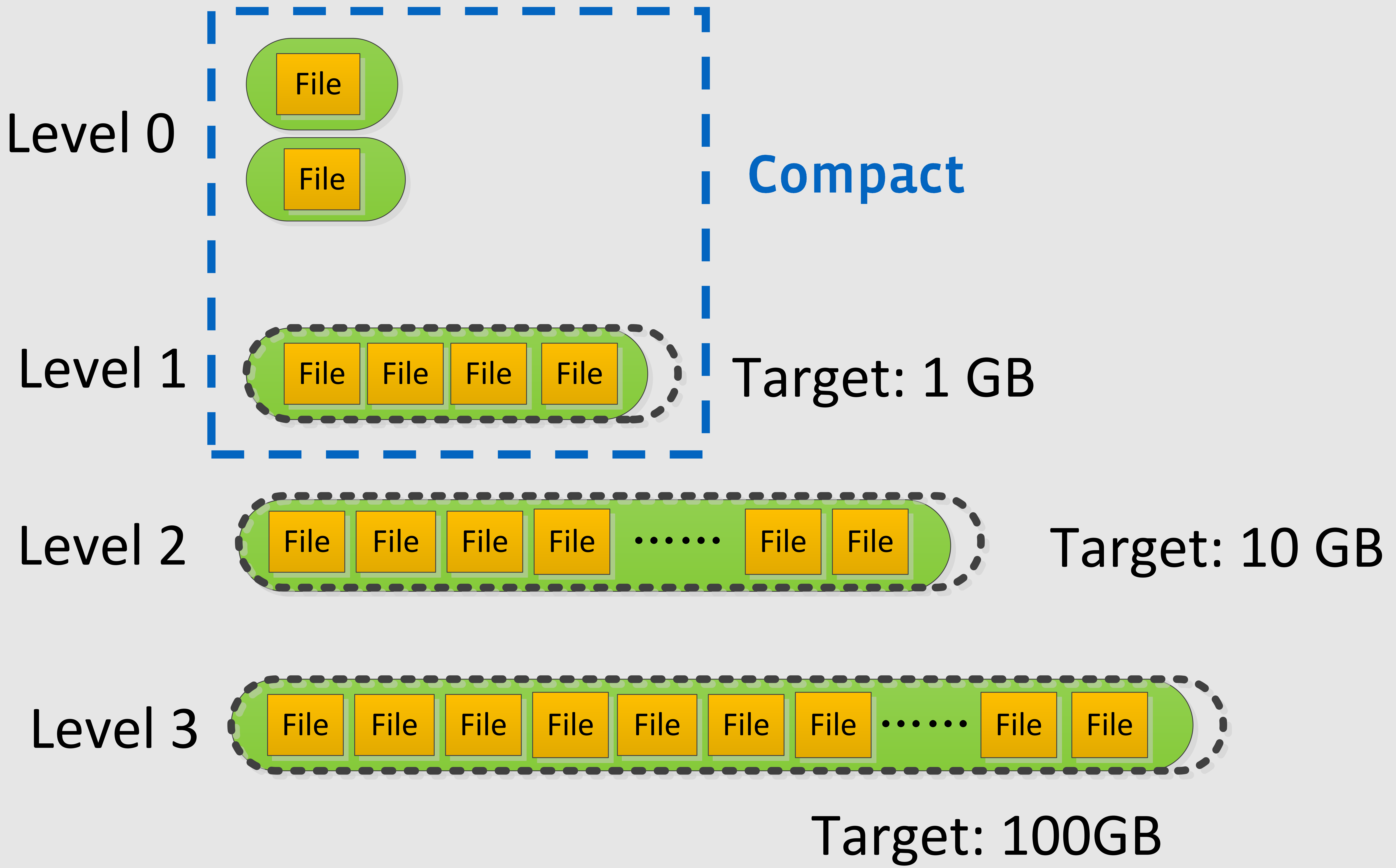


* These Numbers are just examples

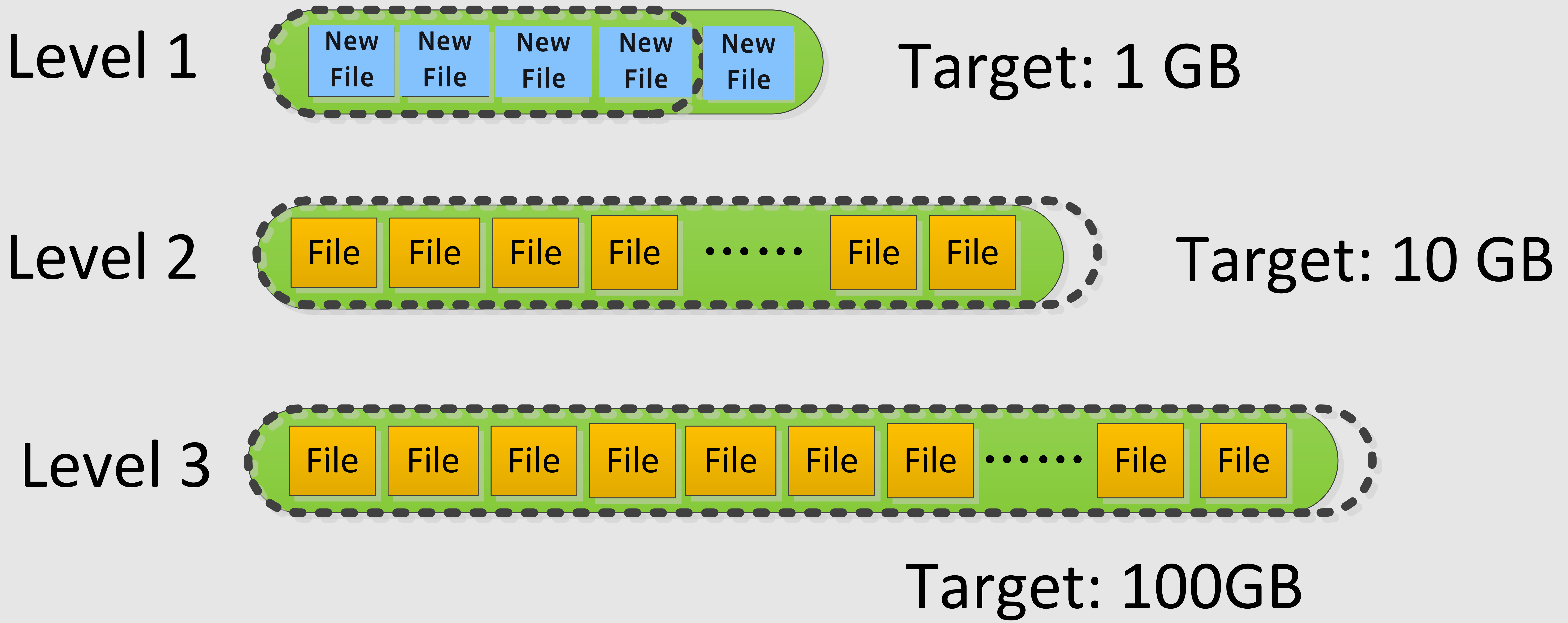
Level-Based Compaction



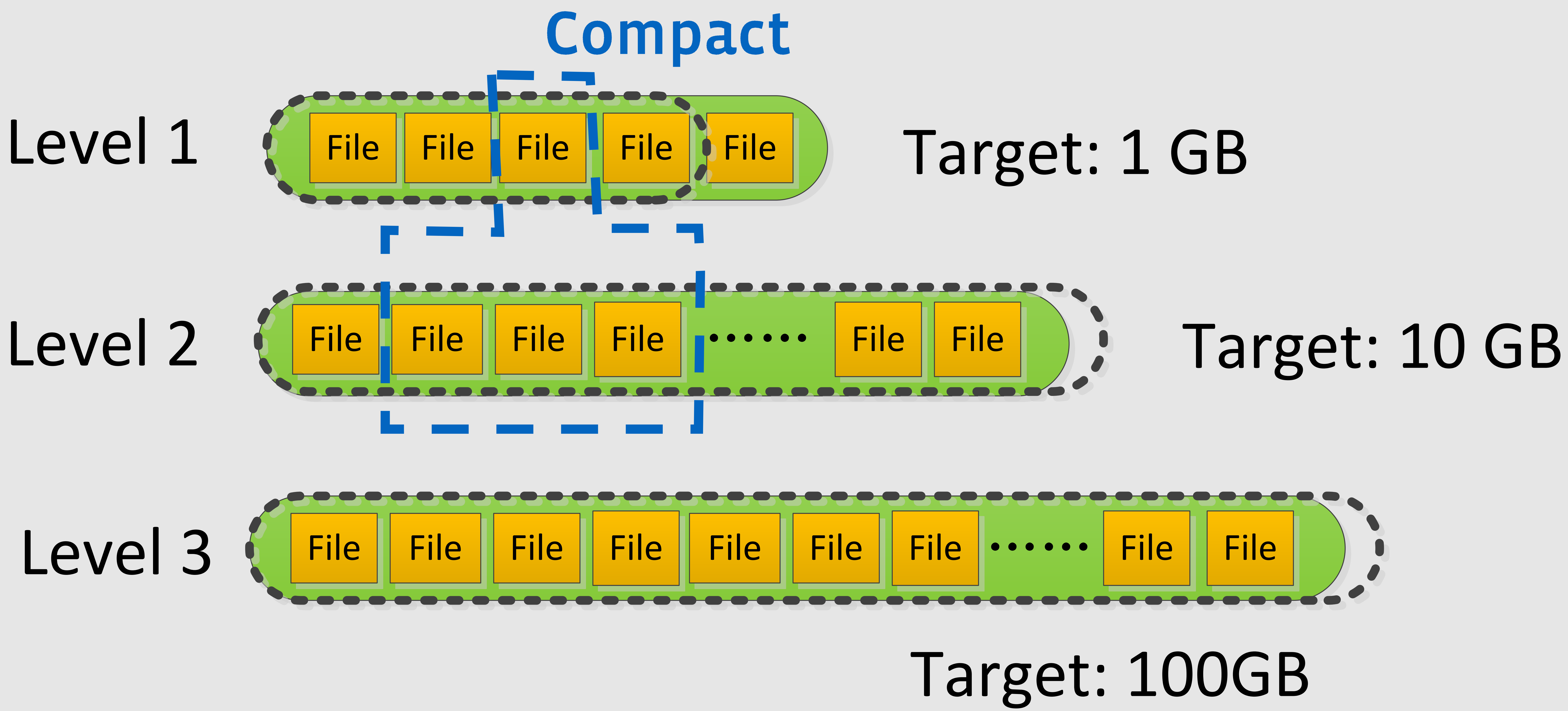
Level-Based Compaction



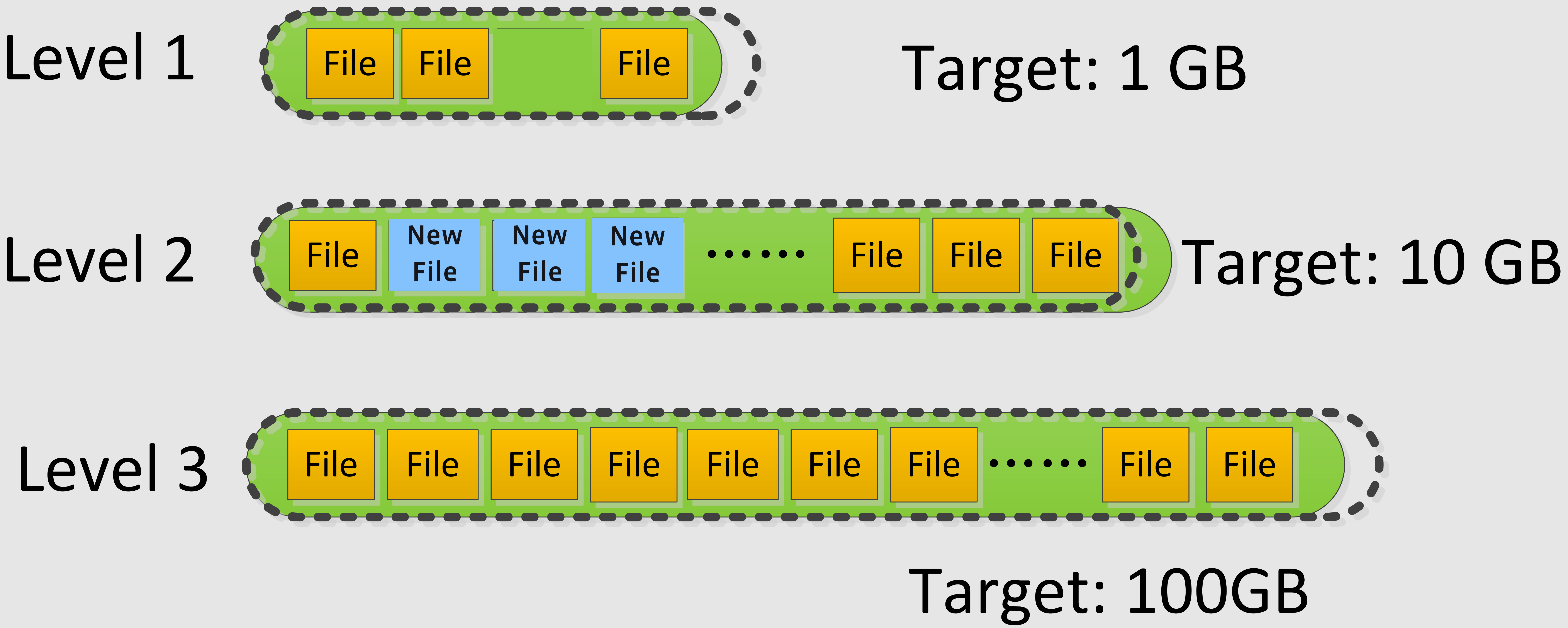
Level-Based Compaction



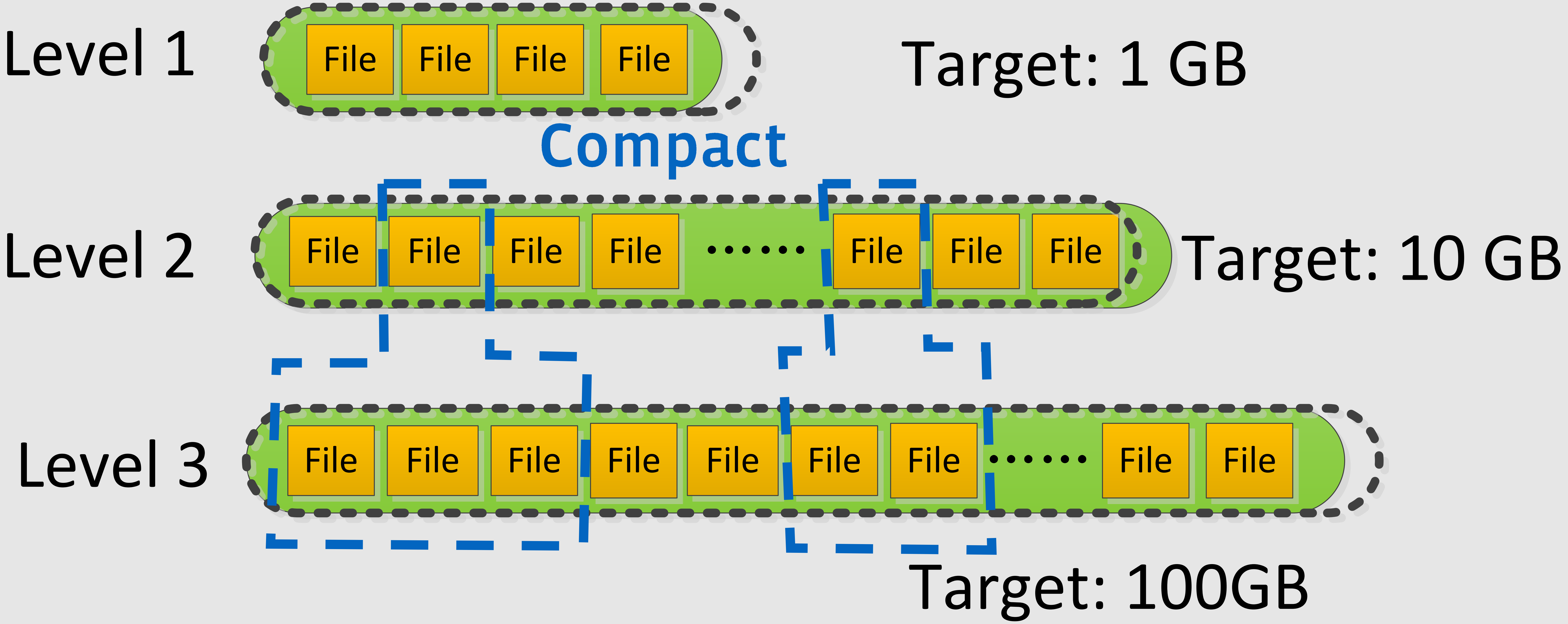
Level-Based Compaction



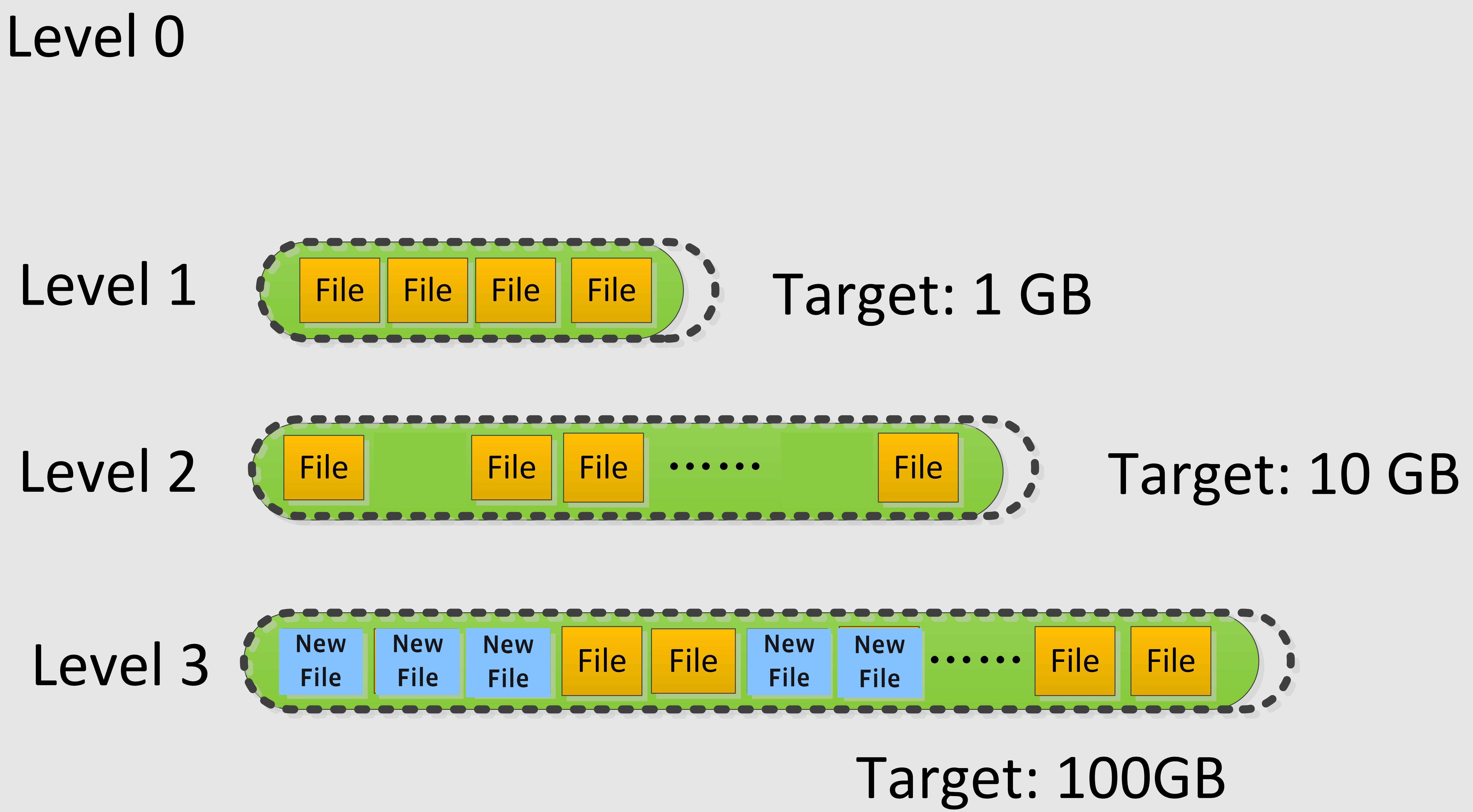
Level-Based Compaction



Level-Based Compaction



Level-Based Compaction

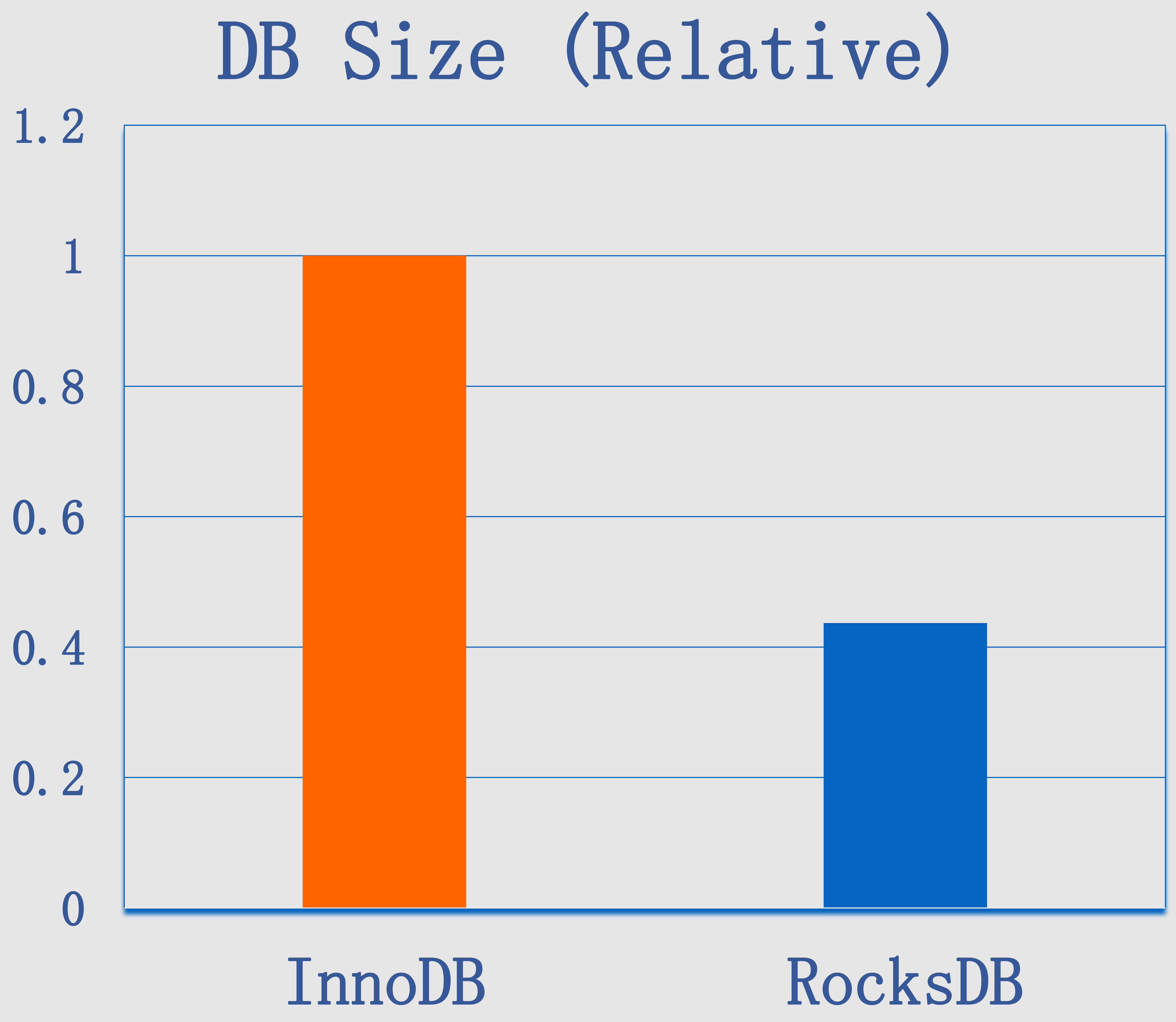


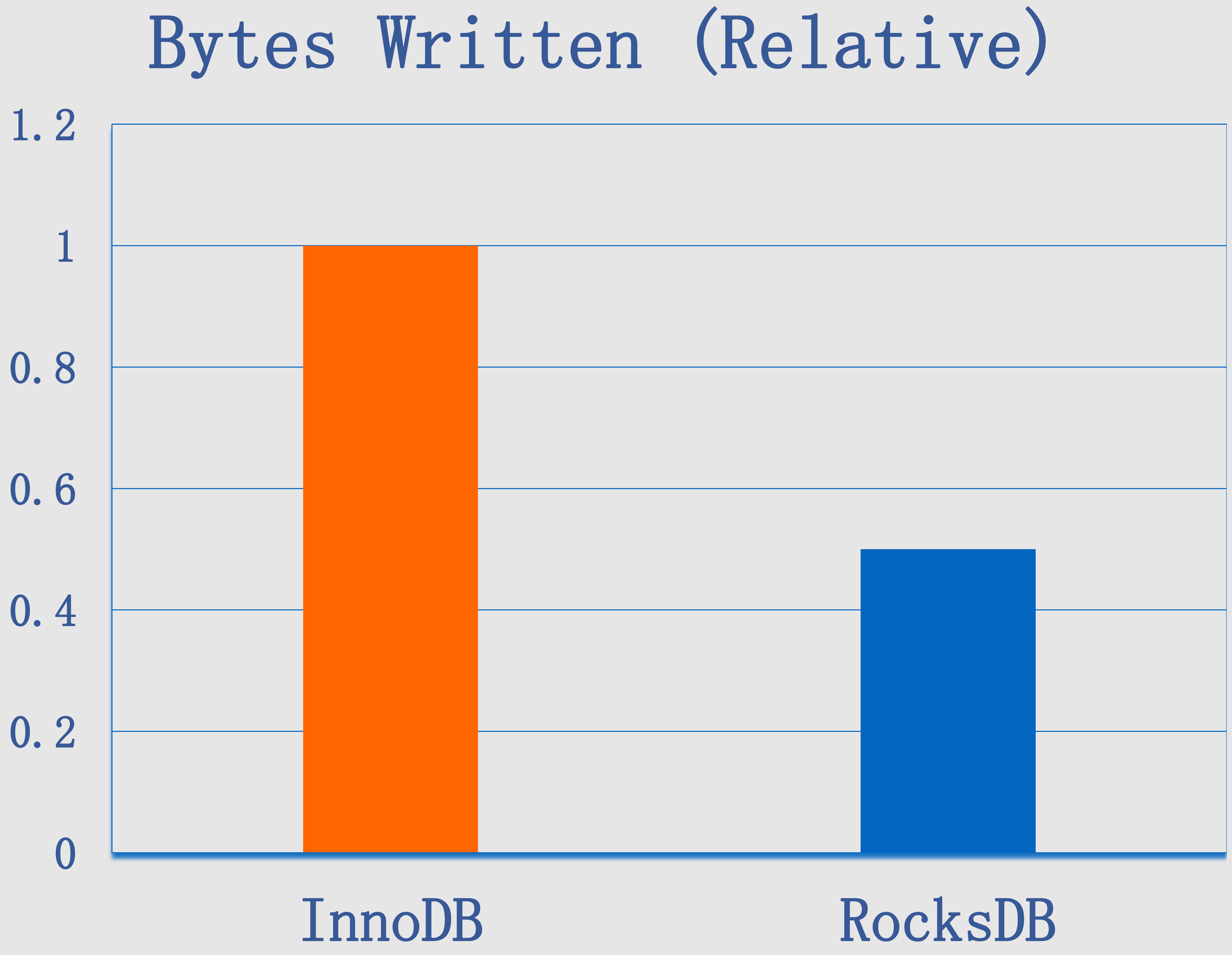
Why Log-Structure Merge Tree?

Measurements of MySQL+InnoDB Hosts' Actual Resource Usages

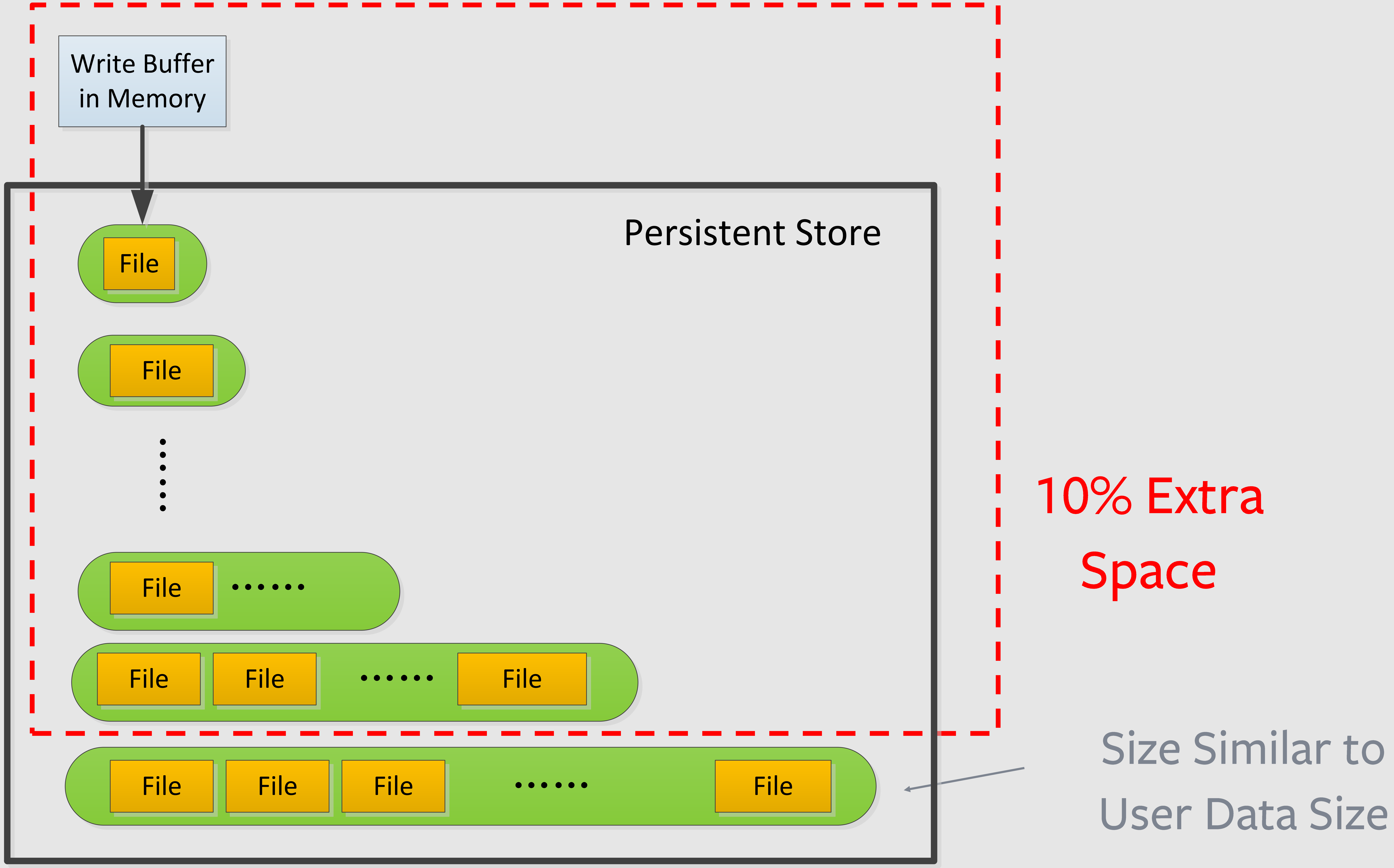
- Read IOPS: < 5%
- Write IOPS: < 5%
- Peak Write Bandwidth: < 15%
- CPU: < 20%
- Write Endurance: last more than 3 years.
- ***Space is the bottleneck***

MySQL + InnoDB vs MySQL + RocksDB





How do we estimate space efficiency in LSM?



Challenges

Challenges

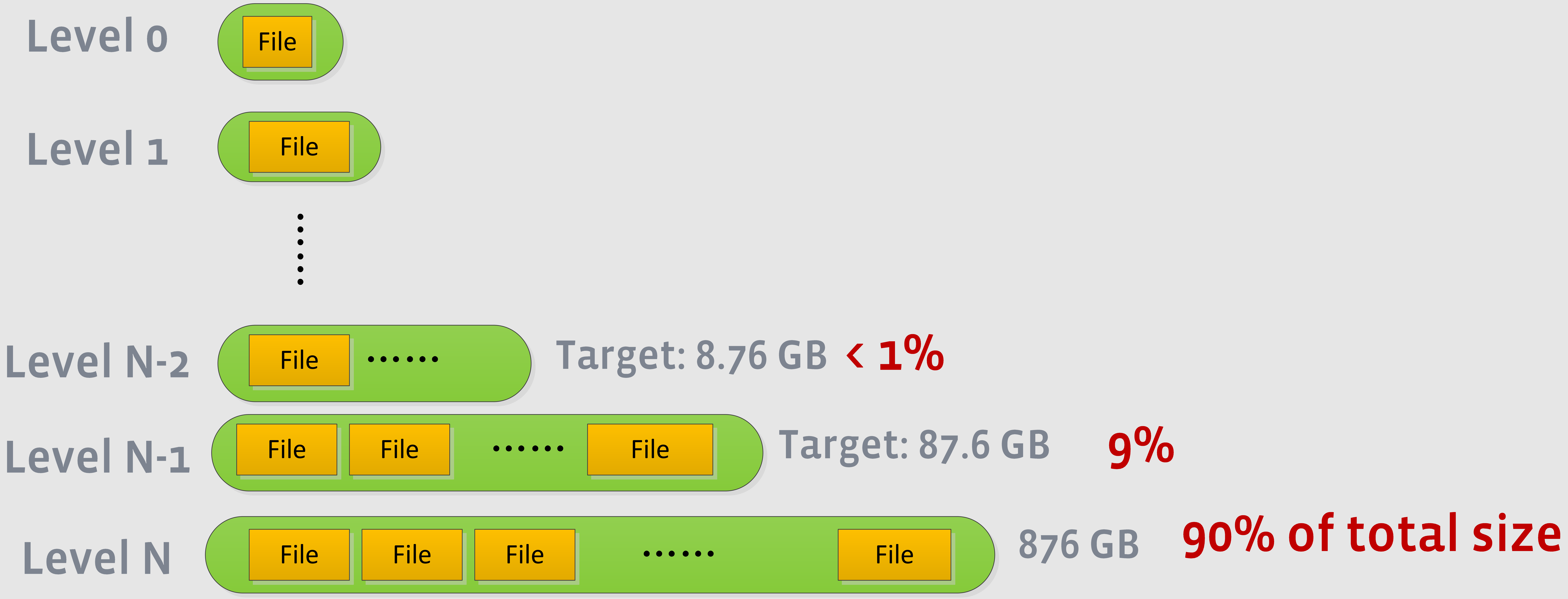
- How to Keep LSM structure in good shape
- Chunk tombstone problem
- Bloom filter for range queries

Challenge 1:
How to Keep LSM Tree in Good Shape?

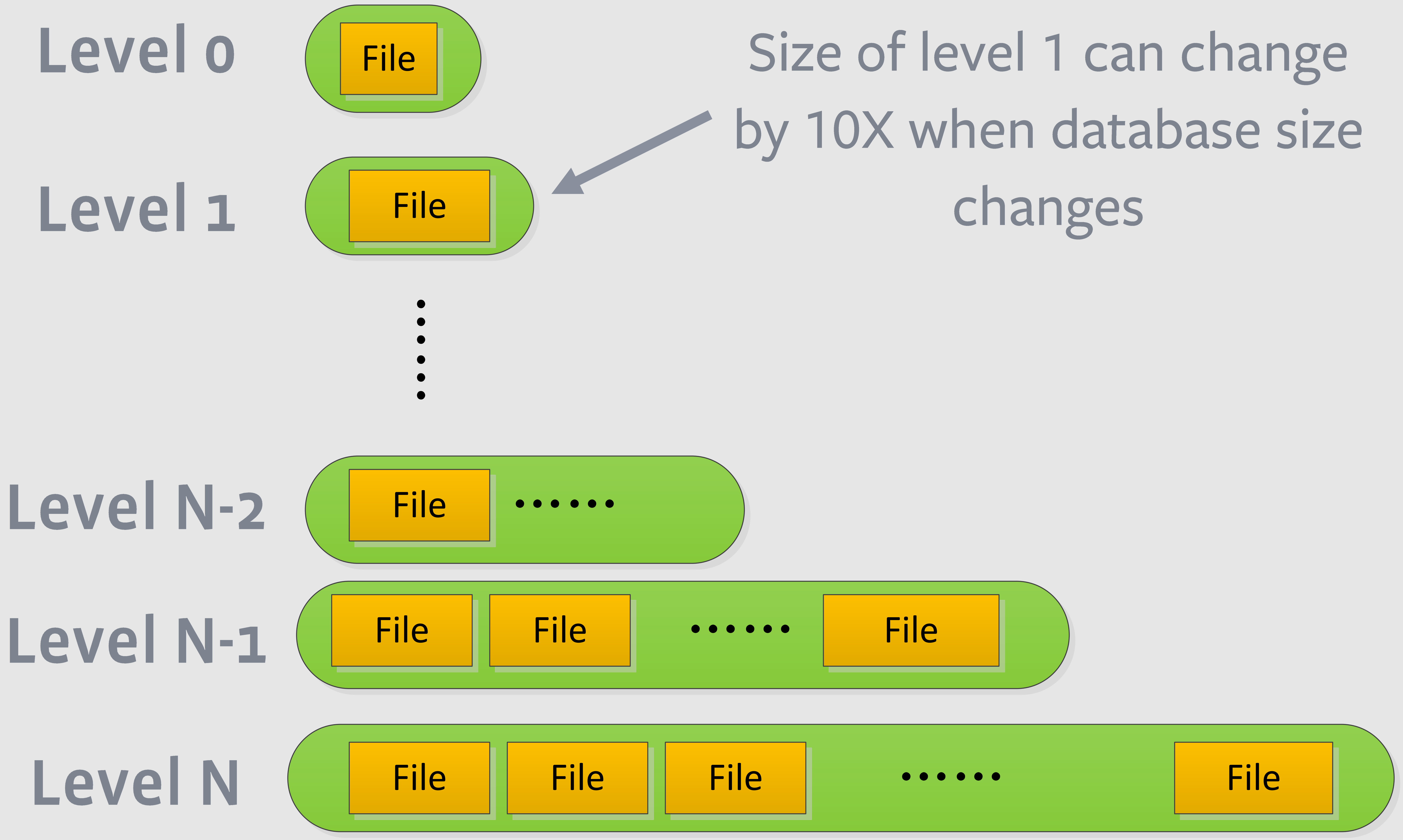
Importance of Keeping LSM Tree In Good Shape

- Worse Case Space Efficiency
- Memory Caching

Our Solution: Dynamic Level Size Target

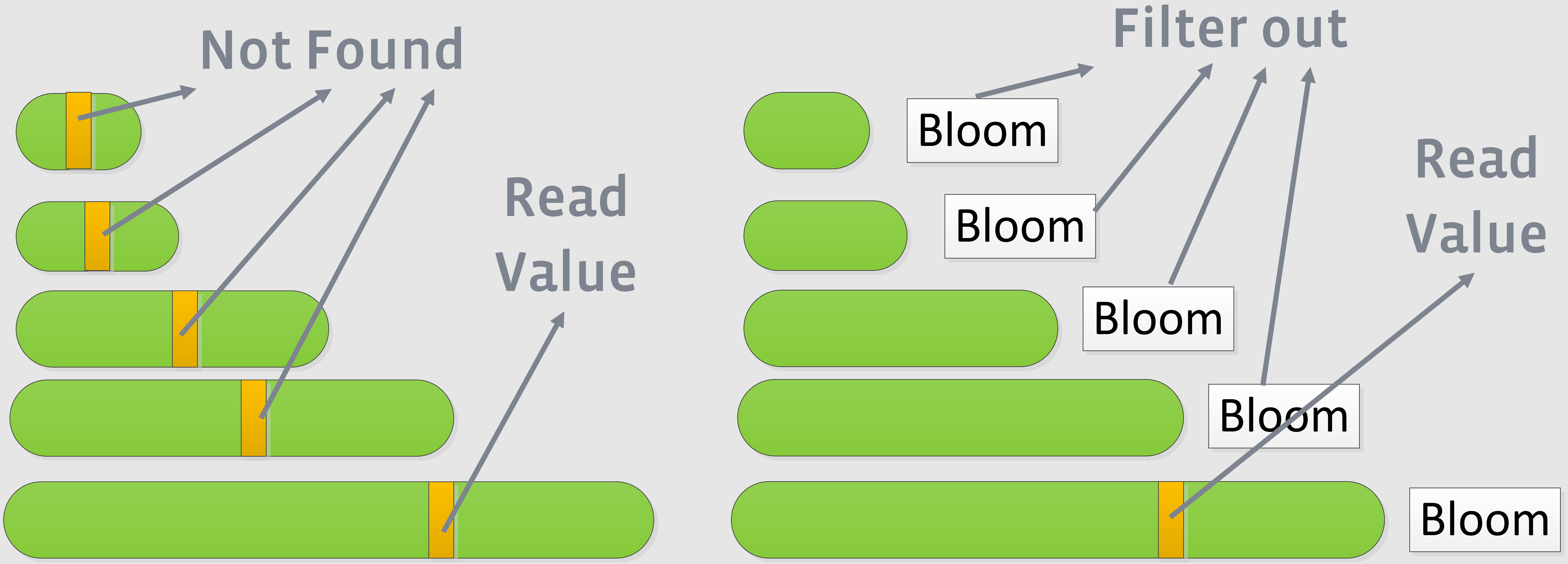


Can we do better?



Challenge 2:
Bloom Filter for Range Queries

Bloom Filter



Bloom Filter cannot be used in Range Queries

Keys
……
Apple2013
Apple2015
Banana2012
Lemon2012
Lemon2013
Lemon2014
……

- Get (“Cherry2013”) can use bloom filter
- Range lookup [Cherry2000, Cherry2015] cannot use bloom filter.

Trick: prefix bloom

Keys
......
Apple2013
Apple2015
Banana2012
Lemon2012
Lemon2013
Lemon2014
......

- Define fruit part as "prefix"
- Can use bloom filter in range query: [cherry2010, cherry2015]

Open question: can we do better?

Keys
......
Apple2013
Apple2015
Banana2012
Lemon2012
Lemon2013
Lemon2014
......

- Fruit part as “prefix”
- Boom filter not useful for range: [Banana2014, Banana2015]
- Bloom filter not useful for range: [C, H]

Challenge 3:
Chunk of Tombstones

Deletions in LSM

Why do we have tombstones?

..., Put(1), Put(2), Put(3), Put(4), Put(5), Put(6), ...

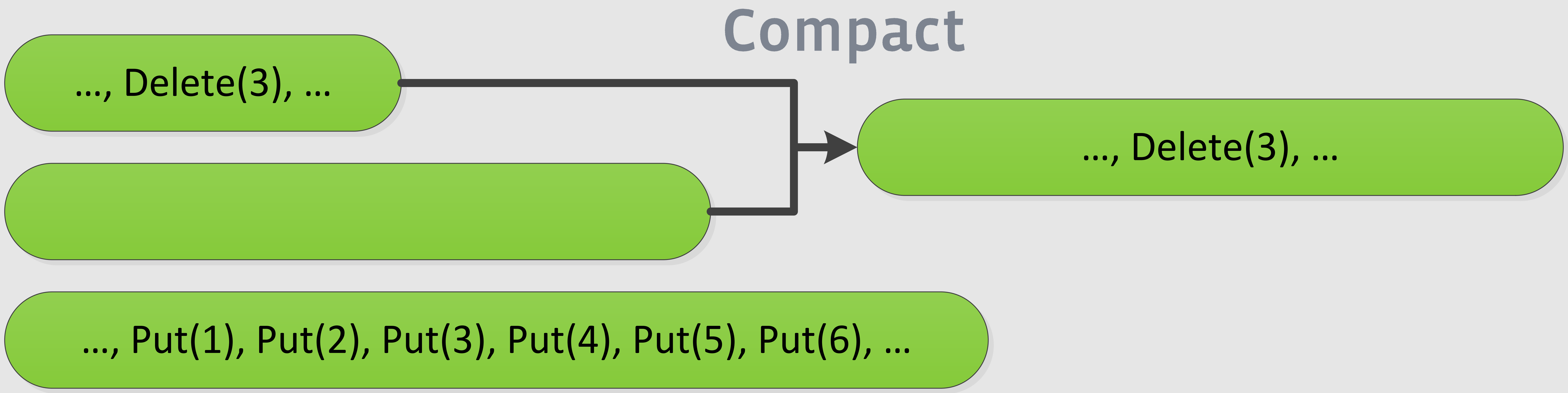
Deletions in LSM

Why do we have tombstones?

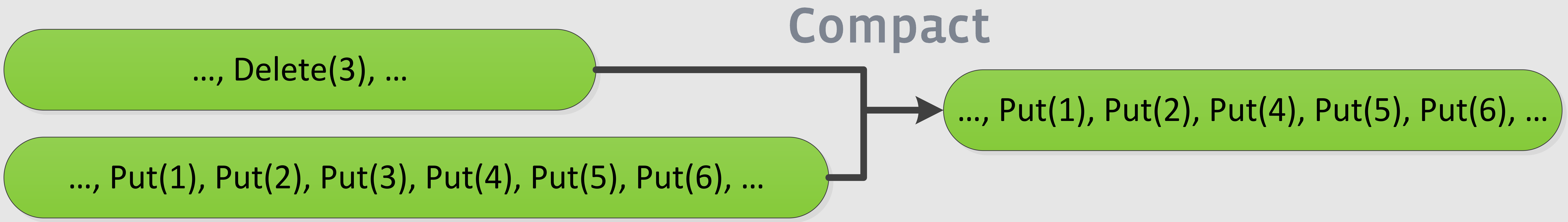
…, Delete(3), …

…, Put(1), Put(2), Put(3), Put(4), Put(5), Put(6), …

When do we clear tombstones?



What is tombstone?



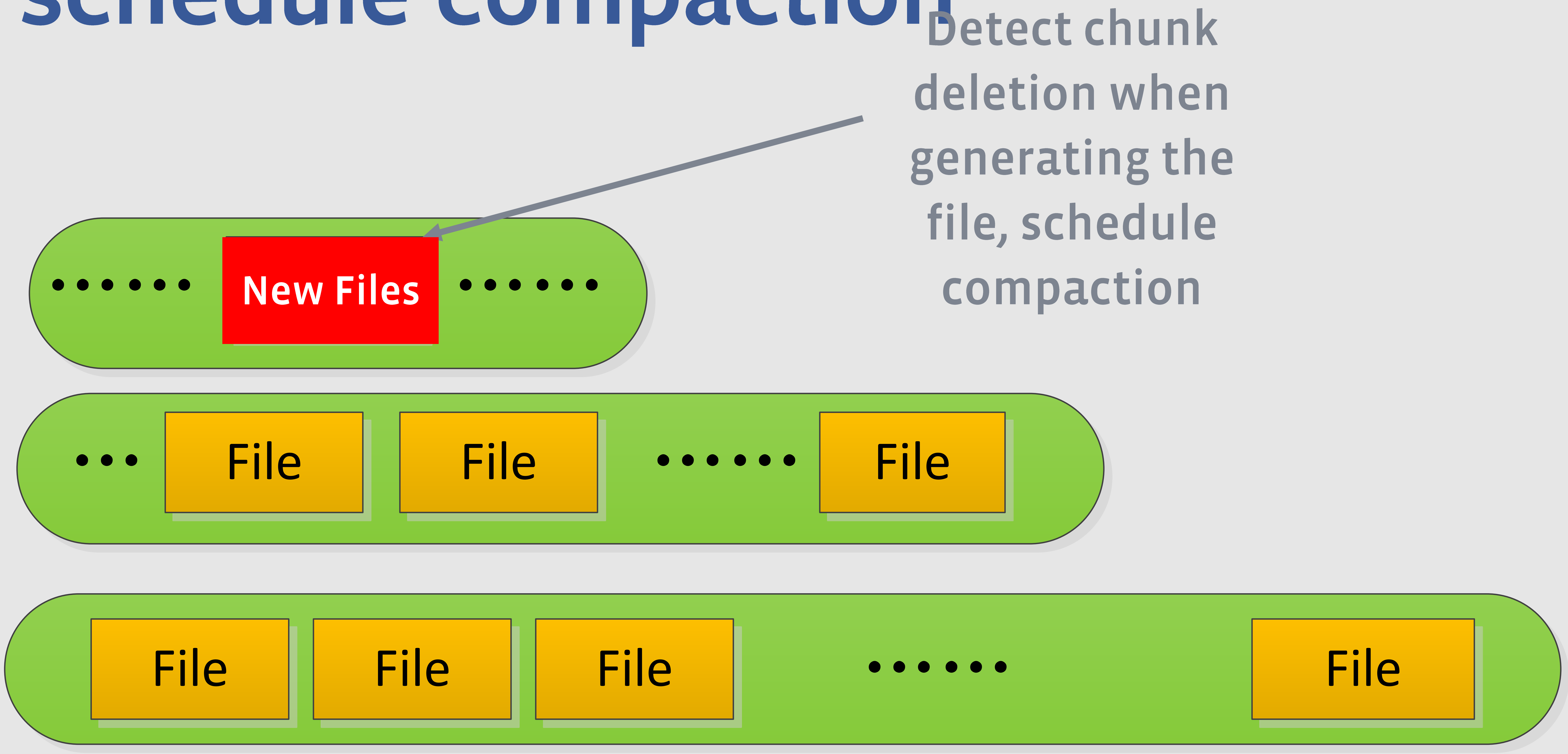
Problem of Chunk Tombstones

…, Delete(1000), Delete(1001), …, Delete(1999), …

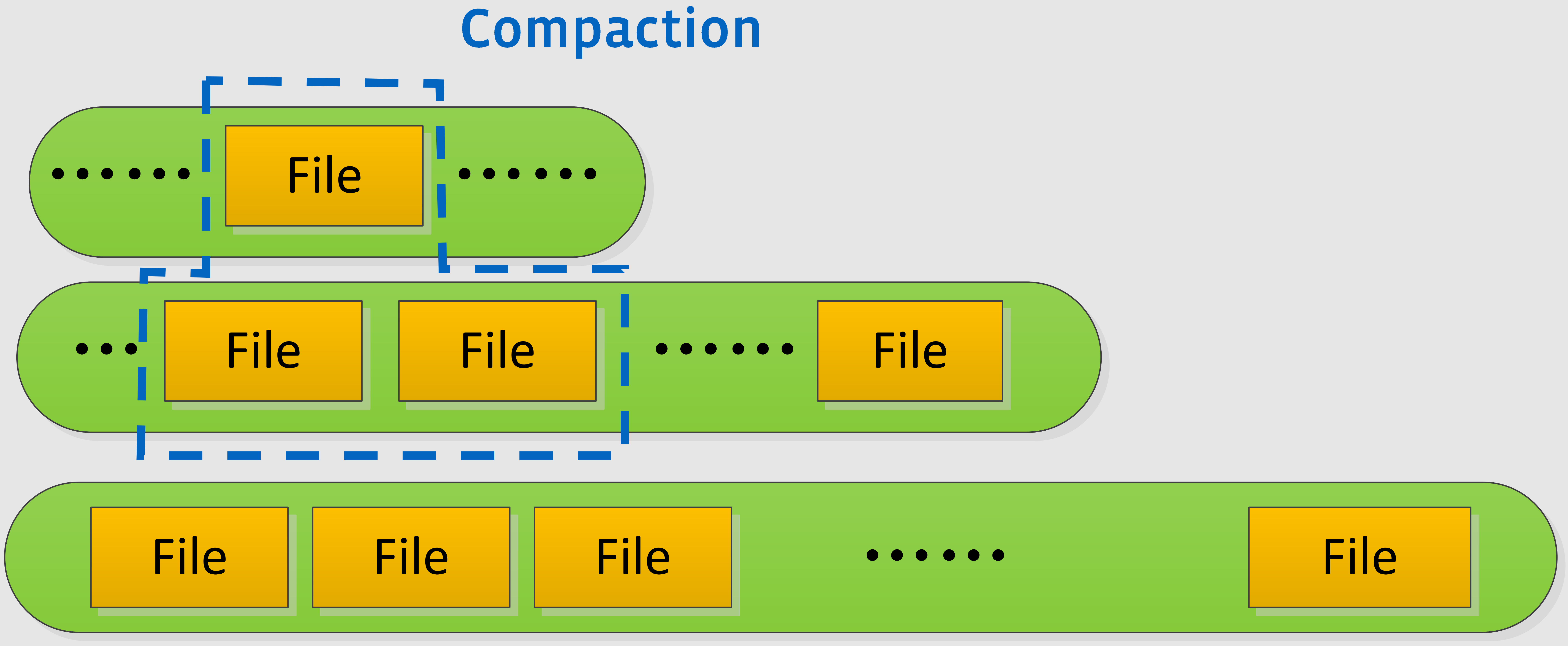
…, Put(999), Put(1000), Put(1001), Put(1002) …, Put(1999), Put(2000), …

- Range Query (1000, 2000)
  needs go through 2000 keys internally.

Our Trick: detect chunk tombstones and schedule compaction

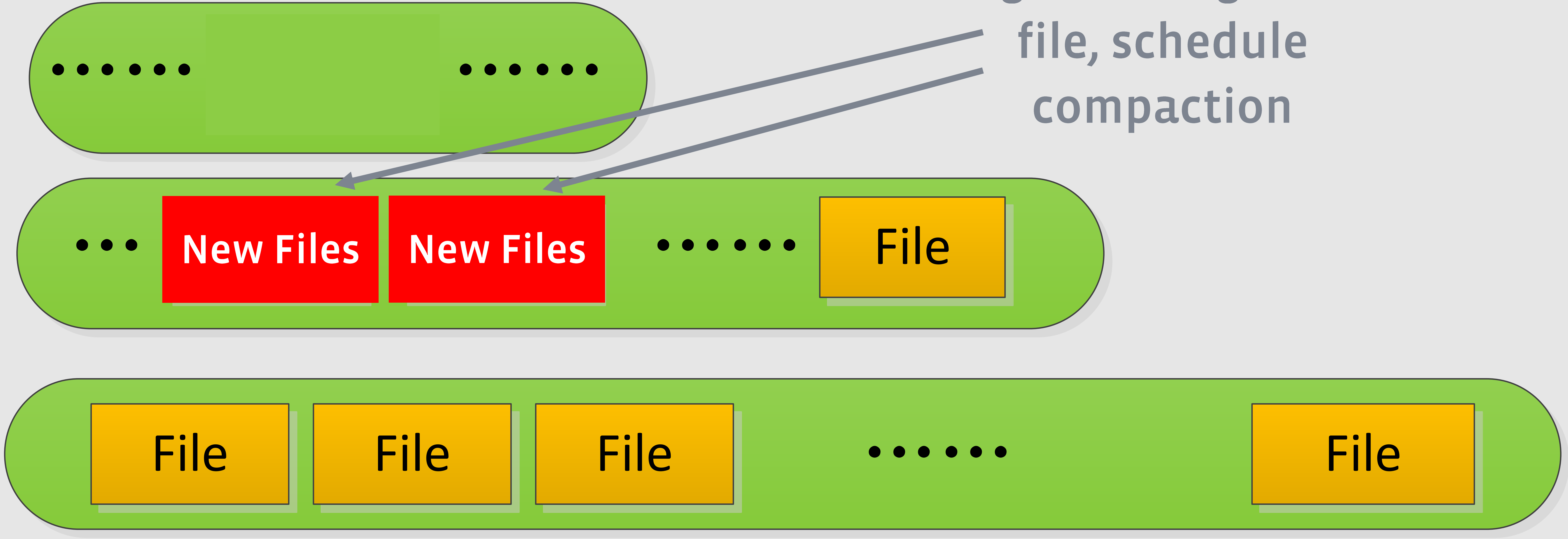


Our Trick: detect chunk tombstone and schedule compaction

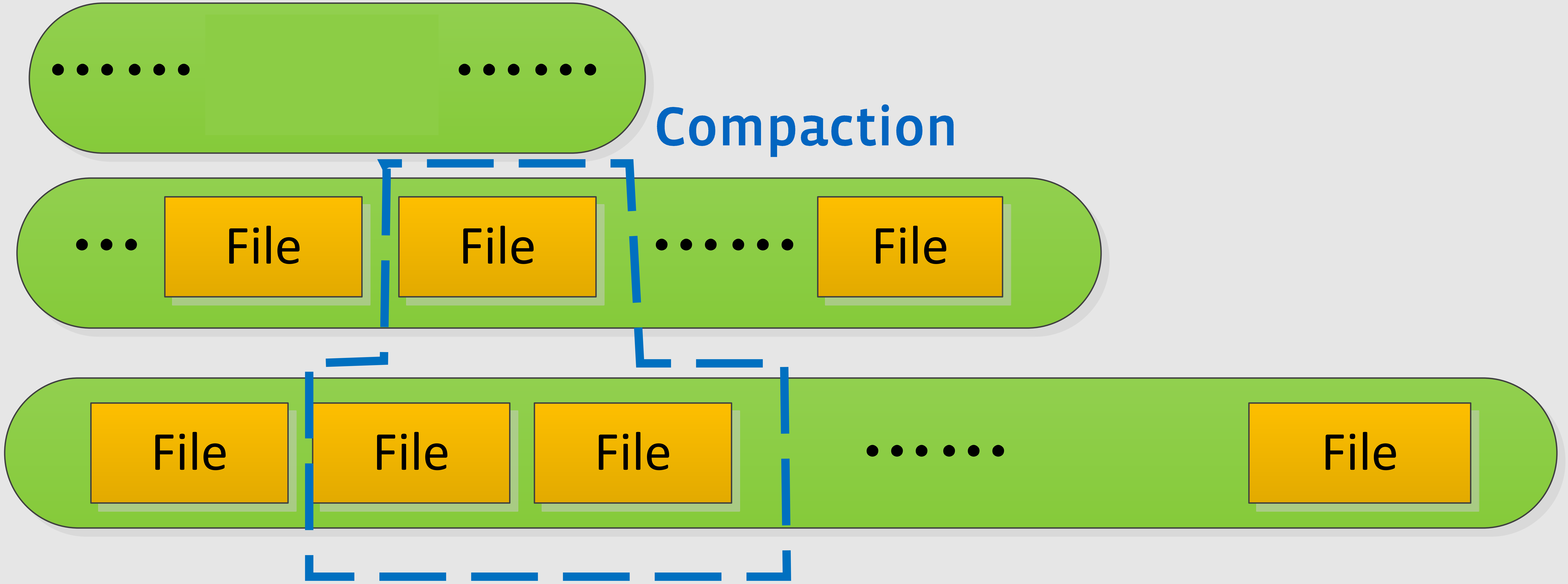


Our Trick: detect chunk deletion and schedule compaction

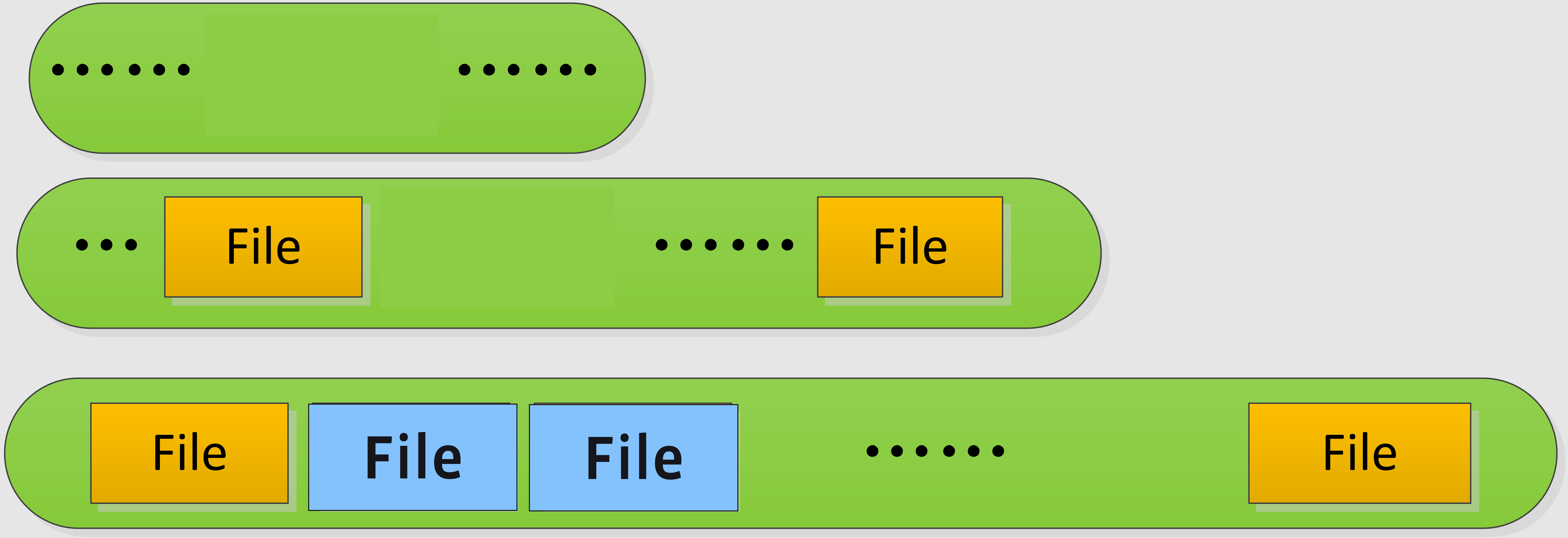
Detect chunk deletion when generating the file, schedule compaction



Our Trick: detect chunk tombstones and schedule compaction



Our Trick: detect chunk deletion and schedule compaction



Remaining Challenges

- Introduce more disk writes for some workloads
- Slow before compaction finishes
- Tombstones in mem tables.

Recap

- RocksDB Uses Log-Structure-Merge tree for space efficiency
- Keep LSM structure in good shape
- Chunk tombstone detection
- Prefix Bloom filter for range queries

Open Challenges

http://rocksdb.org

facebook