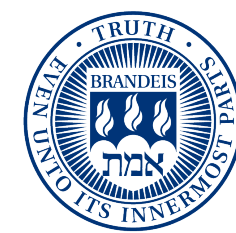
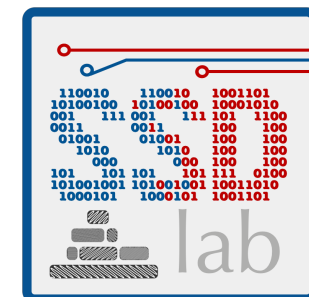


RangeReduce: Query-Driven LSM Compactions

Shubham Kaushik

Manos Athanassoulis

Subhadeep Sarkar



Brandeis
UNIVERSITY





CockroachDB



SQLite



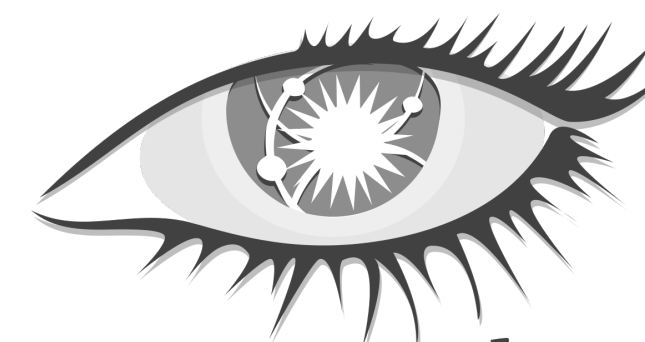
FoundationDB



levelDB



RocksDB



cassandra



APACHE HBASE



influxdb



OpenTSDB

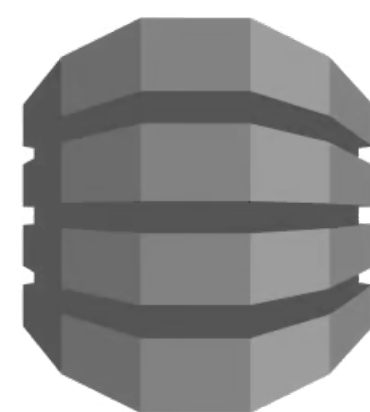
Key-Value Stores



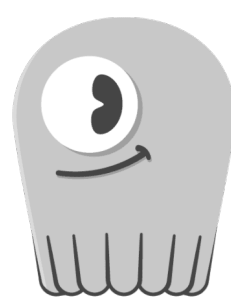
BigTable



Azure Cosmos DB



DynamoDB



ScyllaDB

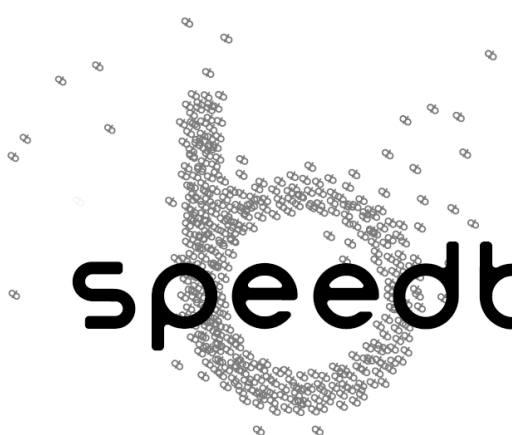


TiKV



mongoDB

WT



speedb

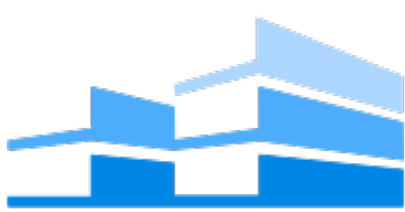


CockroachDB



SQLite

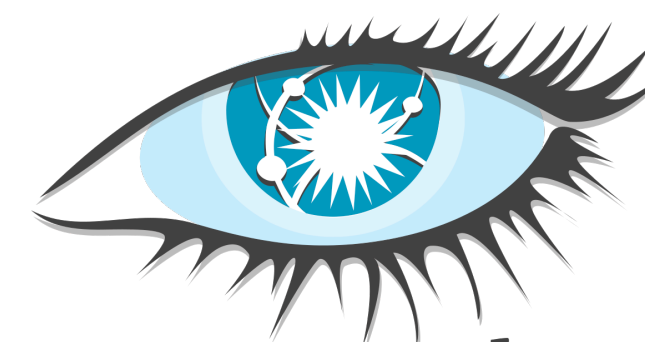
FoundationDB



levelDB



RocksDB



cassandra



APACHE HBASE



influxdb



OpenTSDB

Key-Value Stores



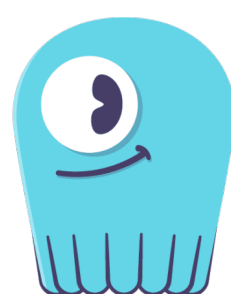
BigTable



Azure Cosmos DB



DynamoDB



ScyllaDB

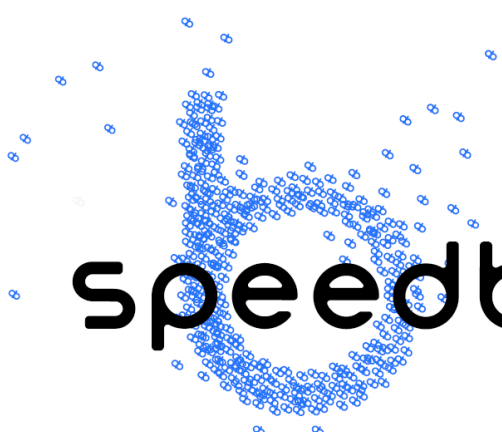


TiKV



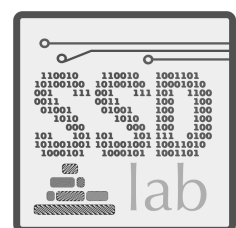
mongoDB

WT

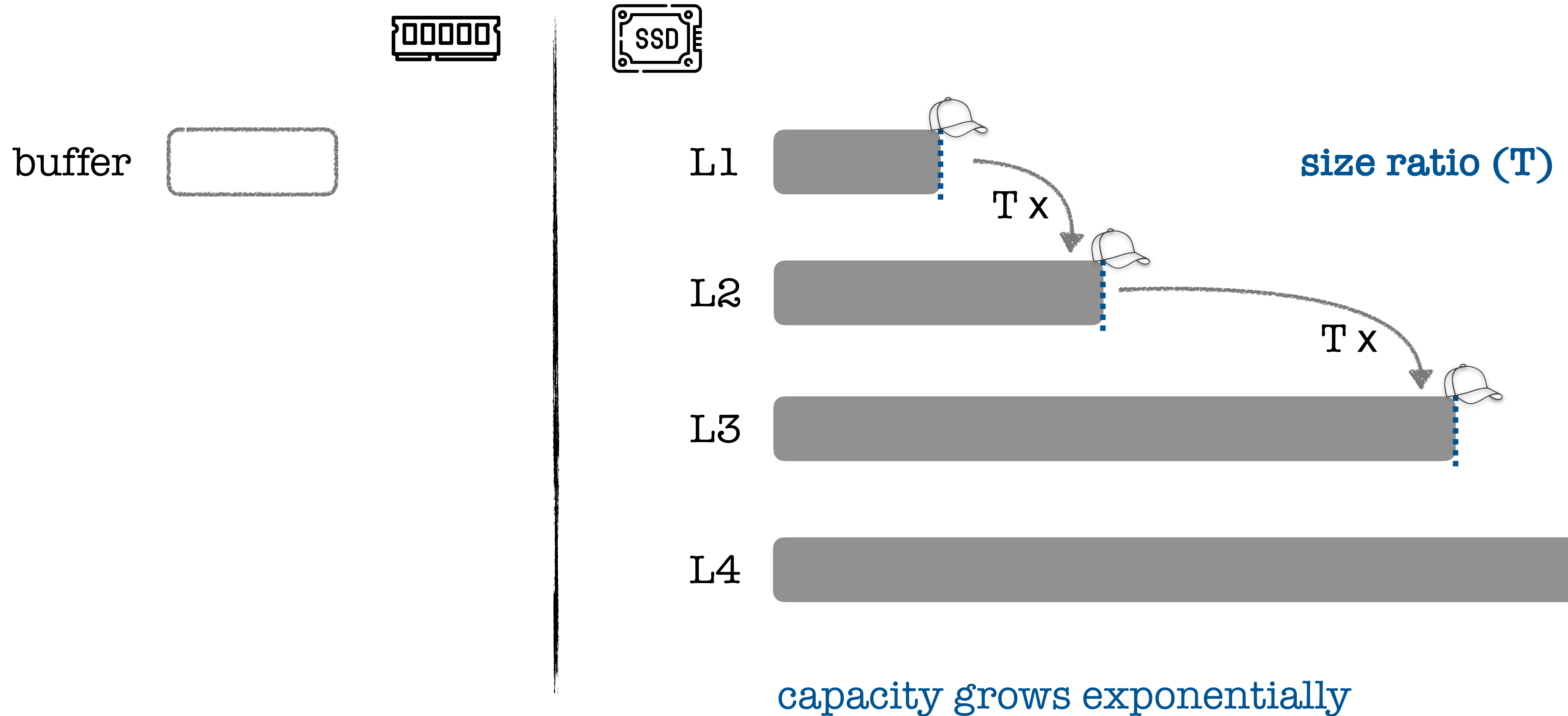


speedb

Log-Structured Merge Trees

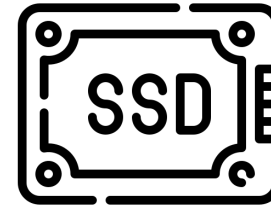
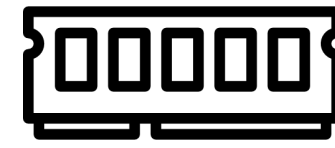
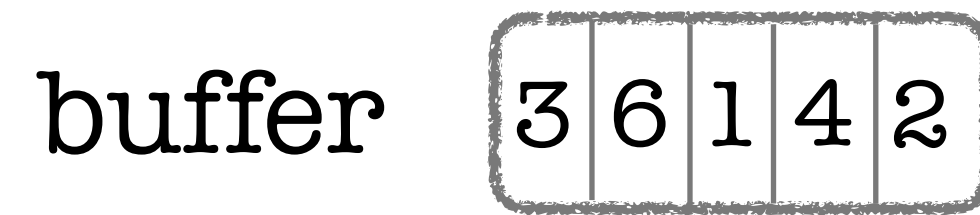


LSM Overview

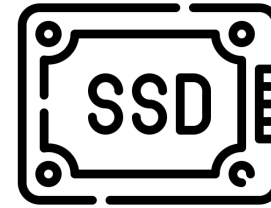
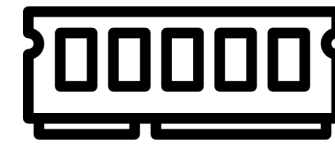
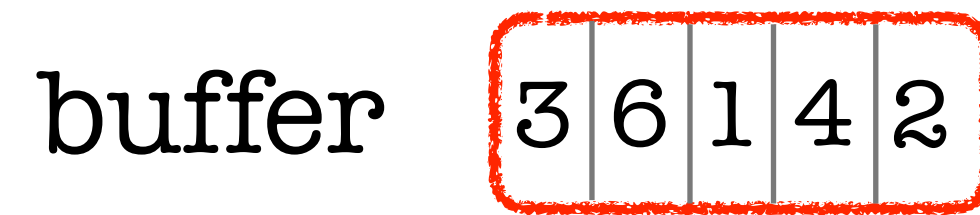


capacity grows exponentially

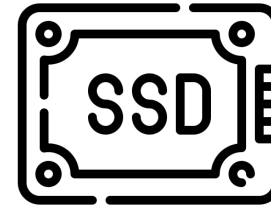
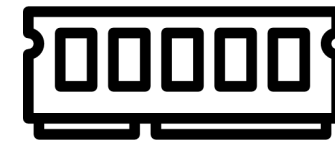
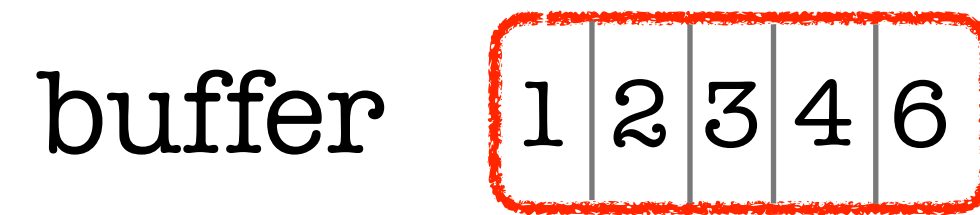
LSM Ingestion



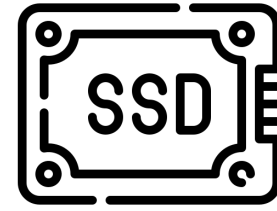
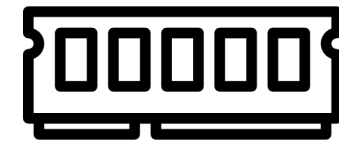
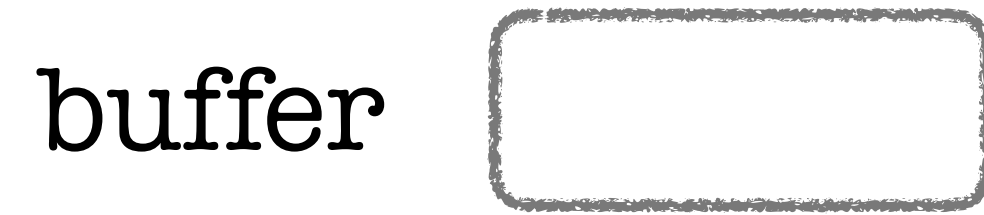
LSM Ingestion



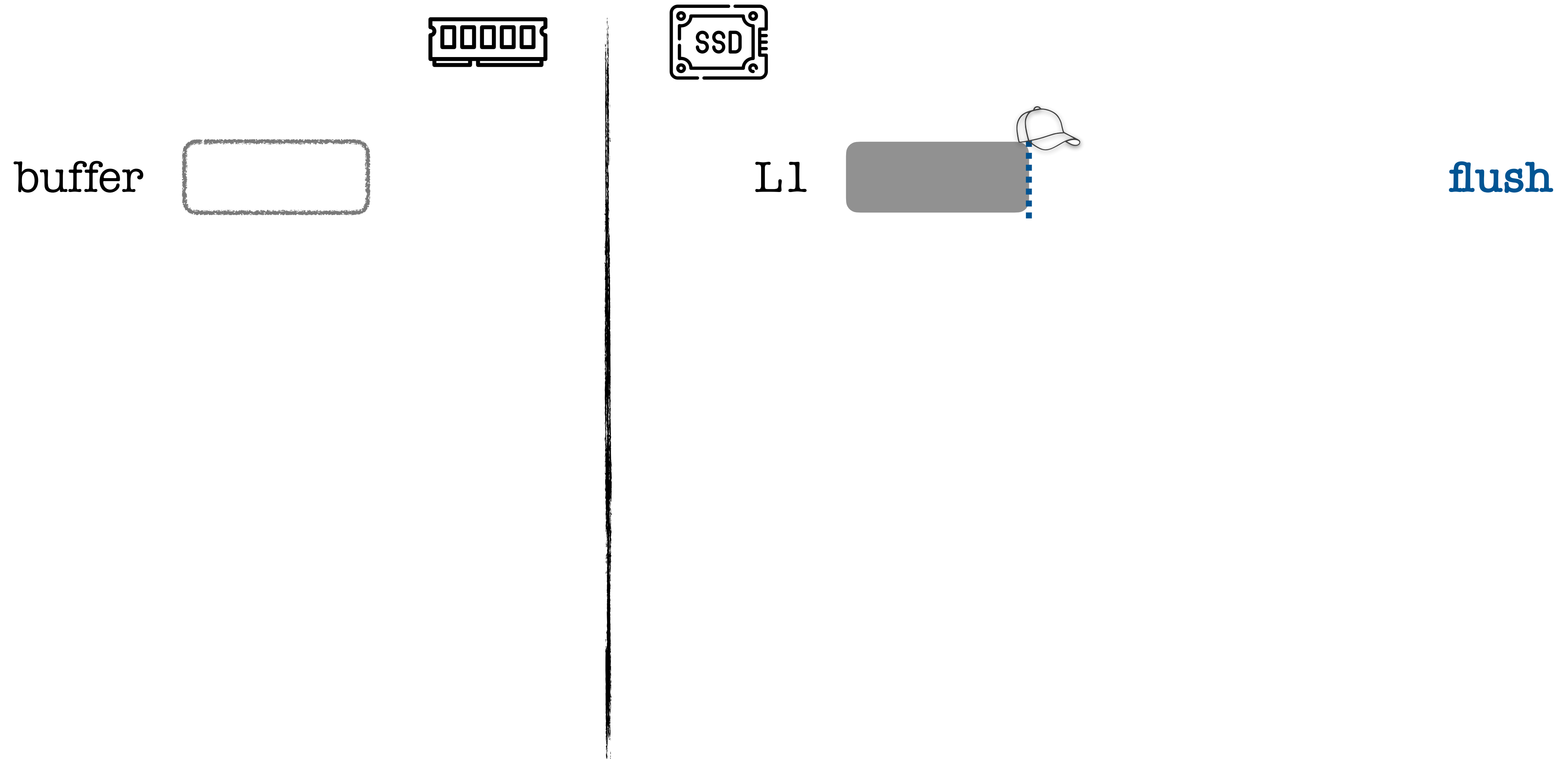
LSM Ingestion



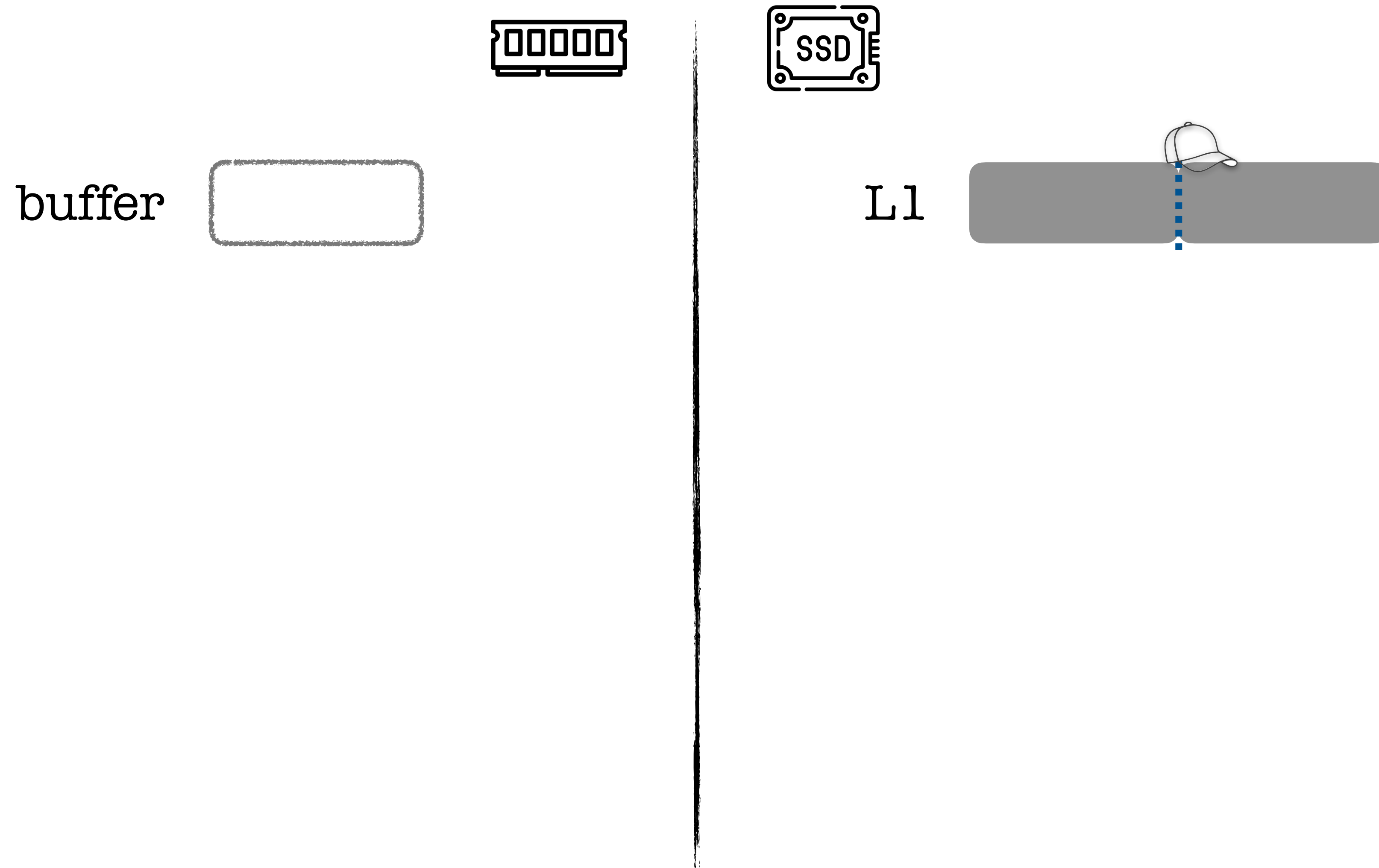
LSM Ingestion



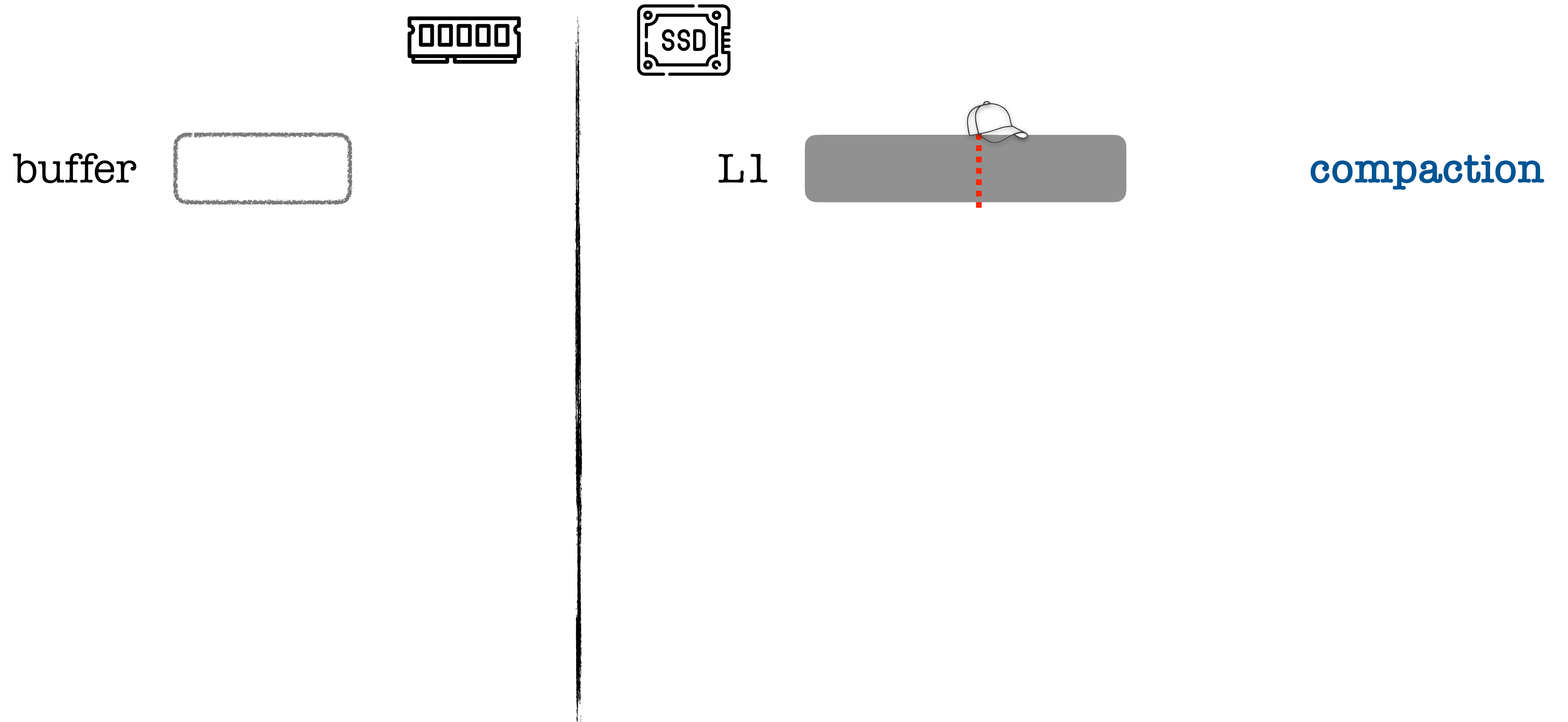
LSM Ingestion



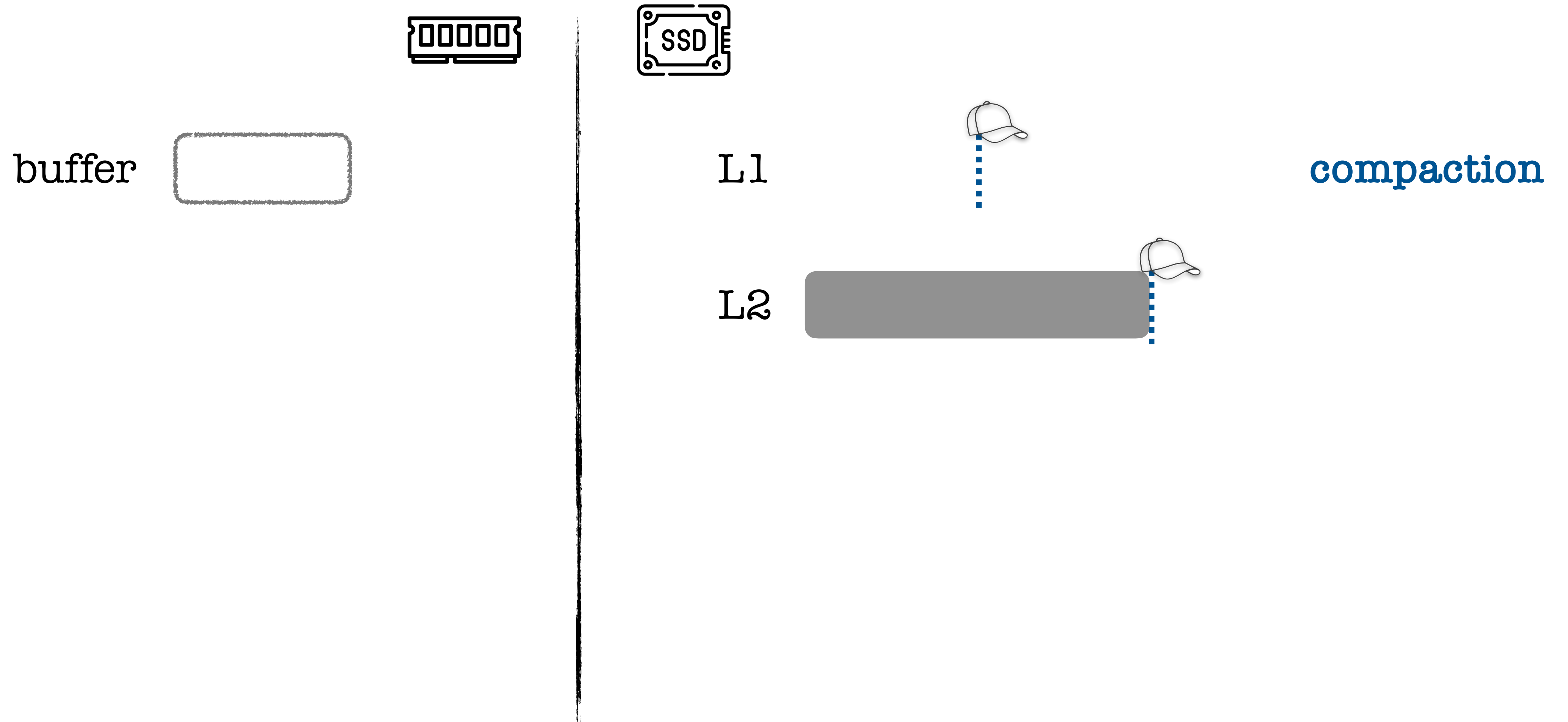
LSM Ingestion



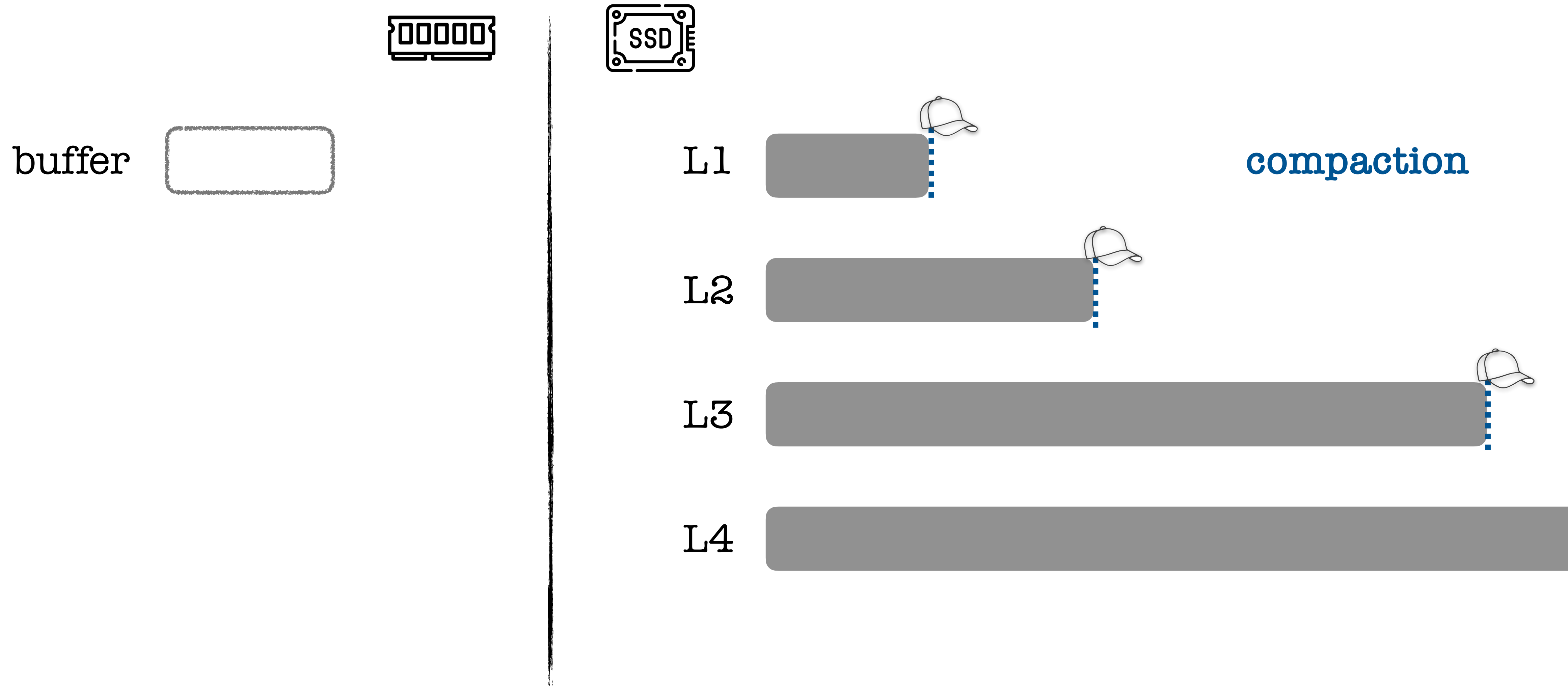
LSM Compaction



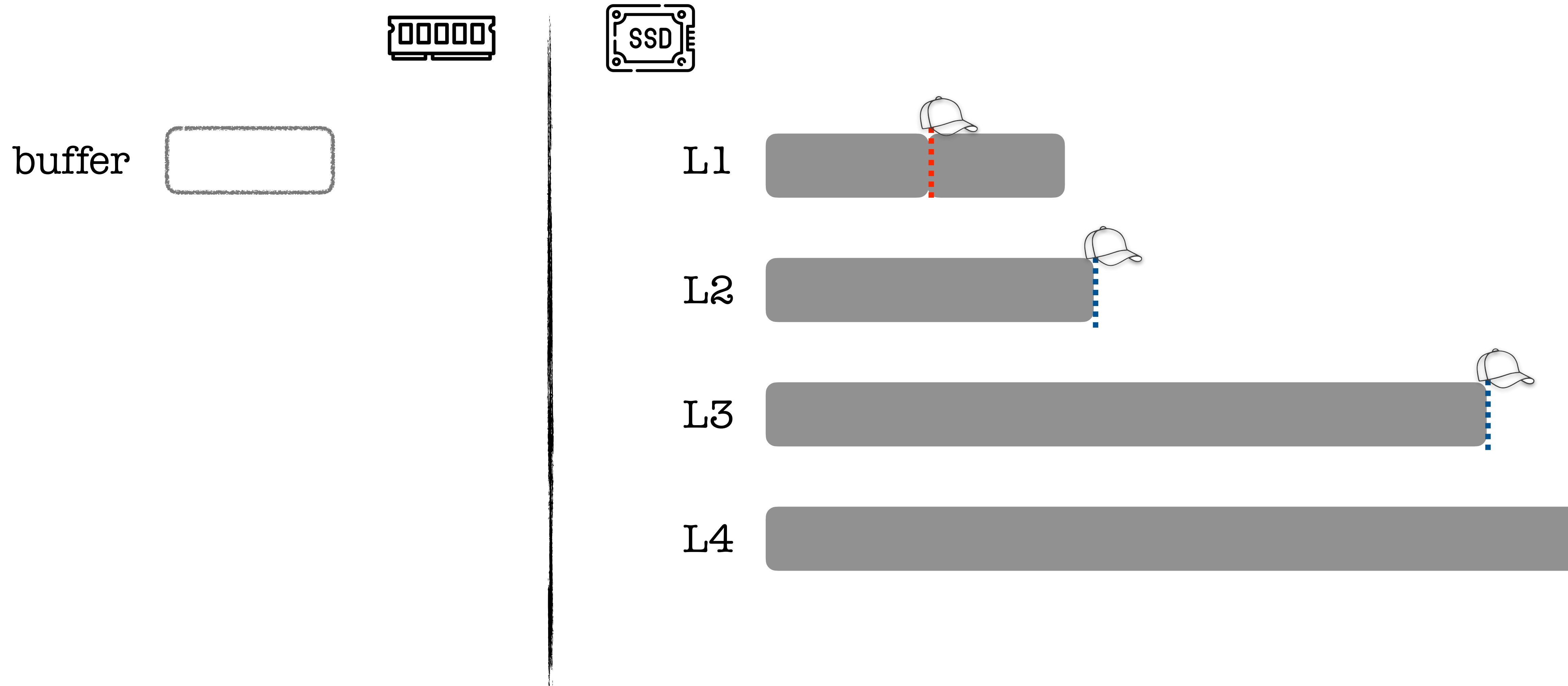
LSM Compaction



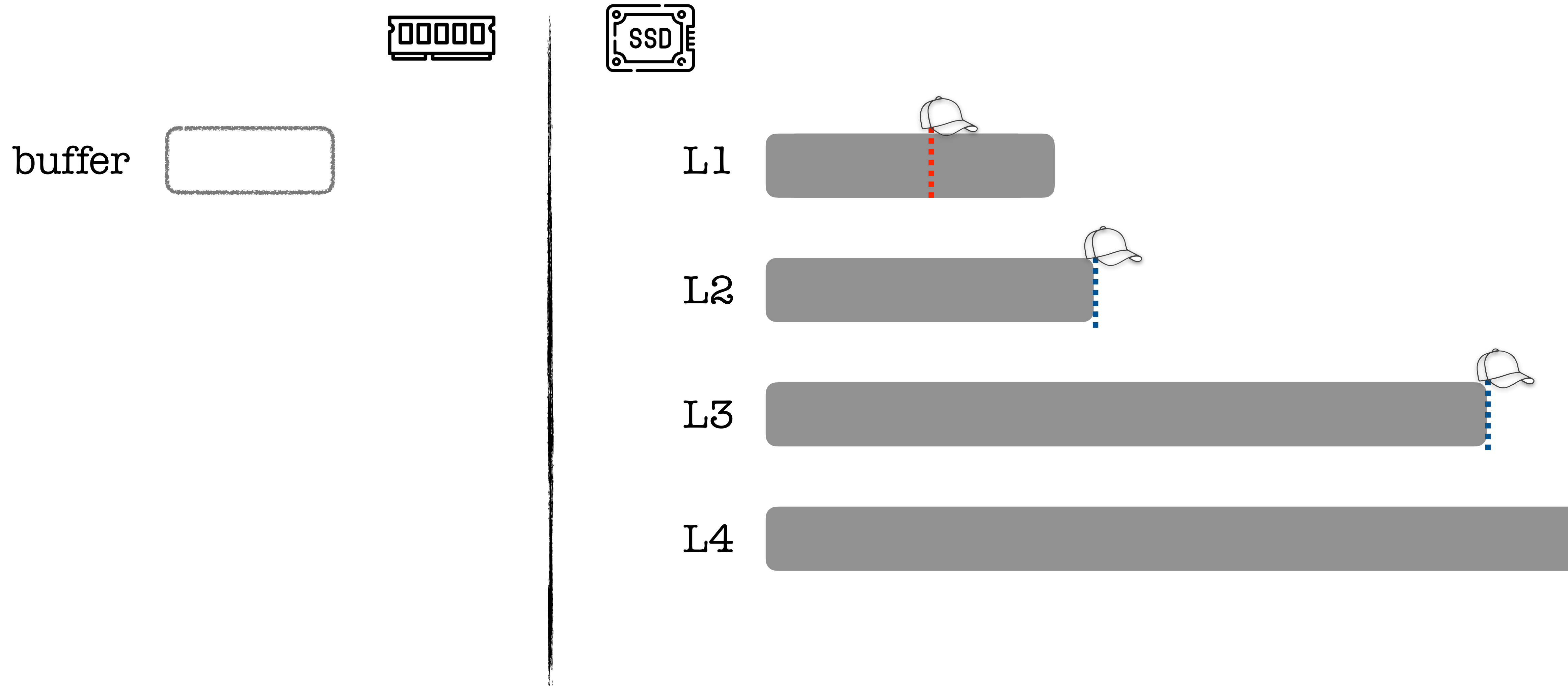
LSM Compaction



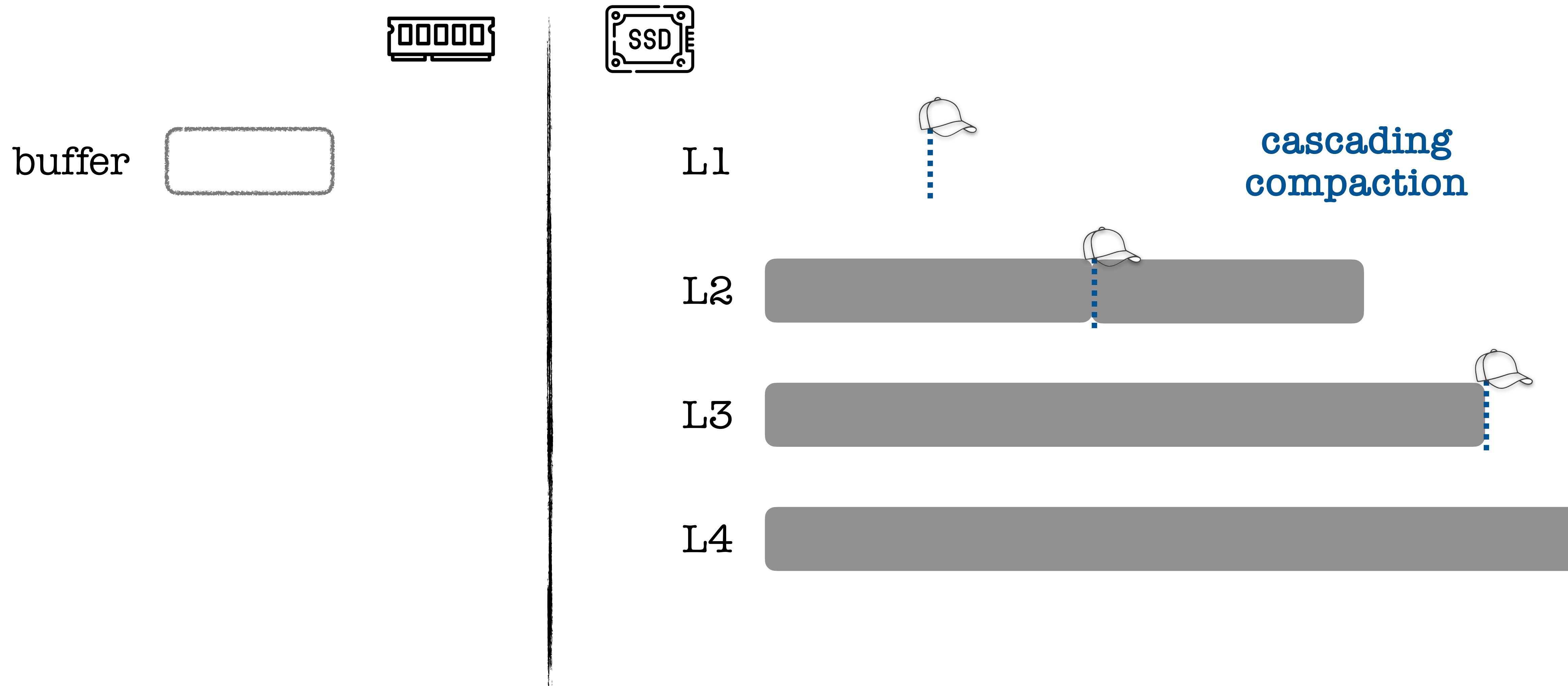
LSM Compaction



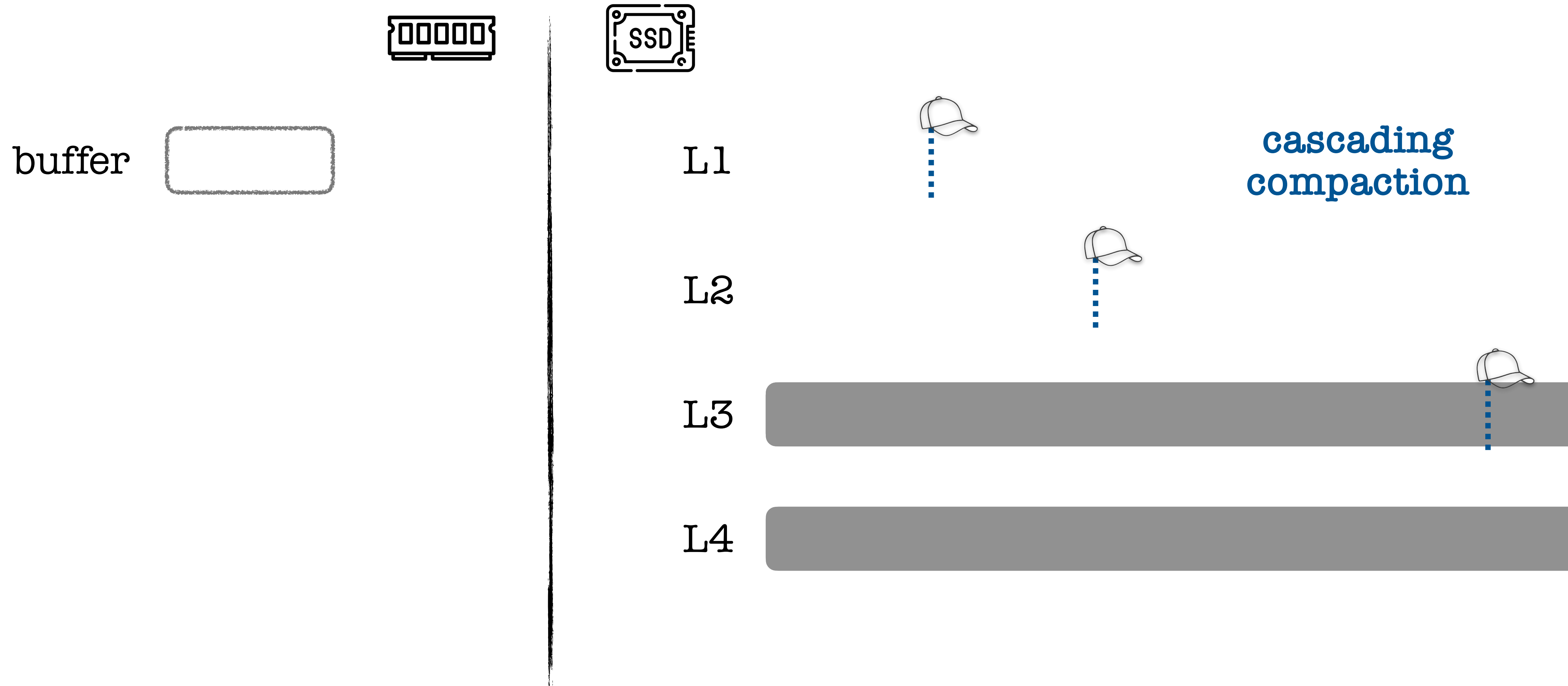
LSM Compaction



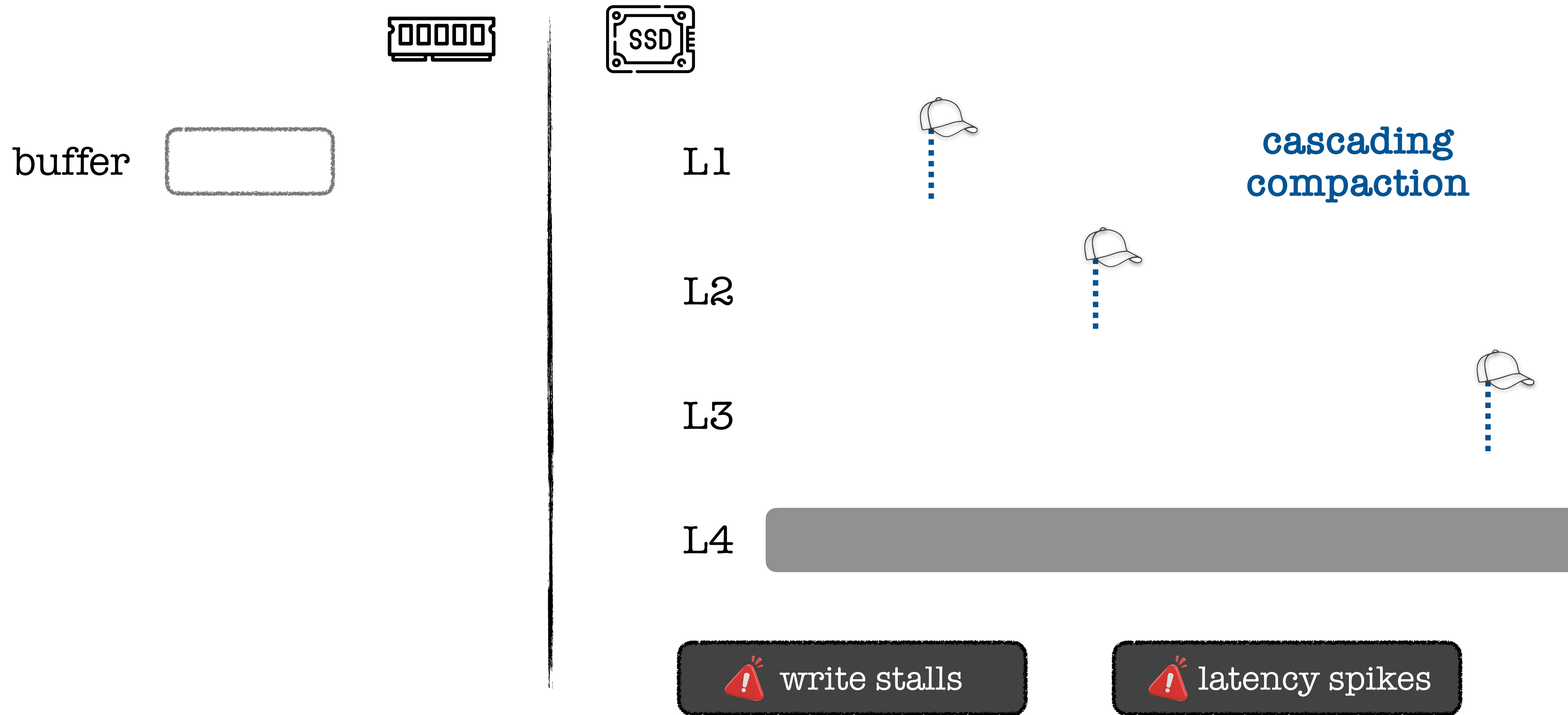
LSM Compaction



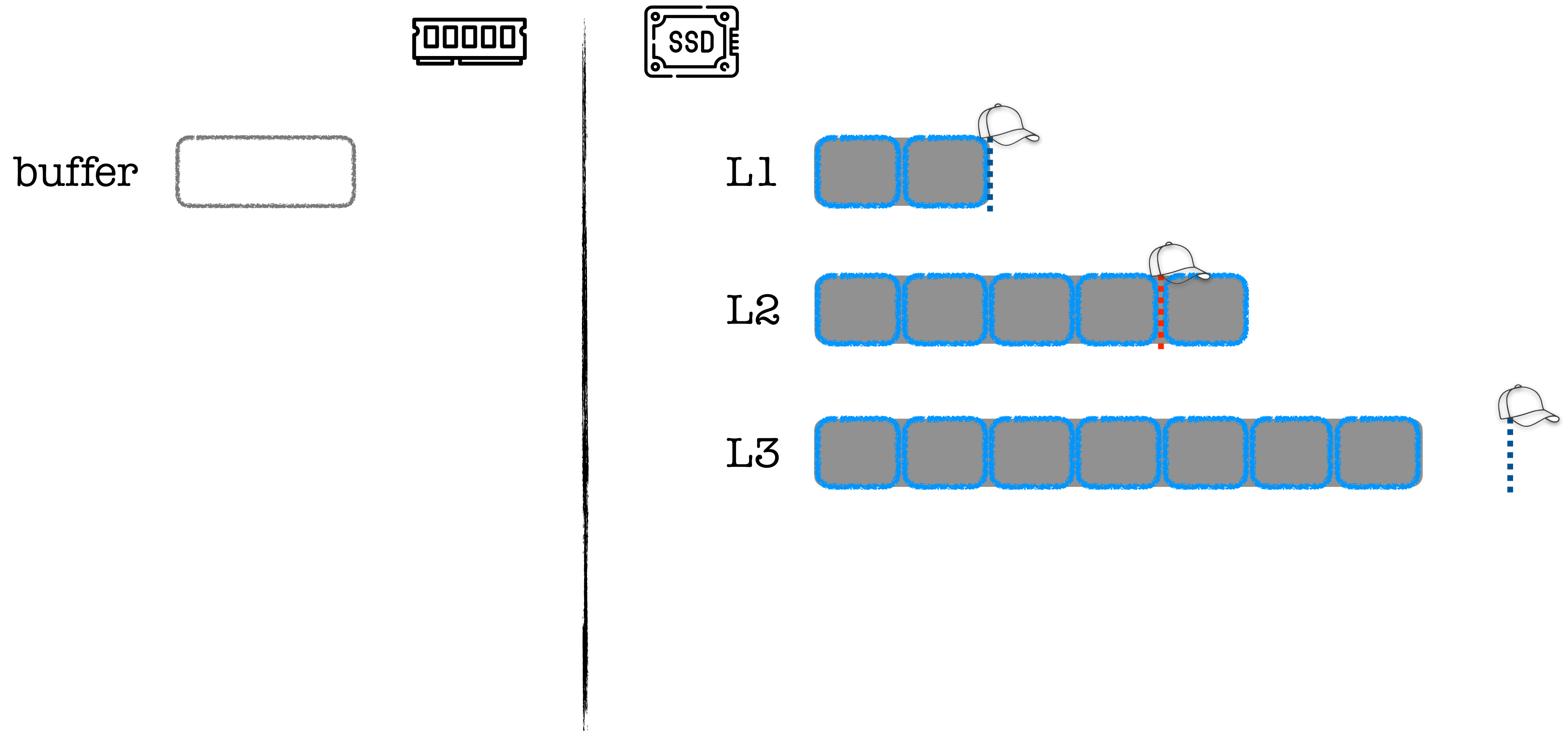
LSM Compaction



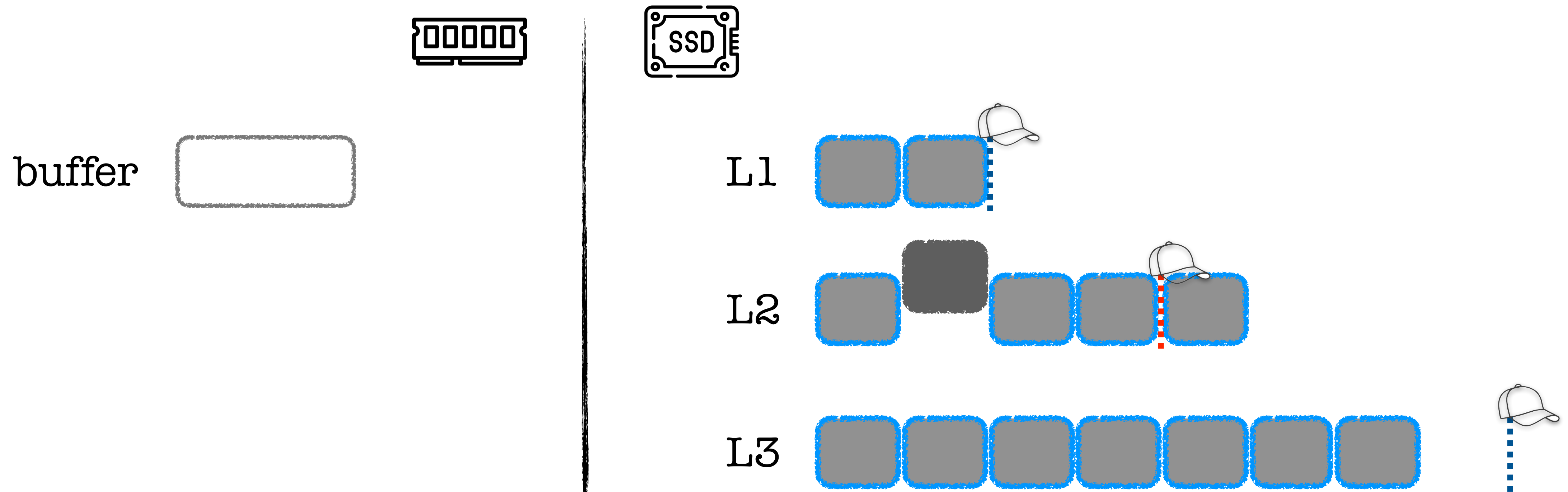
LSM Compaction



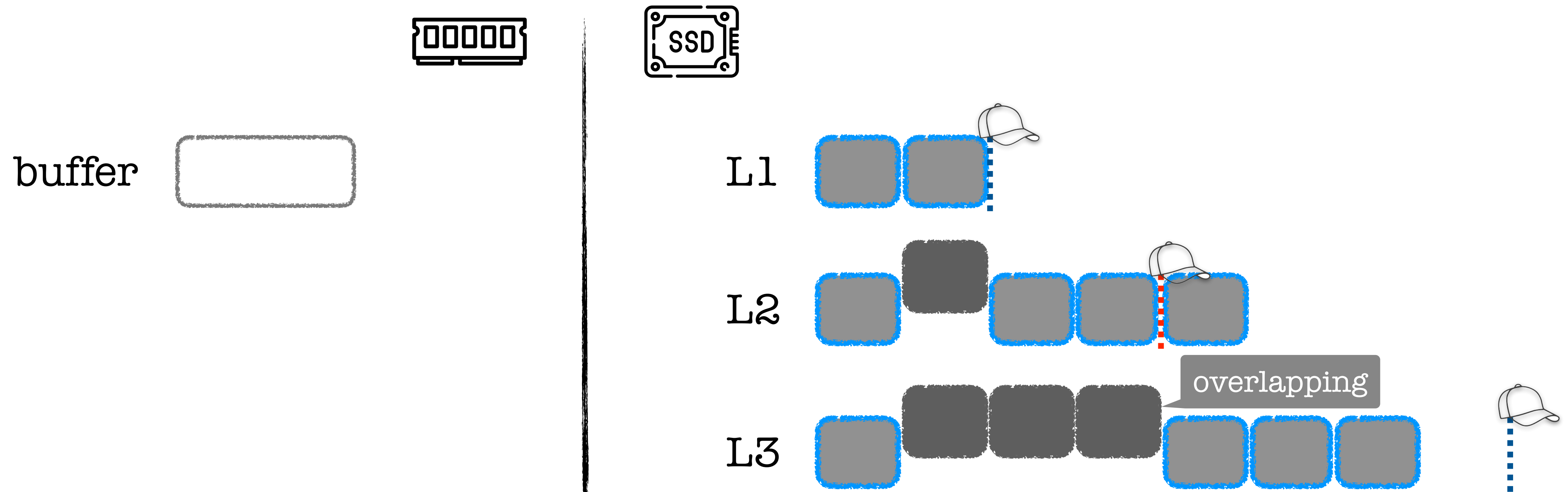
Partial Compaction in LSM



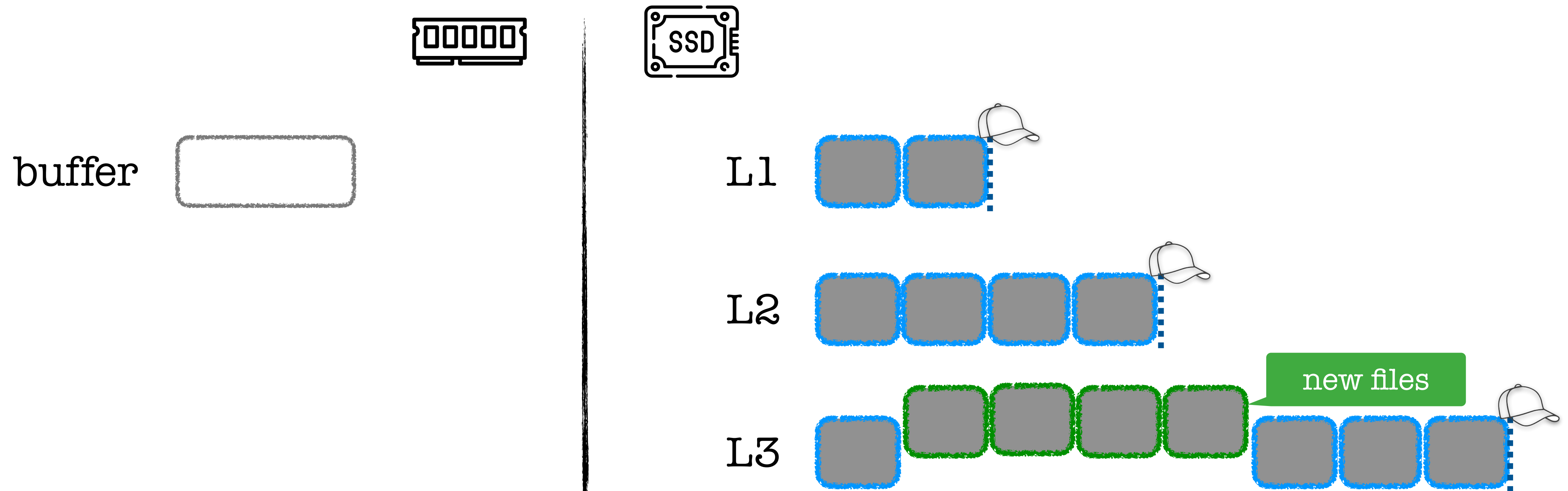
Partial Compaction in LSM



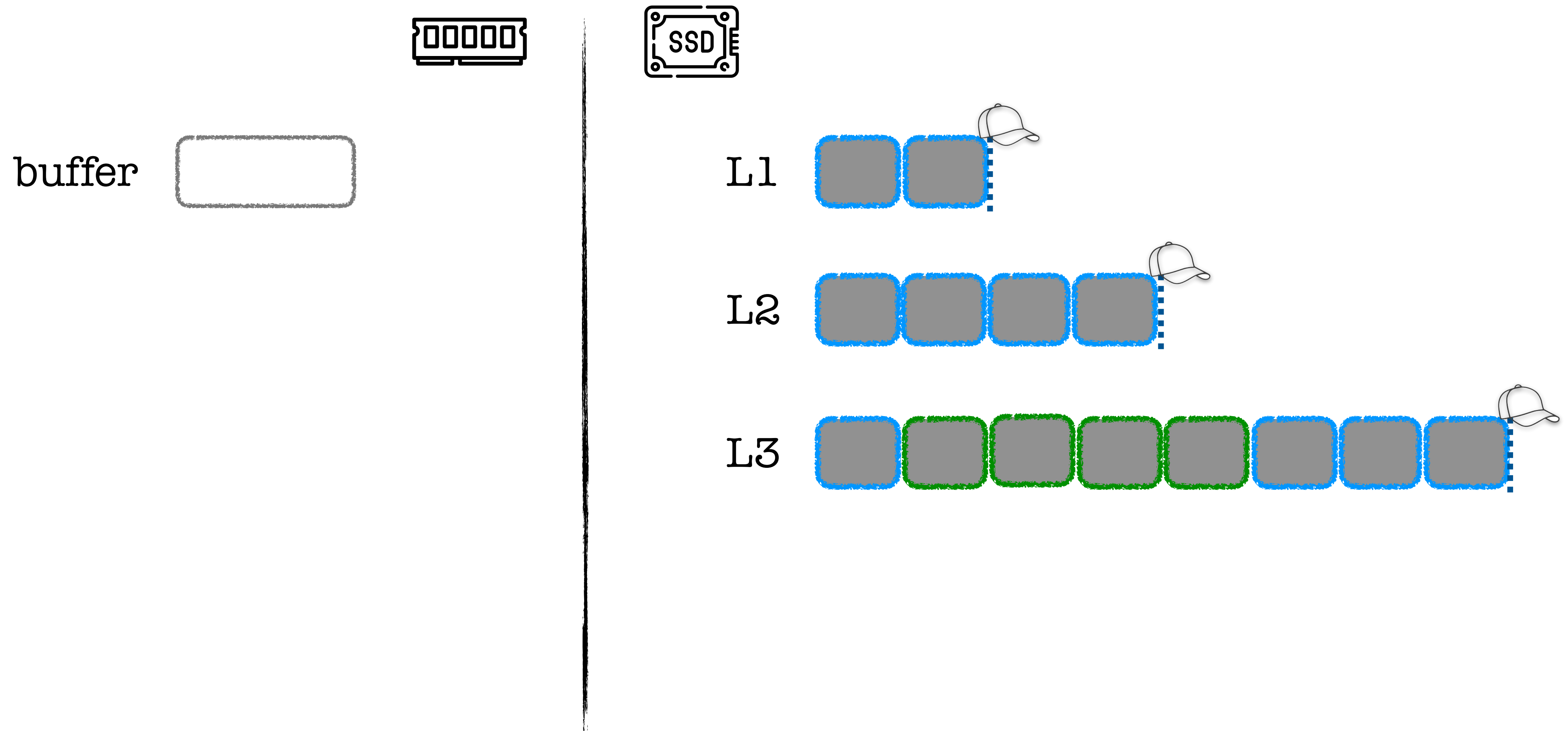
Partial Compaction in LSM



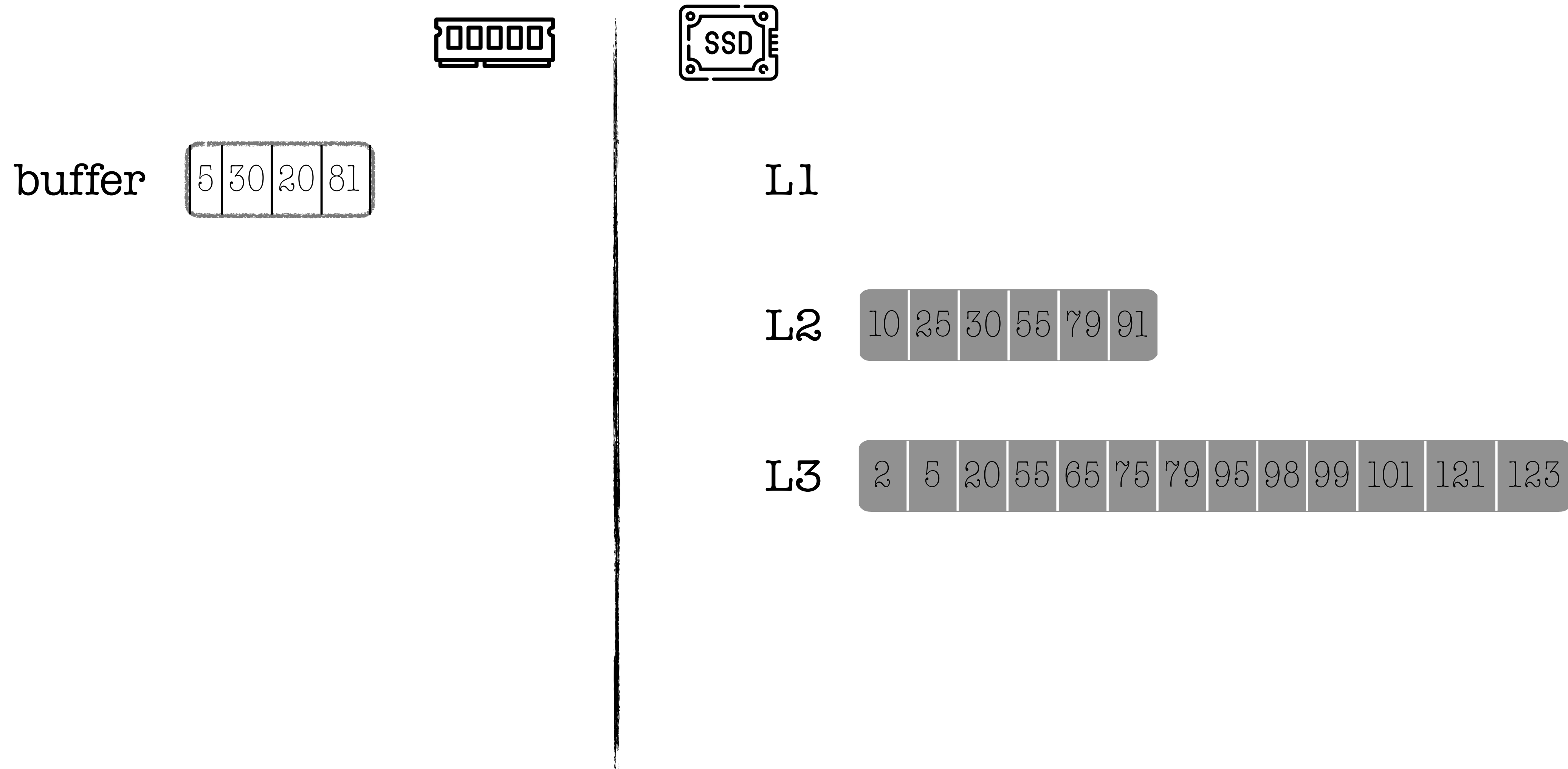
Partial Compaction in LSM



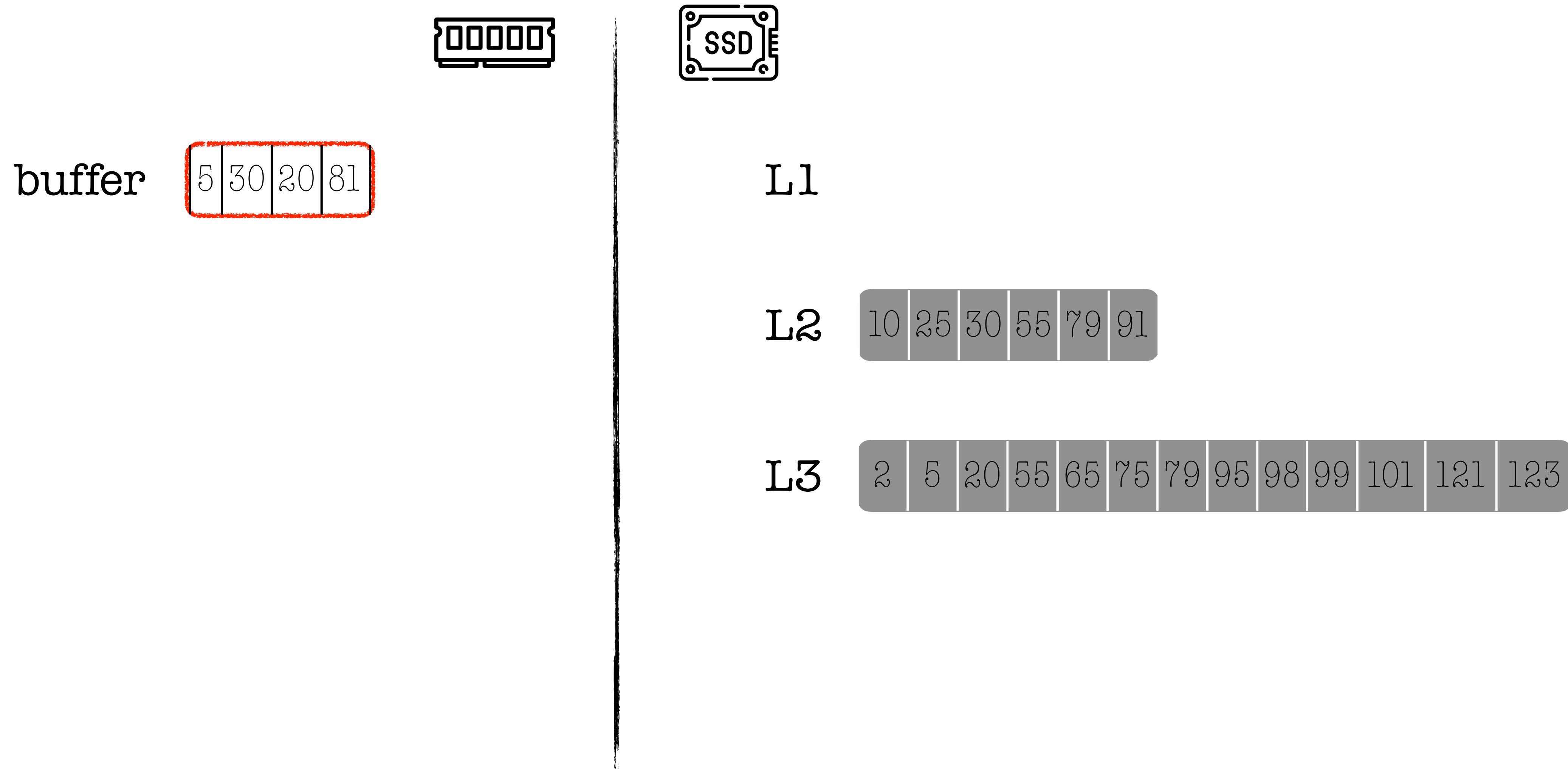
Partial Compaction in LSM



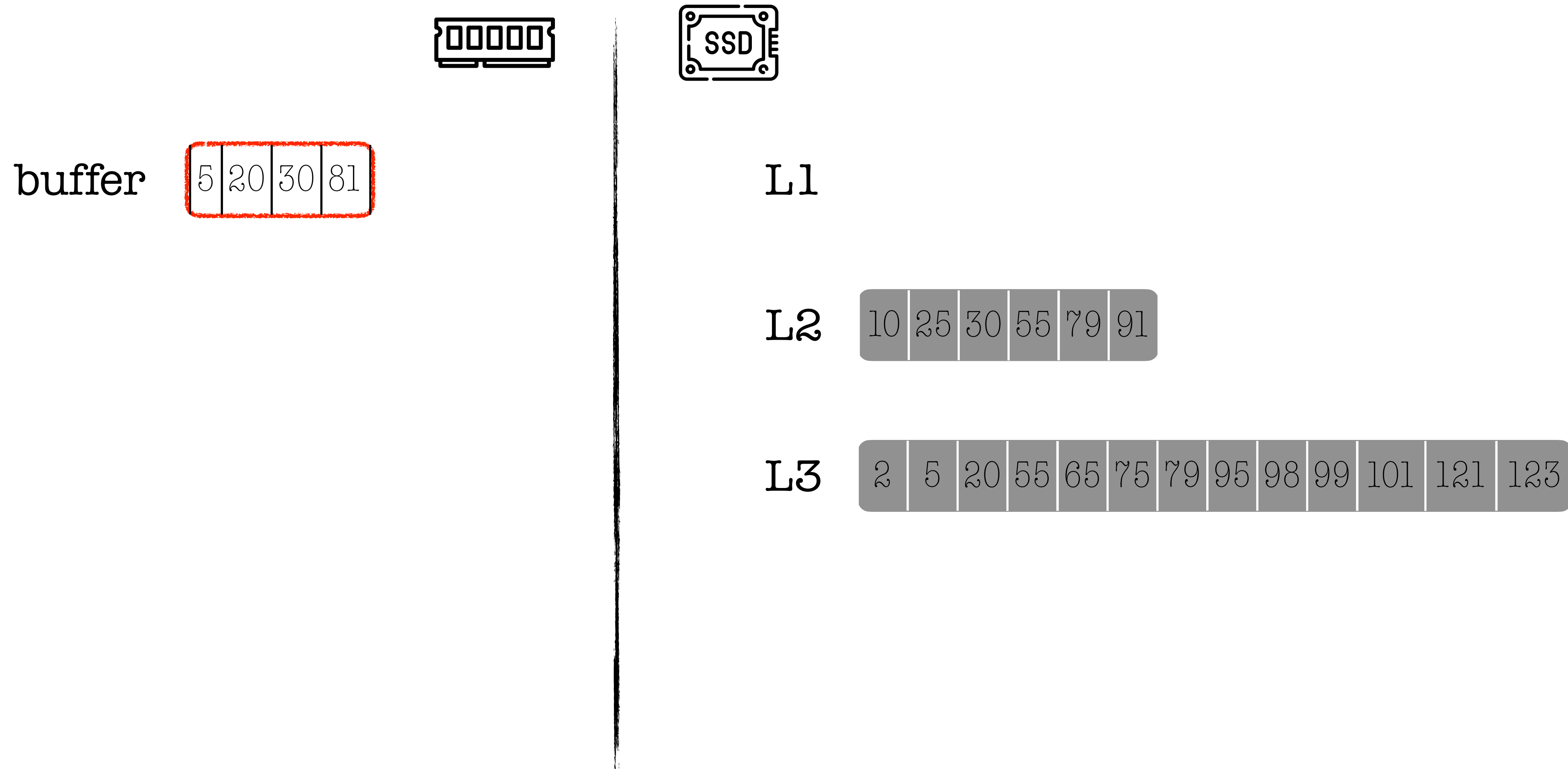
Updates & Deletes in LSM



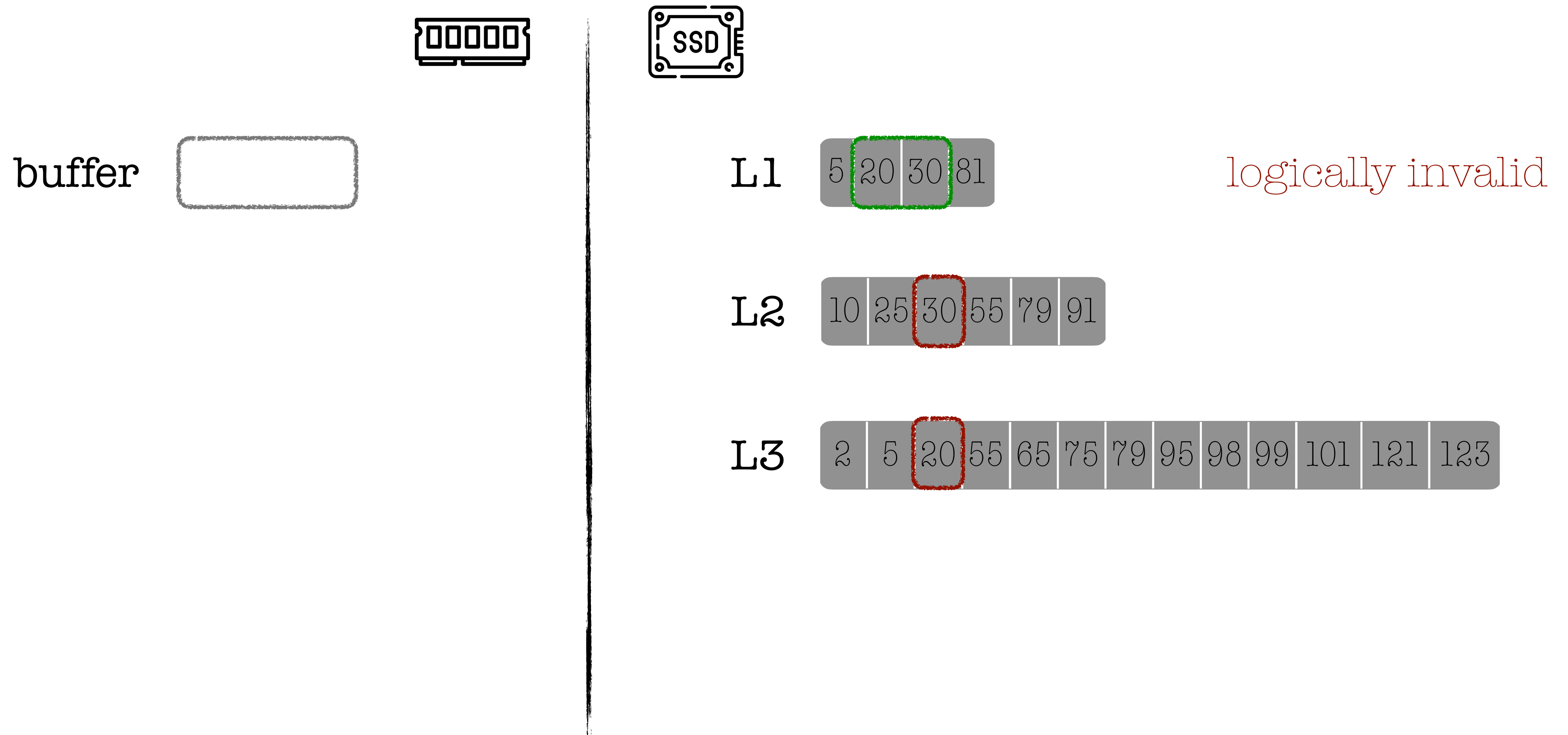
Updates & Deletes in LSM



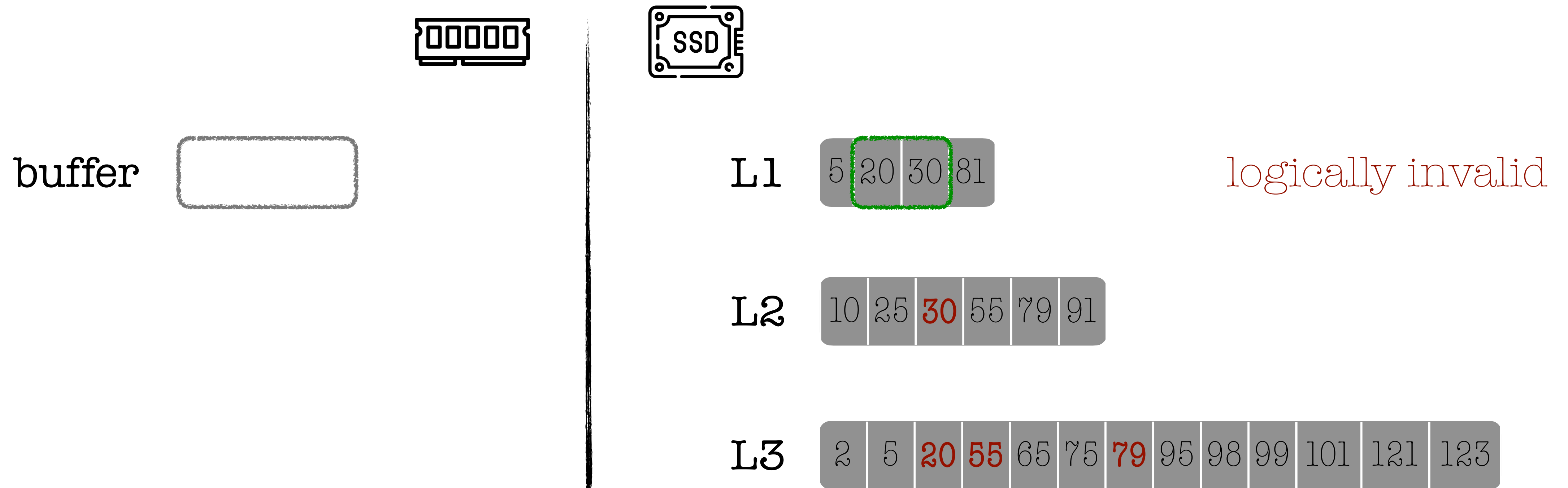
Updates & Deletes in LSM



Updates & Deletes in LSM



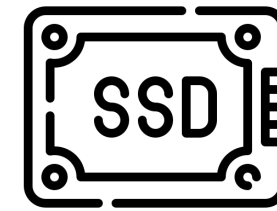
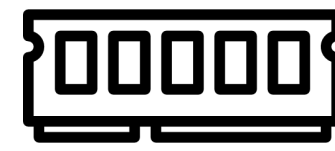
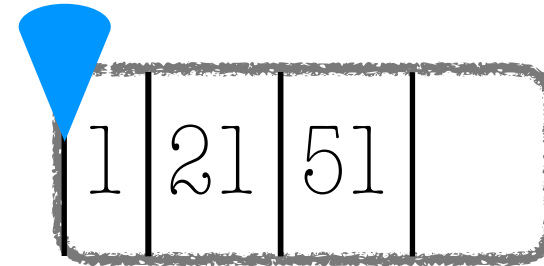
Updates & Deletes in LSM



Range Queries in LSM

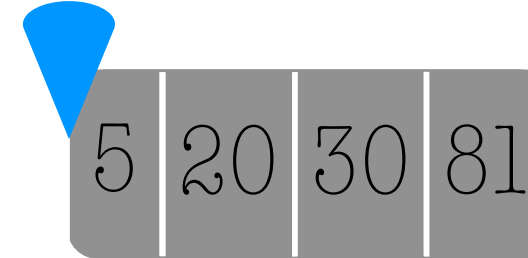
get(20, 80)

buffer

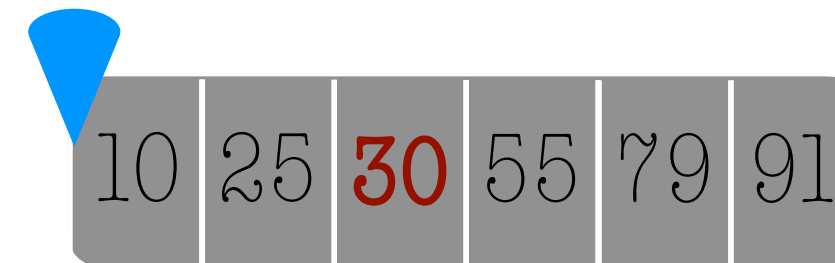


 iterator

L1



L2



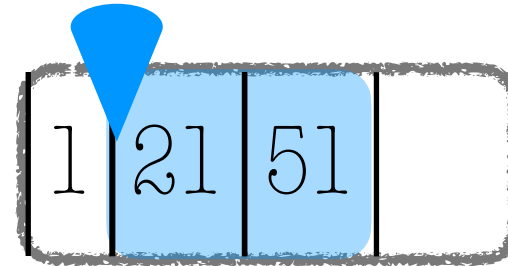
L3



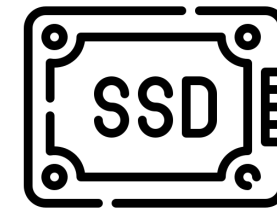
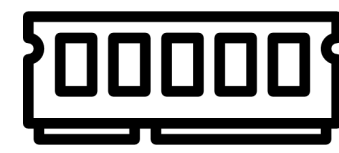
Range Queries in LSM

get(20, 80)

buffer

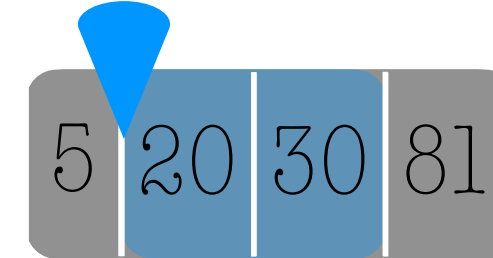


response

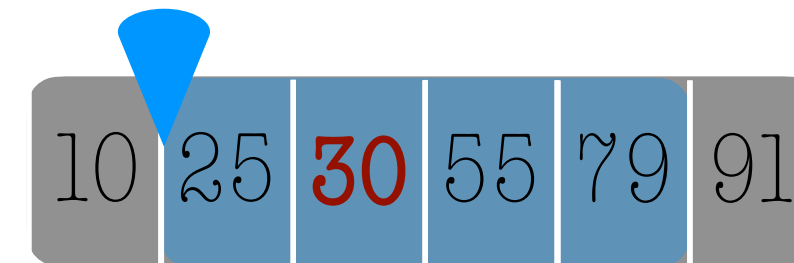


 iterator

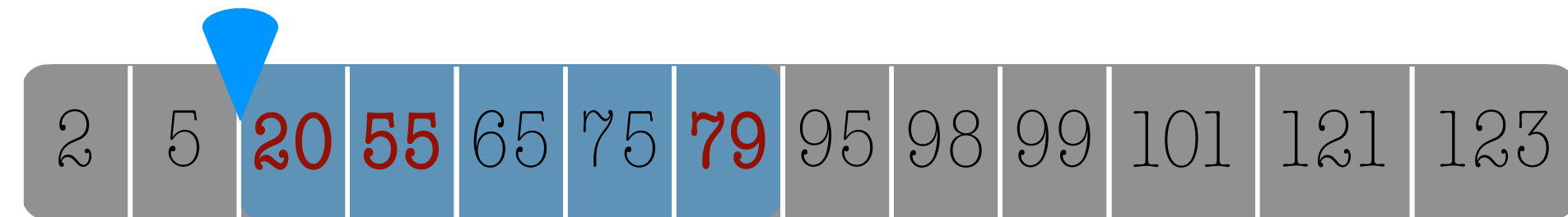
L1



L2



L3

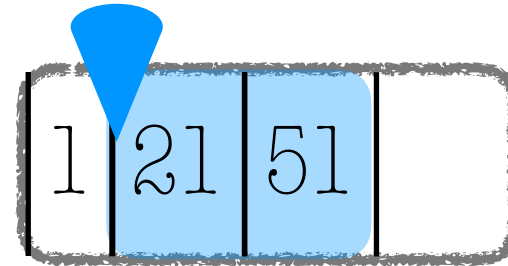


K-way merge

Range Queries in LSM

get(20, 80)

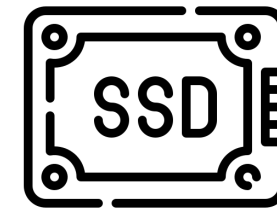
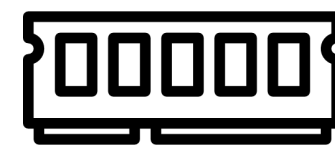
buffer



response

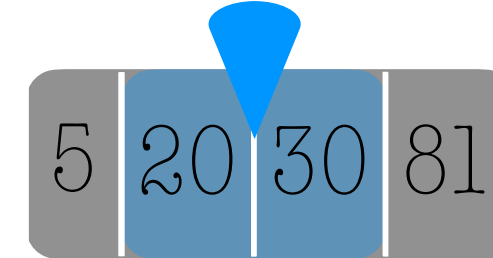
20

~~20~~

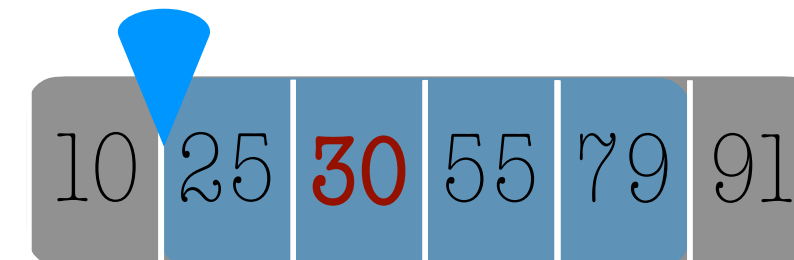


 iterator

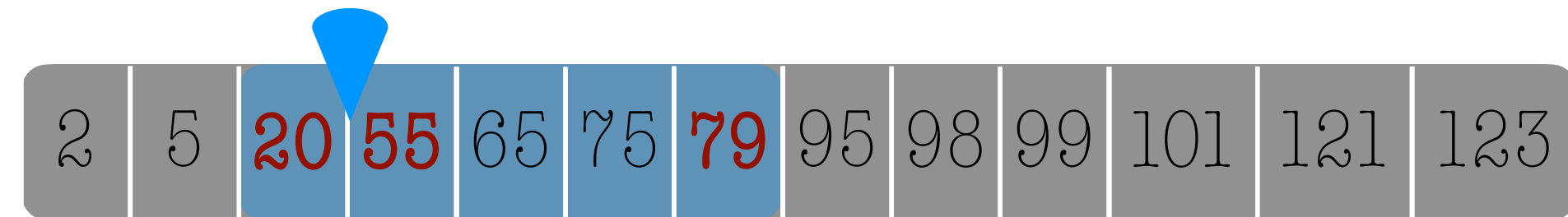
L1



L2



L3

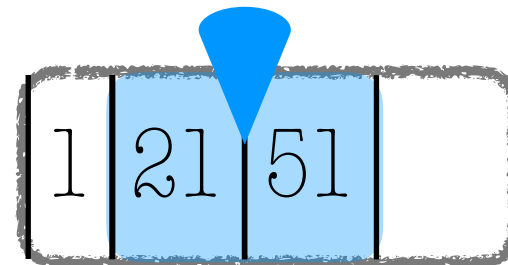


K-way merge

Range Queries in LSM

get(20, 80)

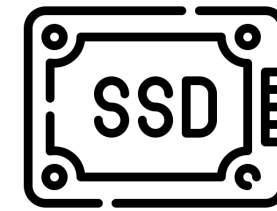
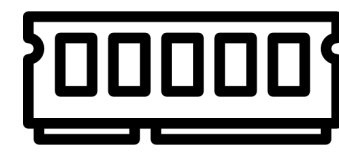
buffer



response

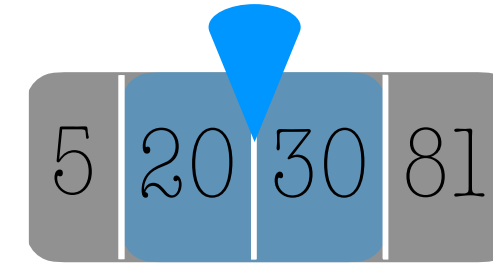
20 21

20

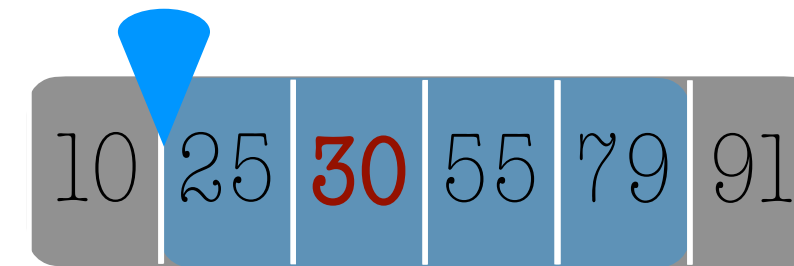


 iterator

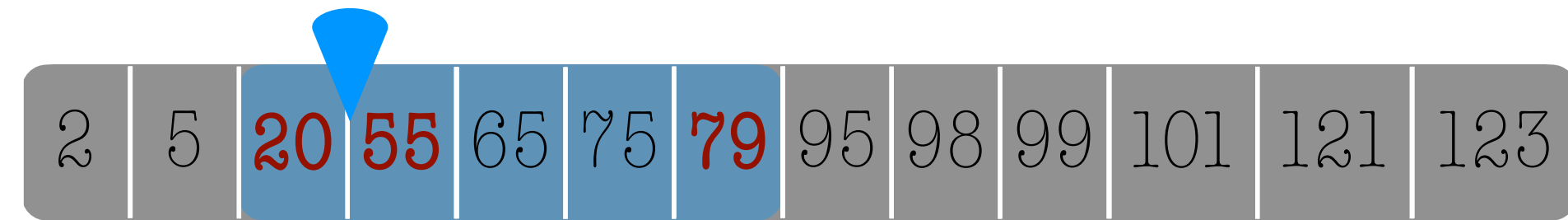
L1



L2



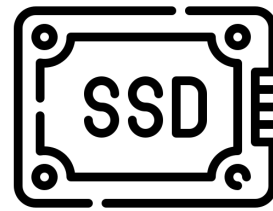
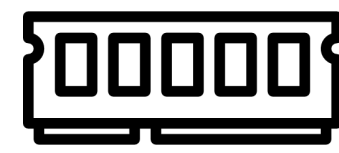
L3



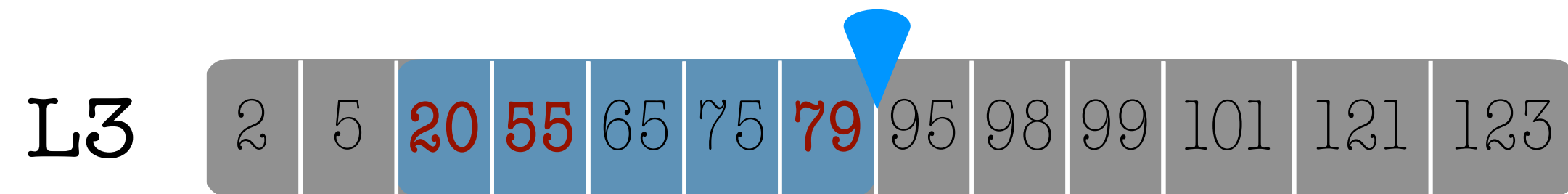
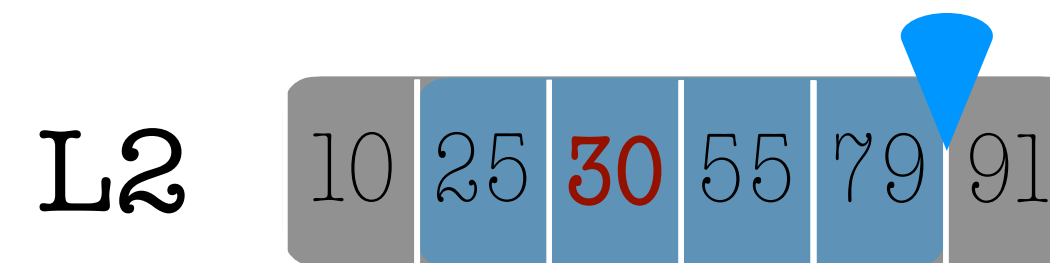
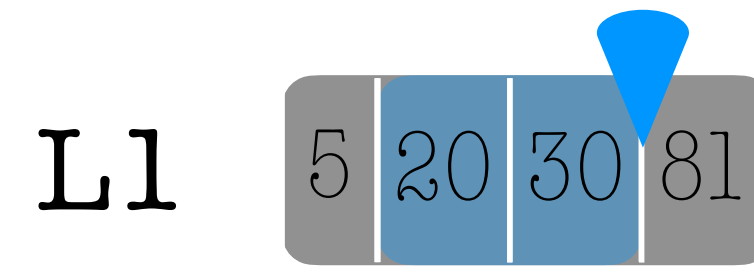
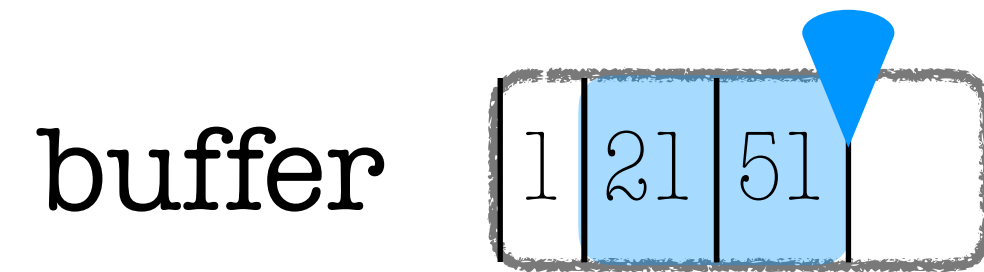
K-way merge

Range Queries in LSM

get(20, 80)



 iterator



response

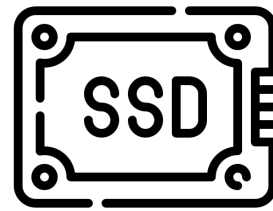
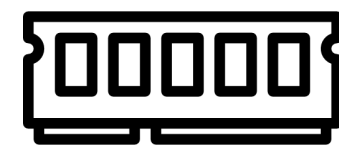
20 21 25 30 51 55 65 75 79

20 30 55 79

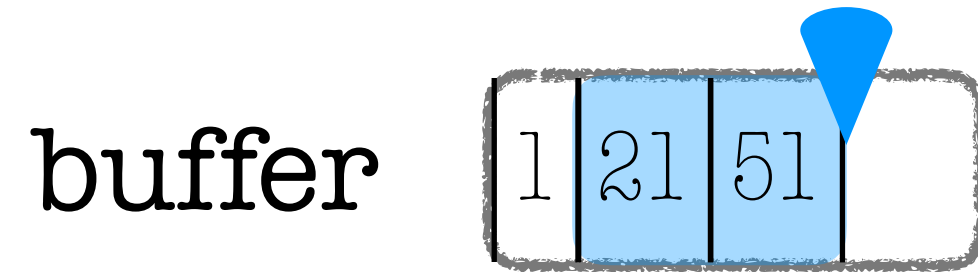
K-way merge

Range Queries in LSM

get(20, 80)



 iterator

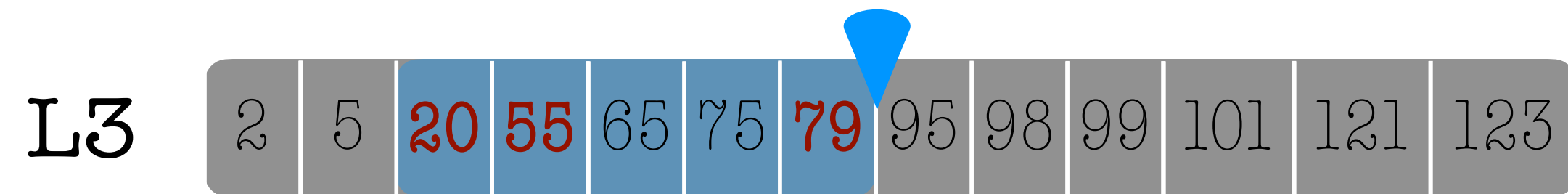
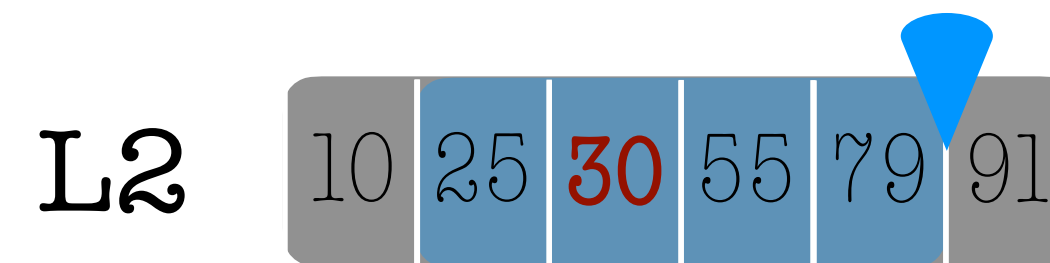
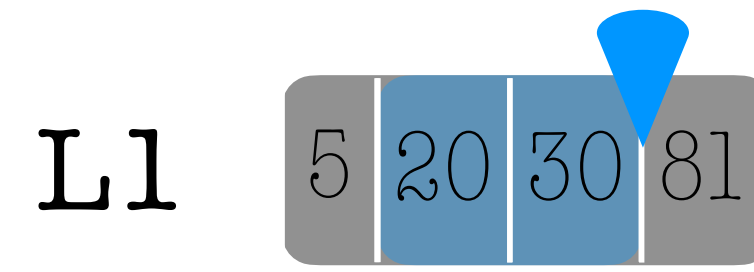


response

20 21 25 30 51 55 65 75 79



~~20~~ ~~30~~ ~~55~~ ~~79~~



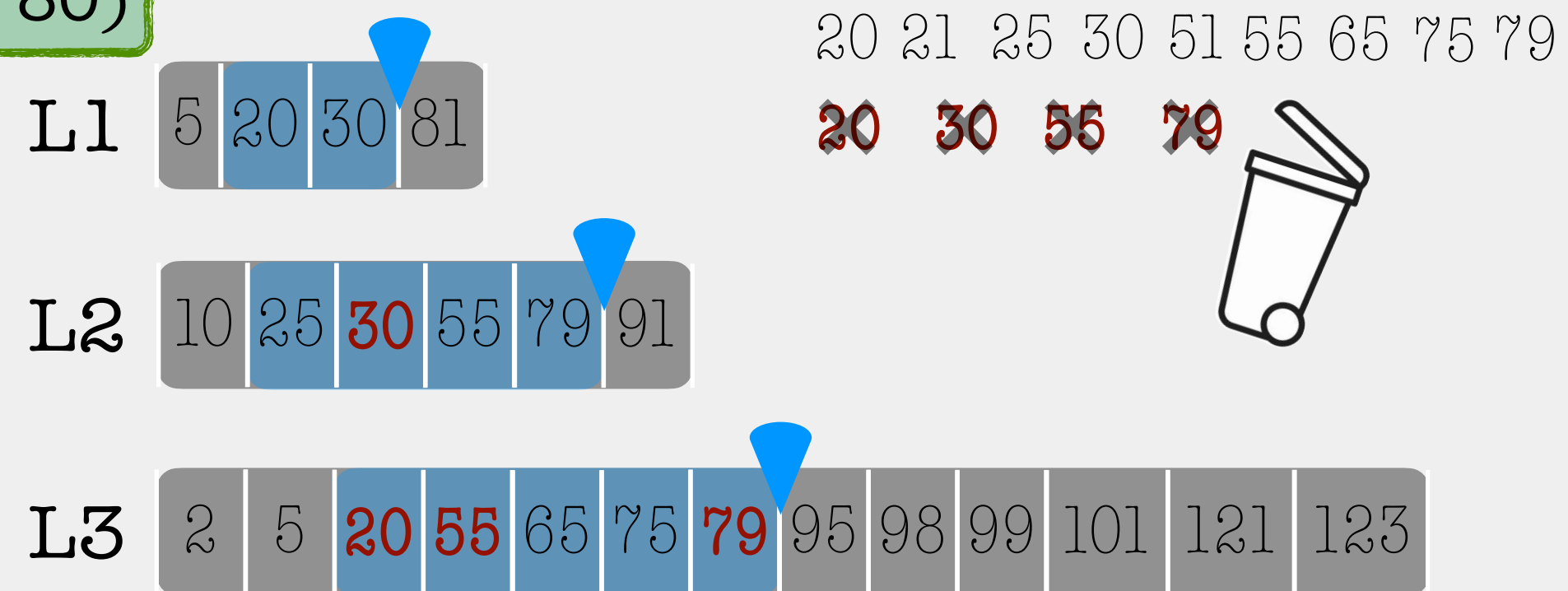
 repetitive merging

 wasted work

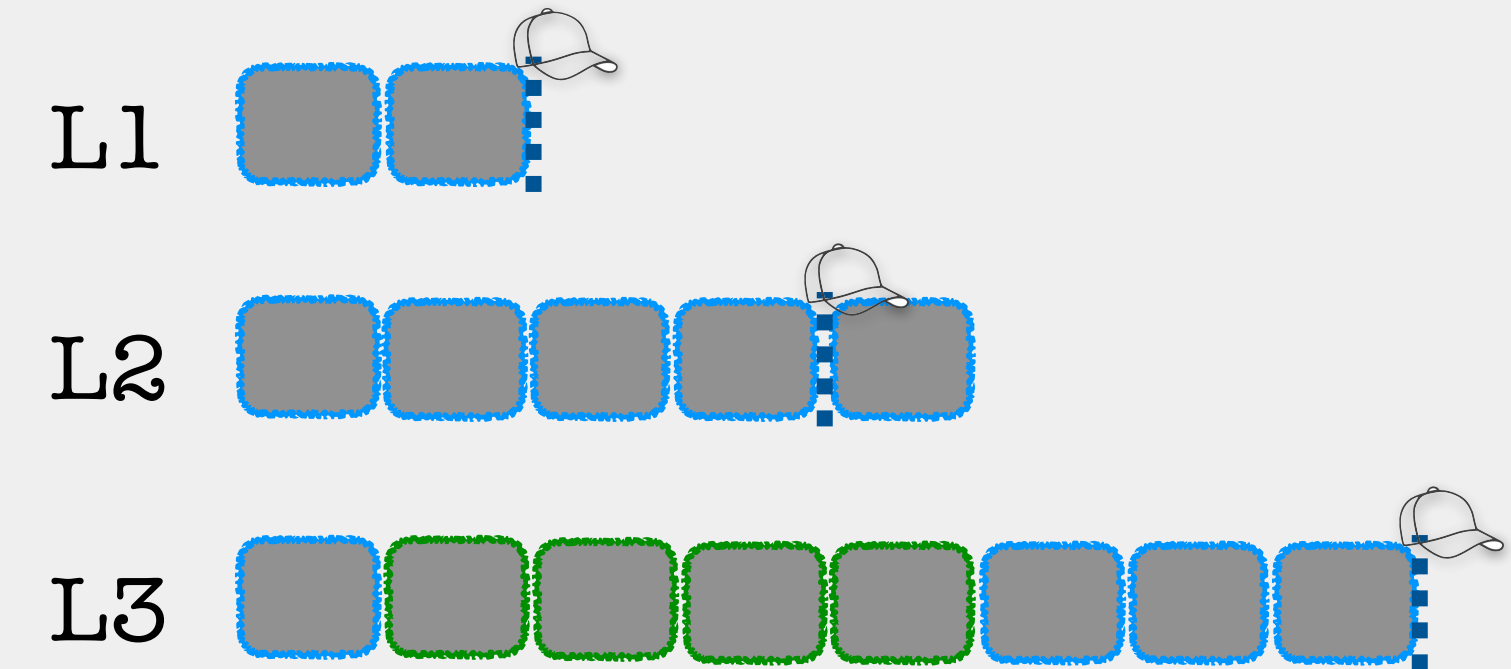
Limitations of the State-of-the-Art

Problem 1

get(20, 80)



Problem 2



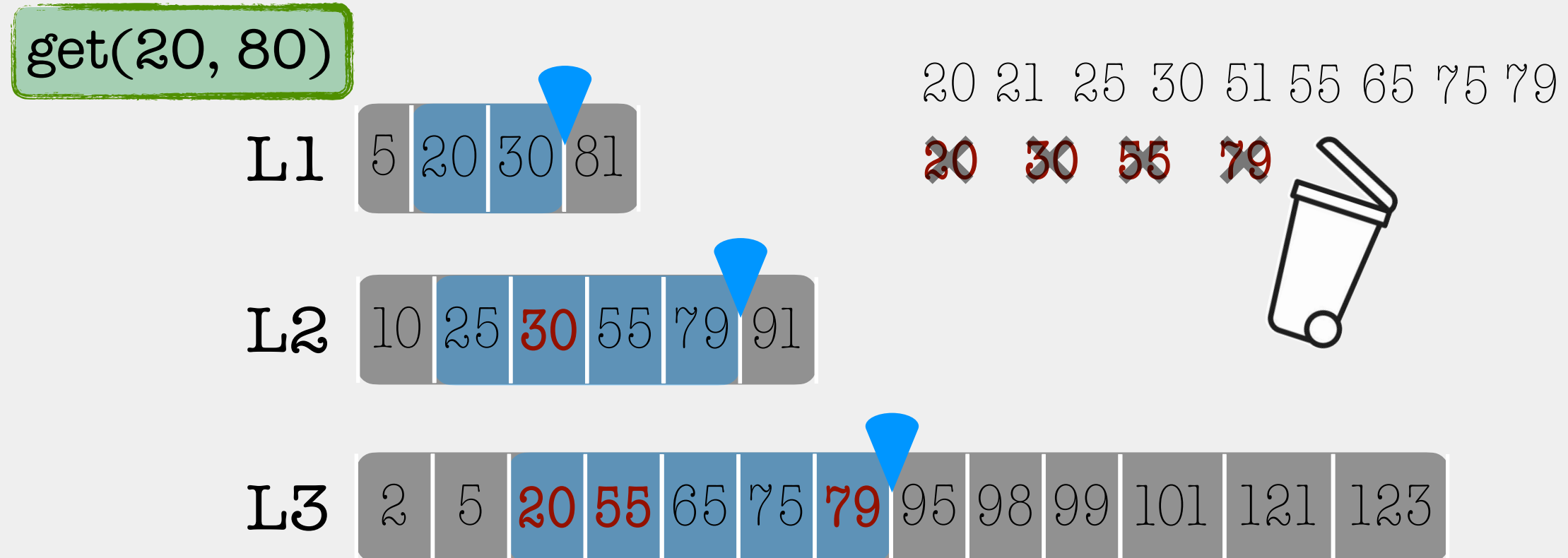
repetitive merging



wasted work

Limitations of the State-of-the-Art

Problem 1

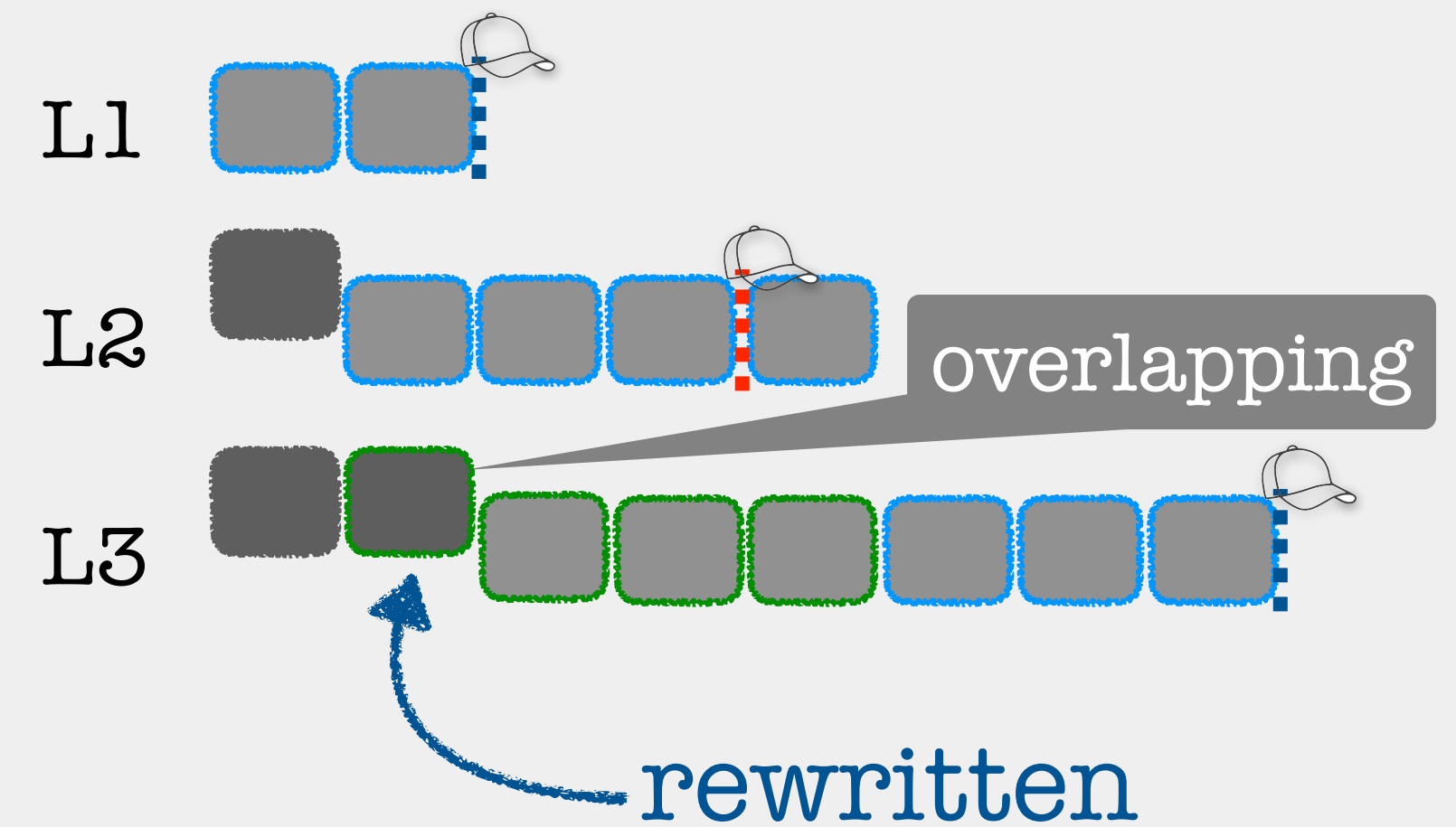


repetitive merging



wasted work

Problem 2



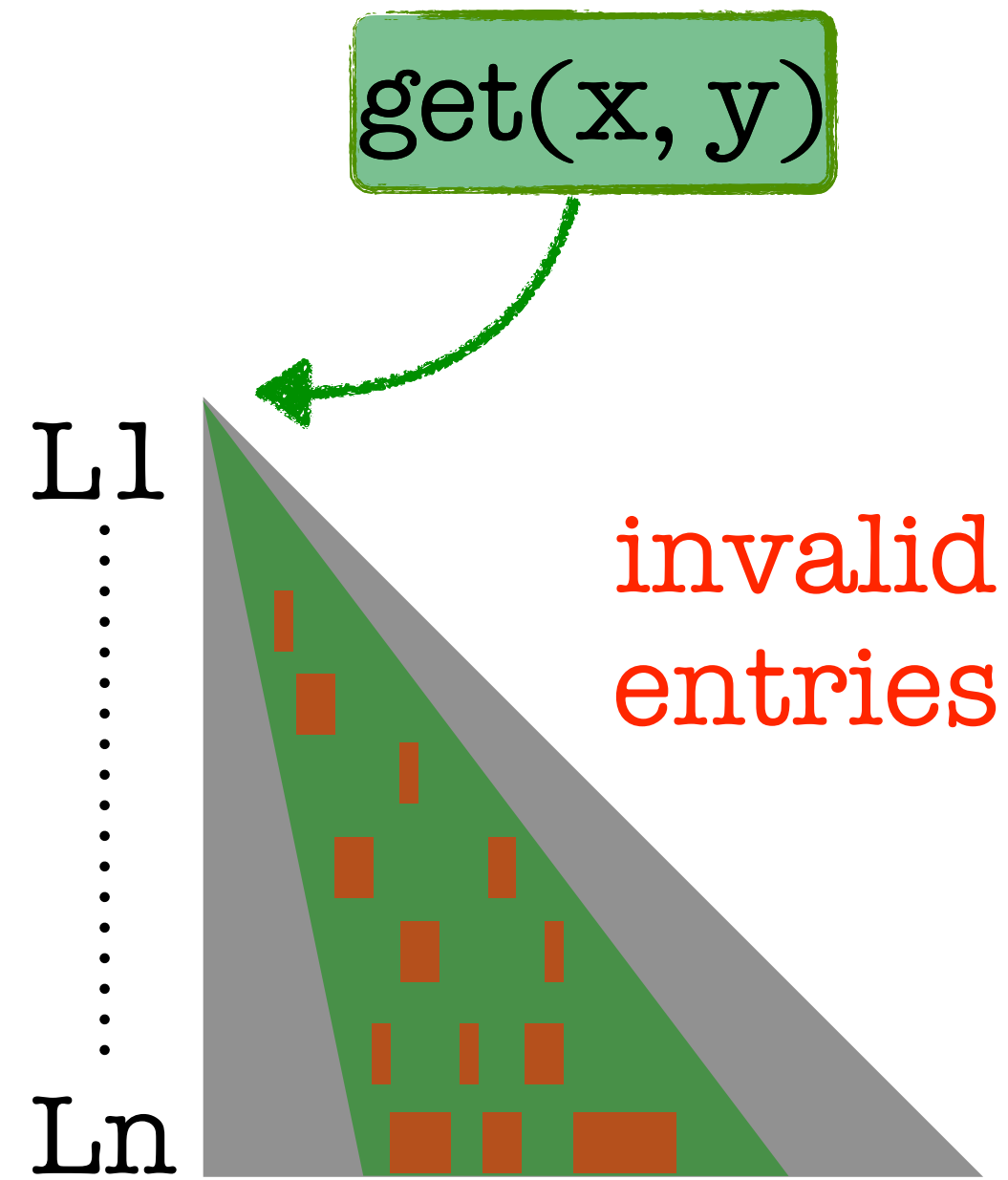
read amp.



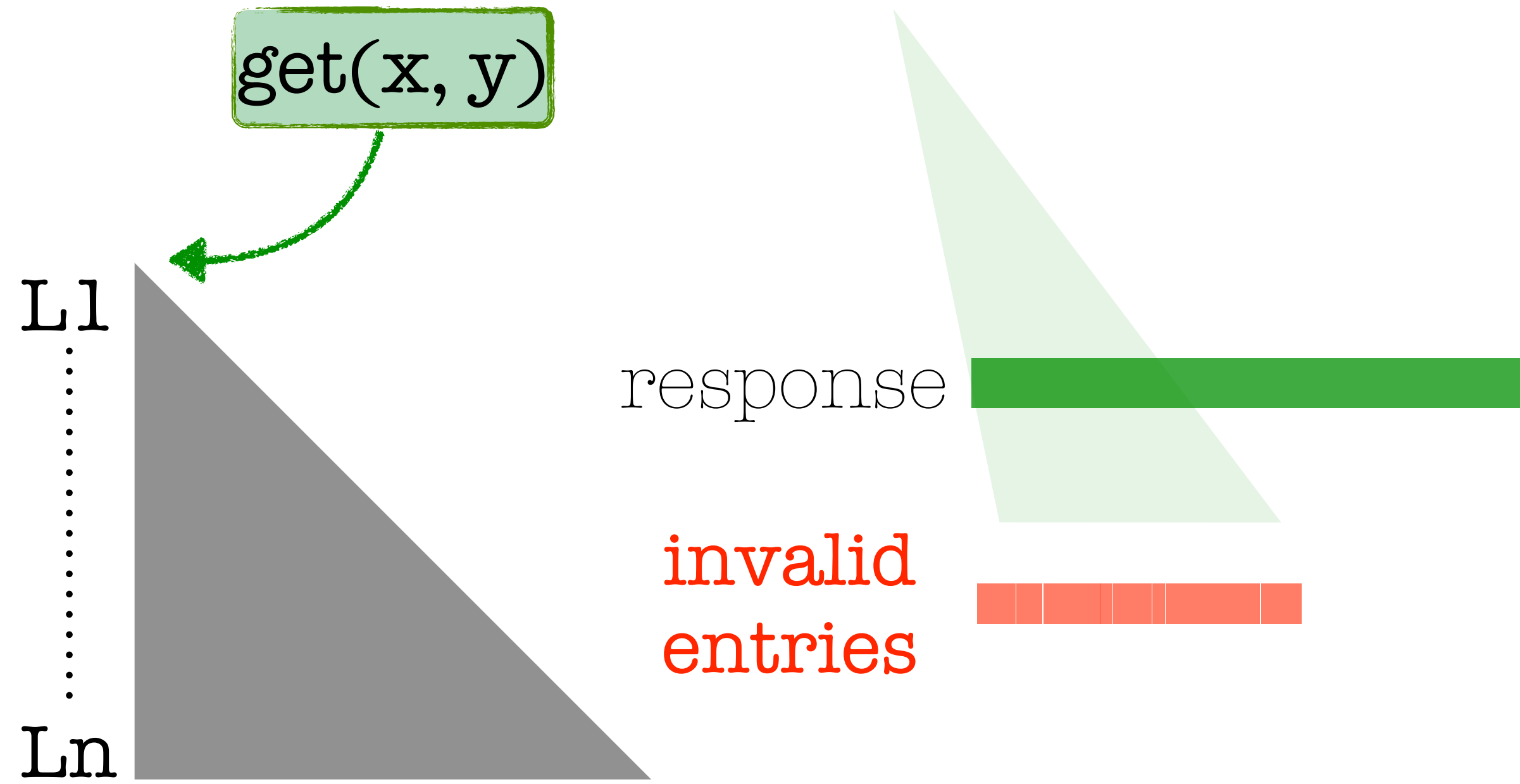
write amp.

Can we do something better?

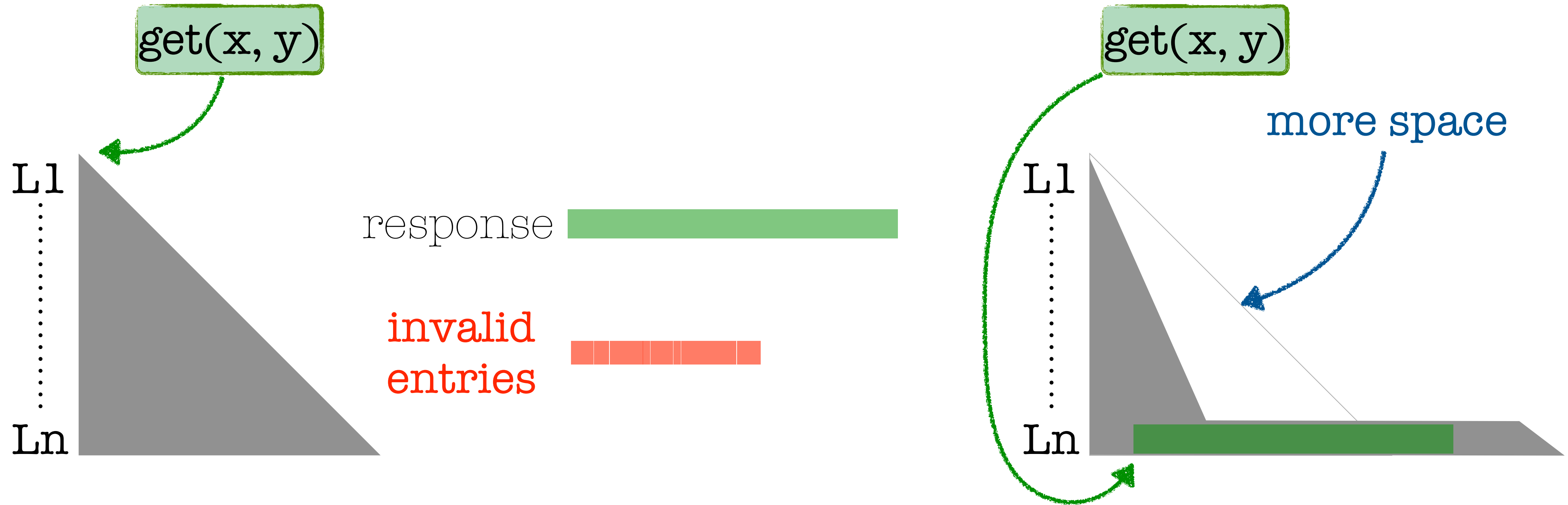
Naive Solution: Merge-on-Scan



Naive Solution: Merge-on-Scan



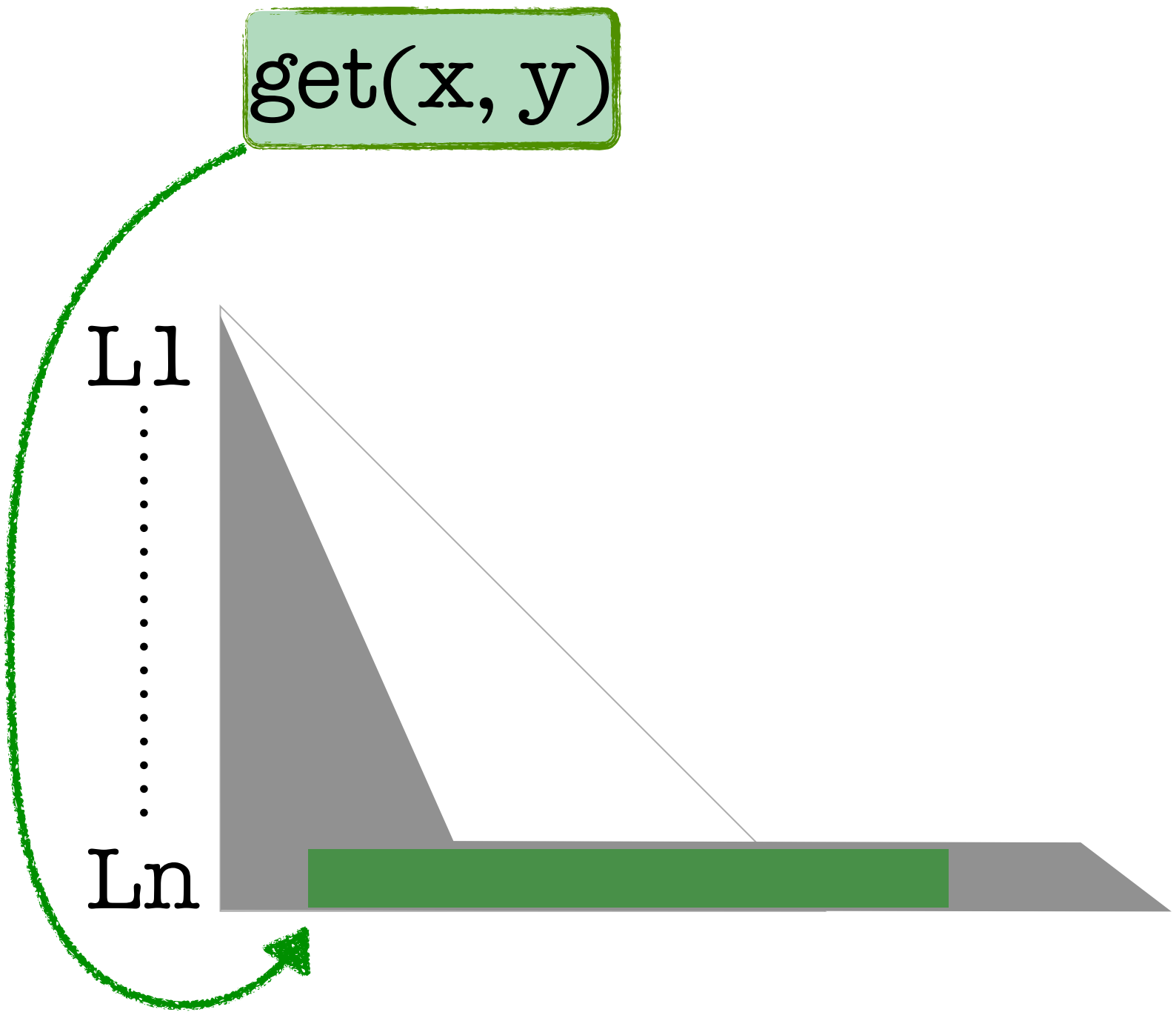
Naive Solution: Merge-on-Scan



💡 less read & write amp.

💡 no repetitive merging

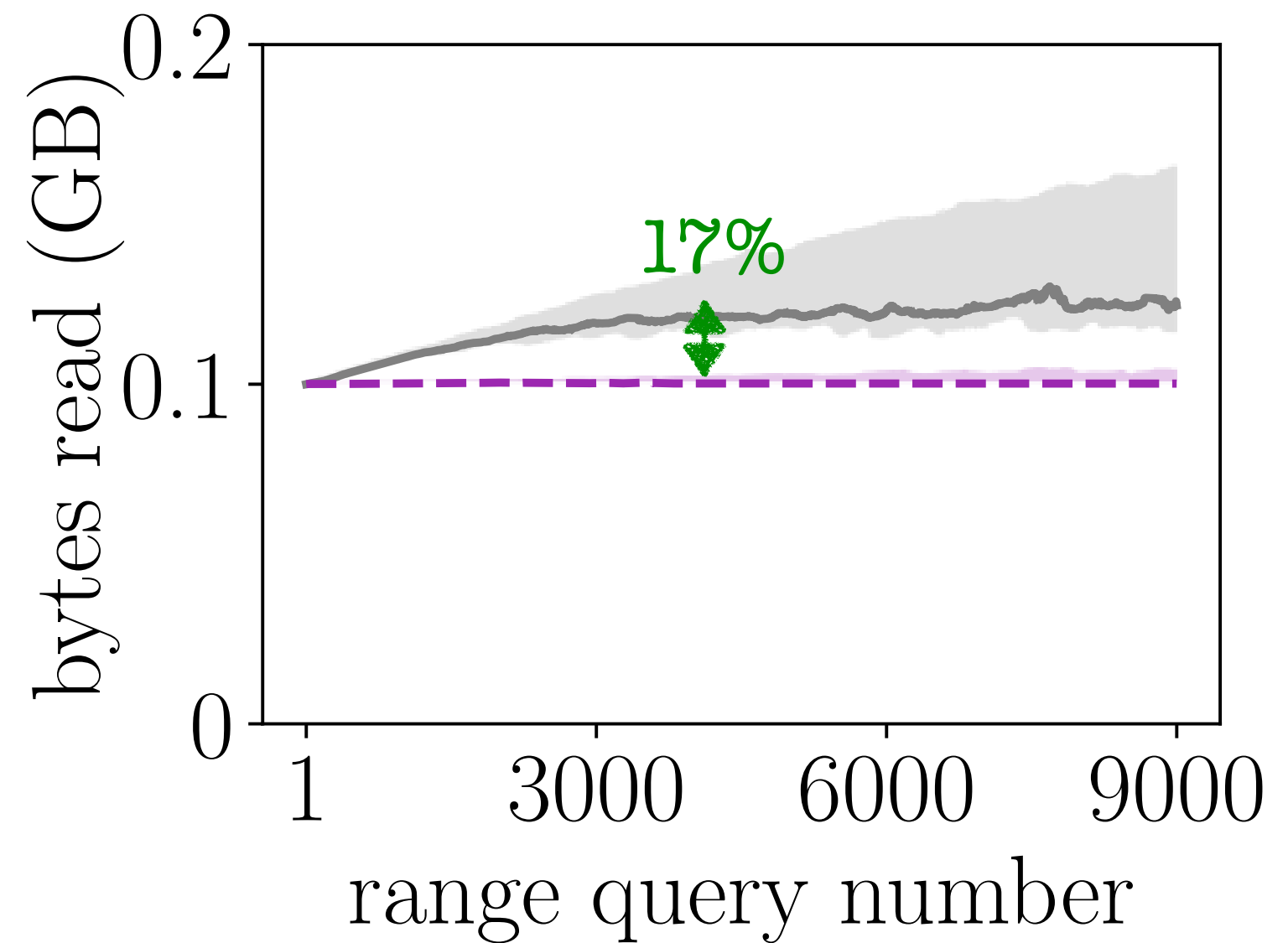
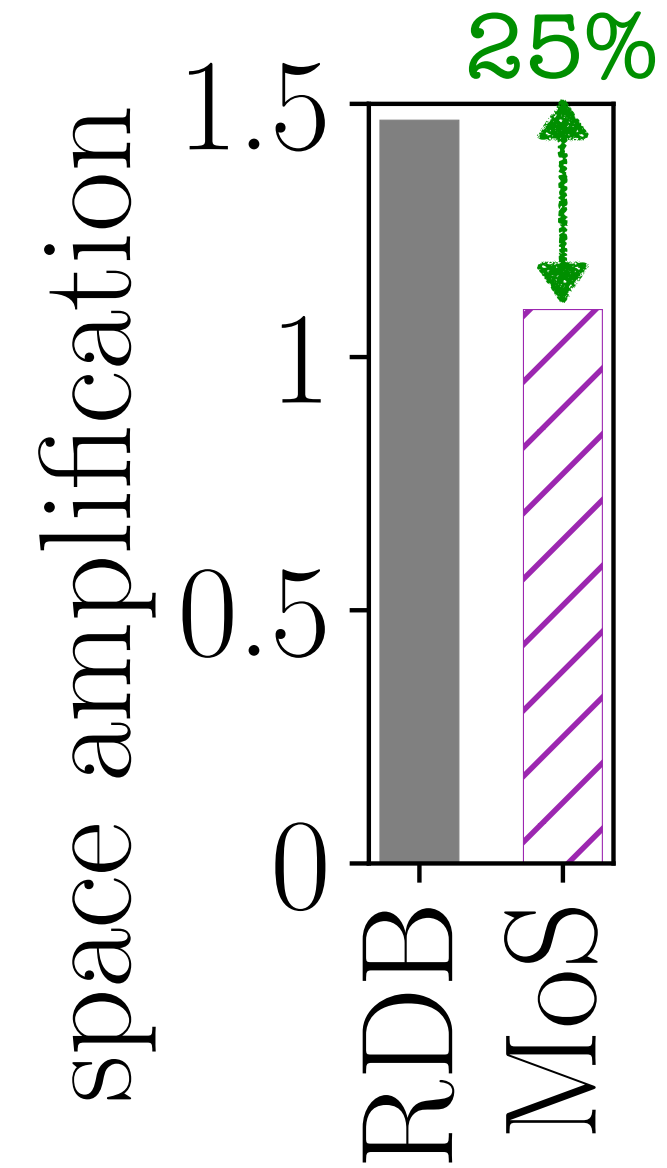
Merge-on-Scan



💡 less read & write amp.

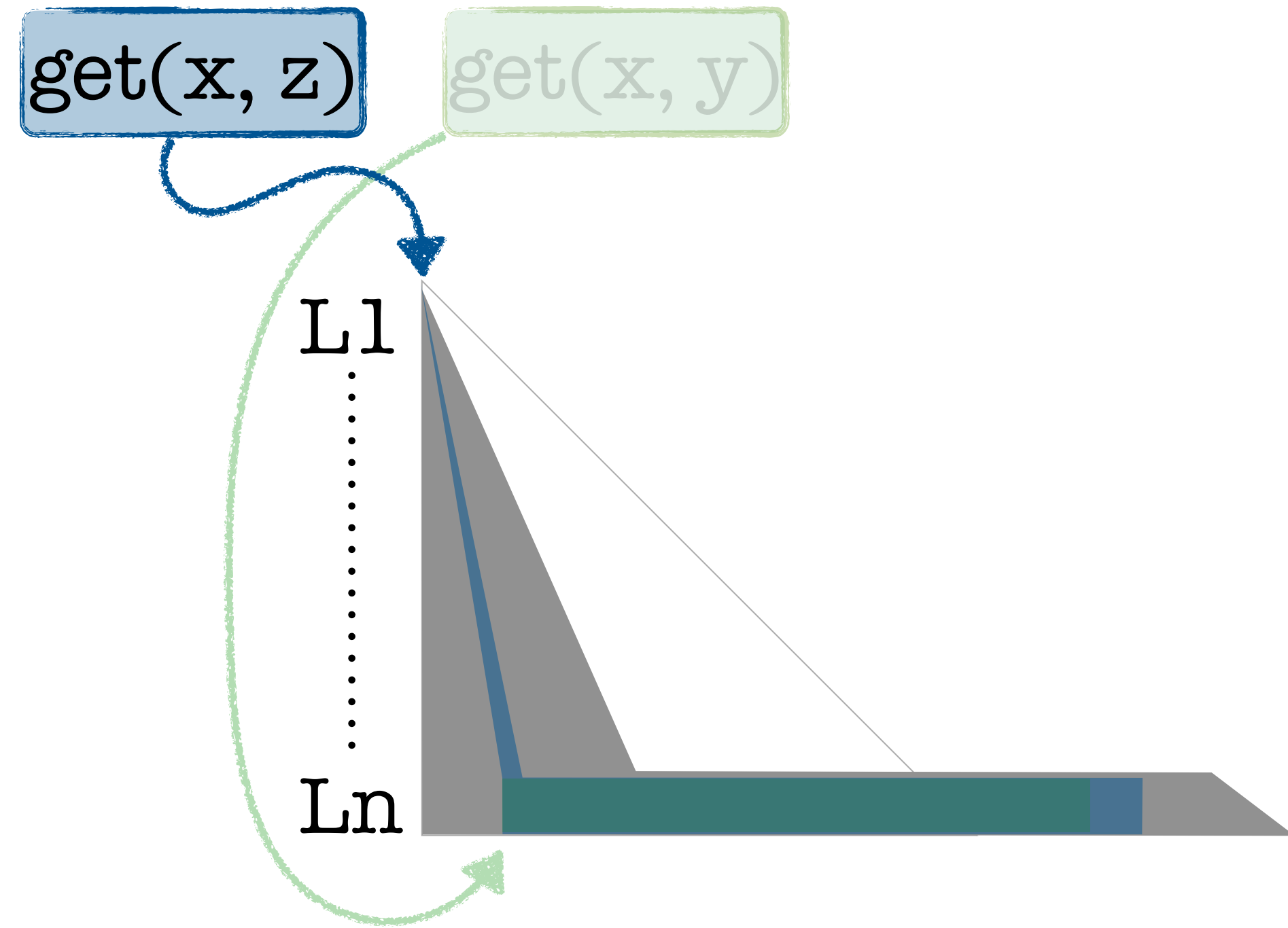
💡 no repetitive merging

— RocksDB (RDB) - - - Merge-on-Scan (MoS)
■ RocksDB ▨ Merge-on-Scan



MoS reduces space amplification by 25% and reads up to 17% less data during range queries

Merge-on-Scan



! slower RQ

! increases write amp.

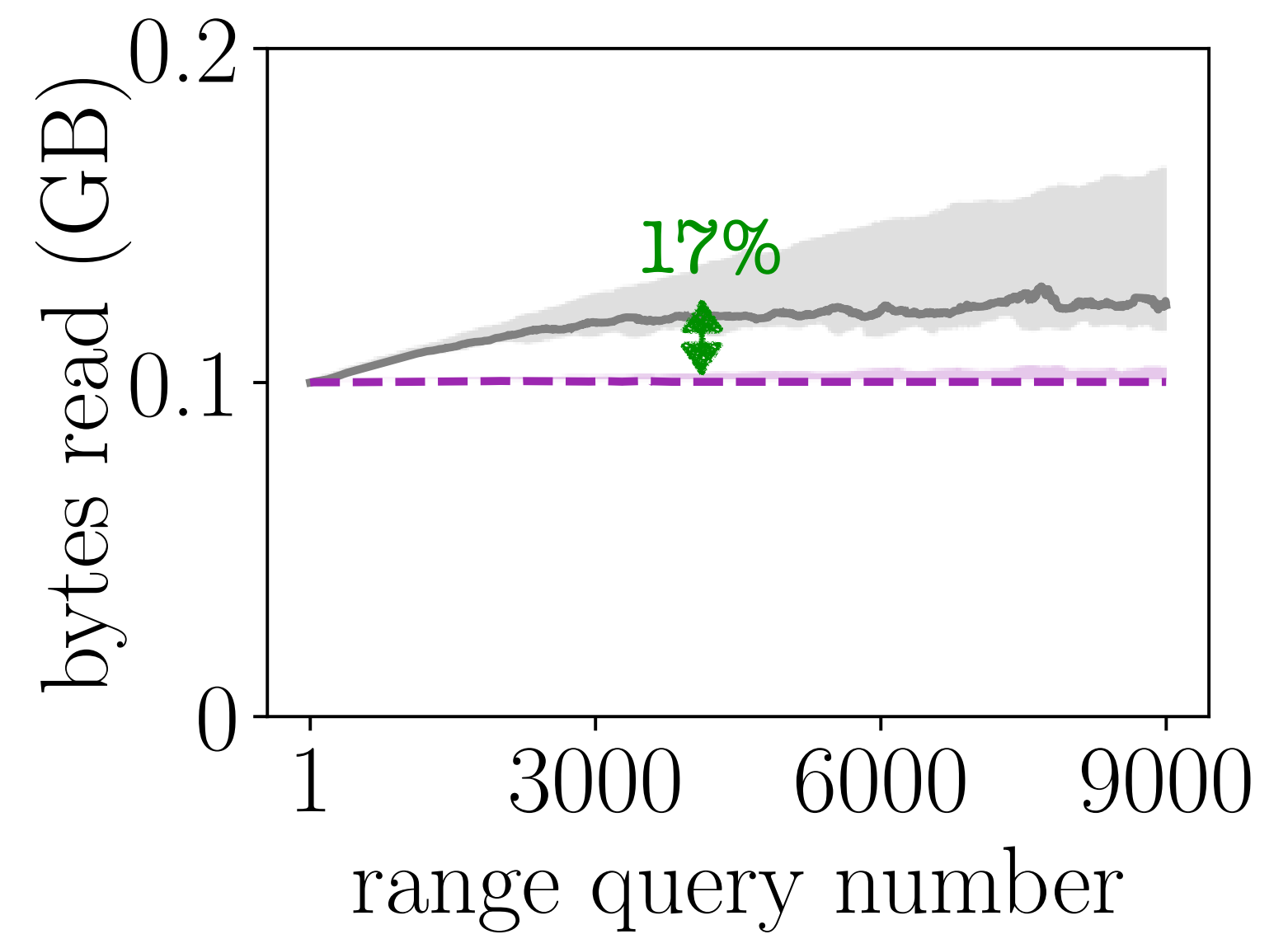
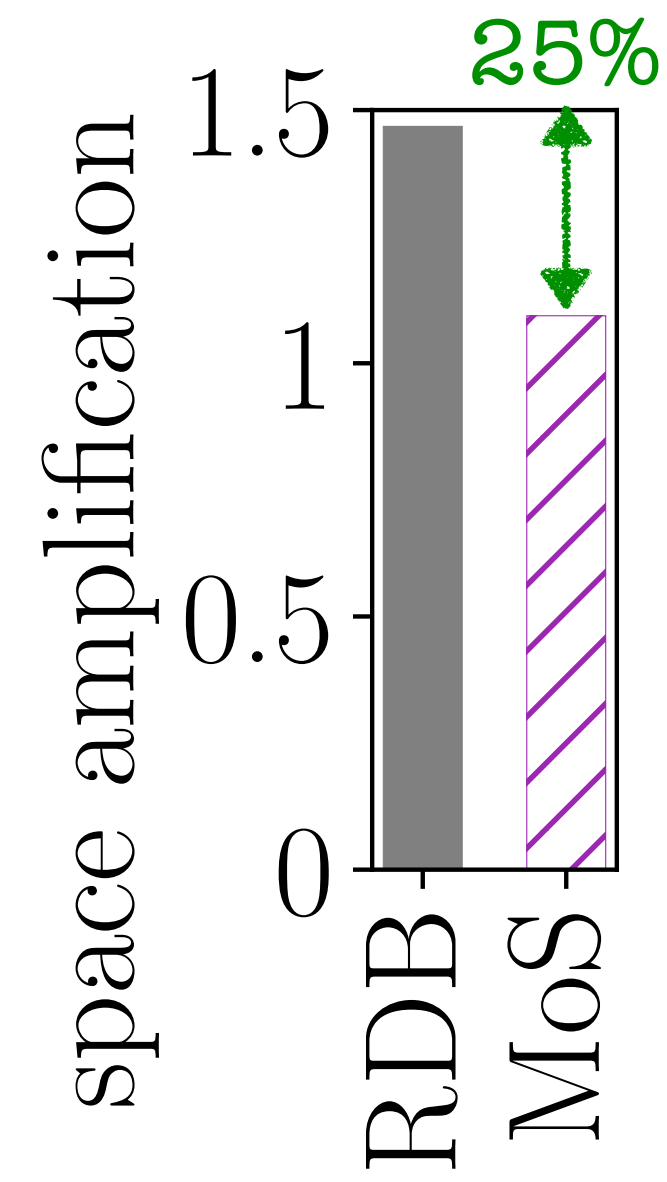
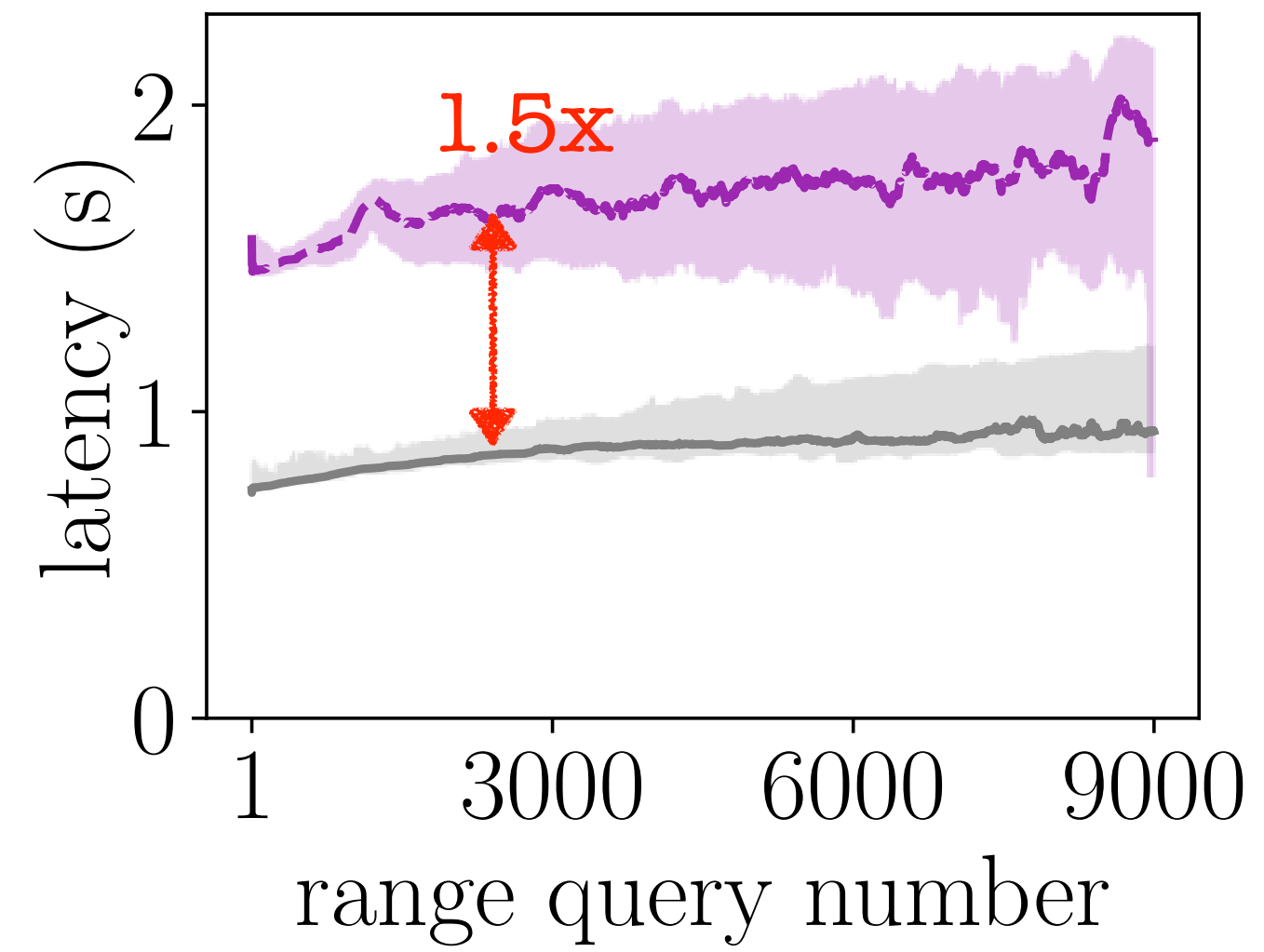
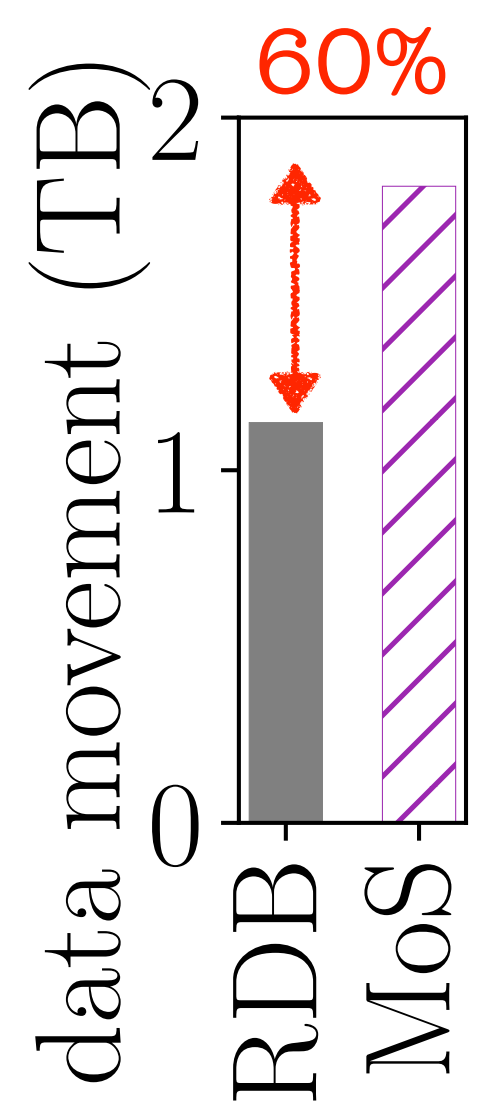
M: buffer size
E: entry size
T: size ratio

Merge-on-Scan

I: inserts
U: updates
S: range queries

M=4MB E=128B T=6 I=8.4M U=8.4M S=9K

— RocksDB (RDB) - - - Merge-on-Scan (MoS) ■ RocksDB ▨ Merge-on-Scan

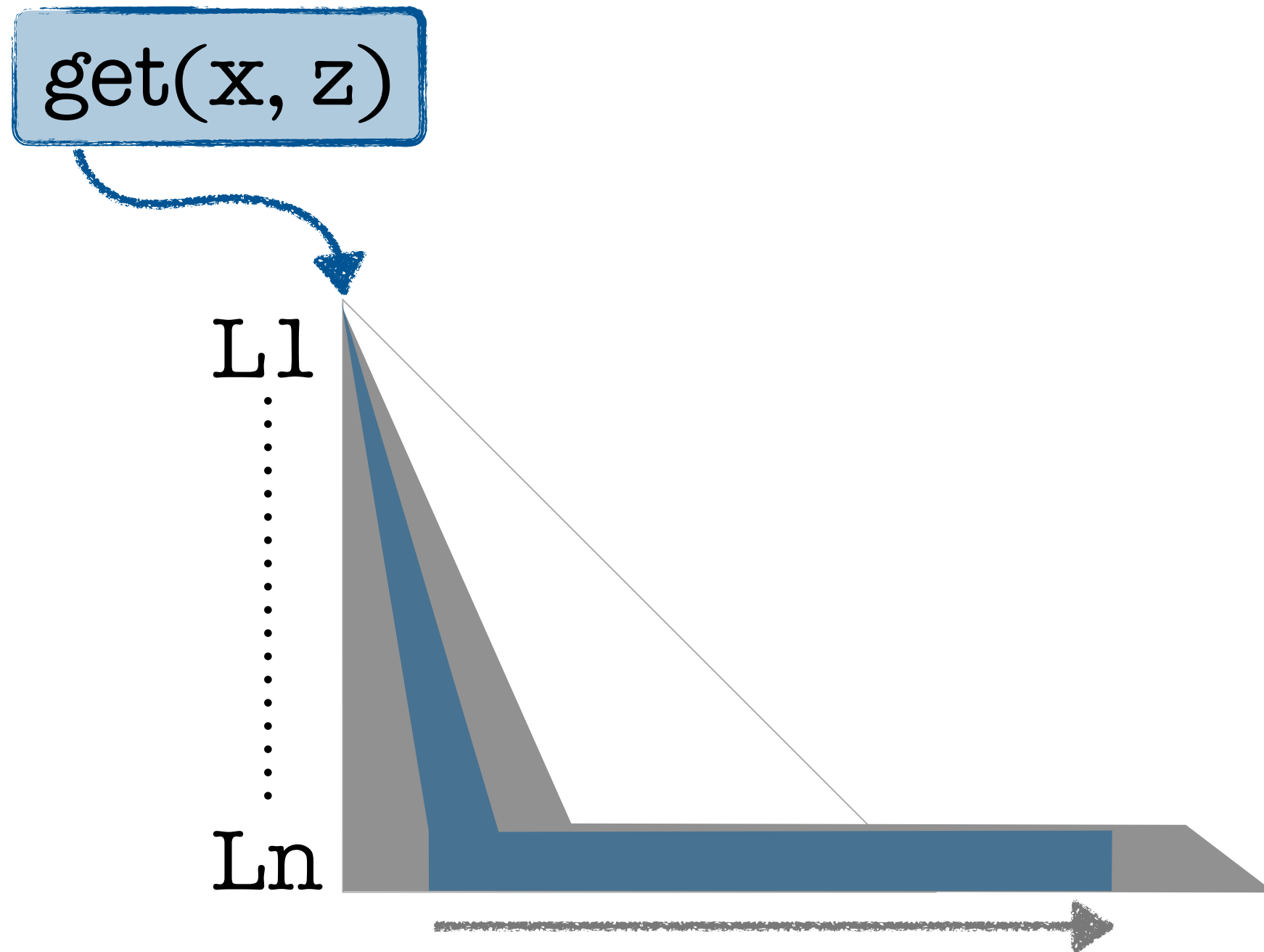


MoS is moving (reading & writing) more data overall

range queries are 1.5x slower than RocksDB

MoS reduces space amplification by 25% and reads up to 17% less data during range queries

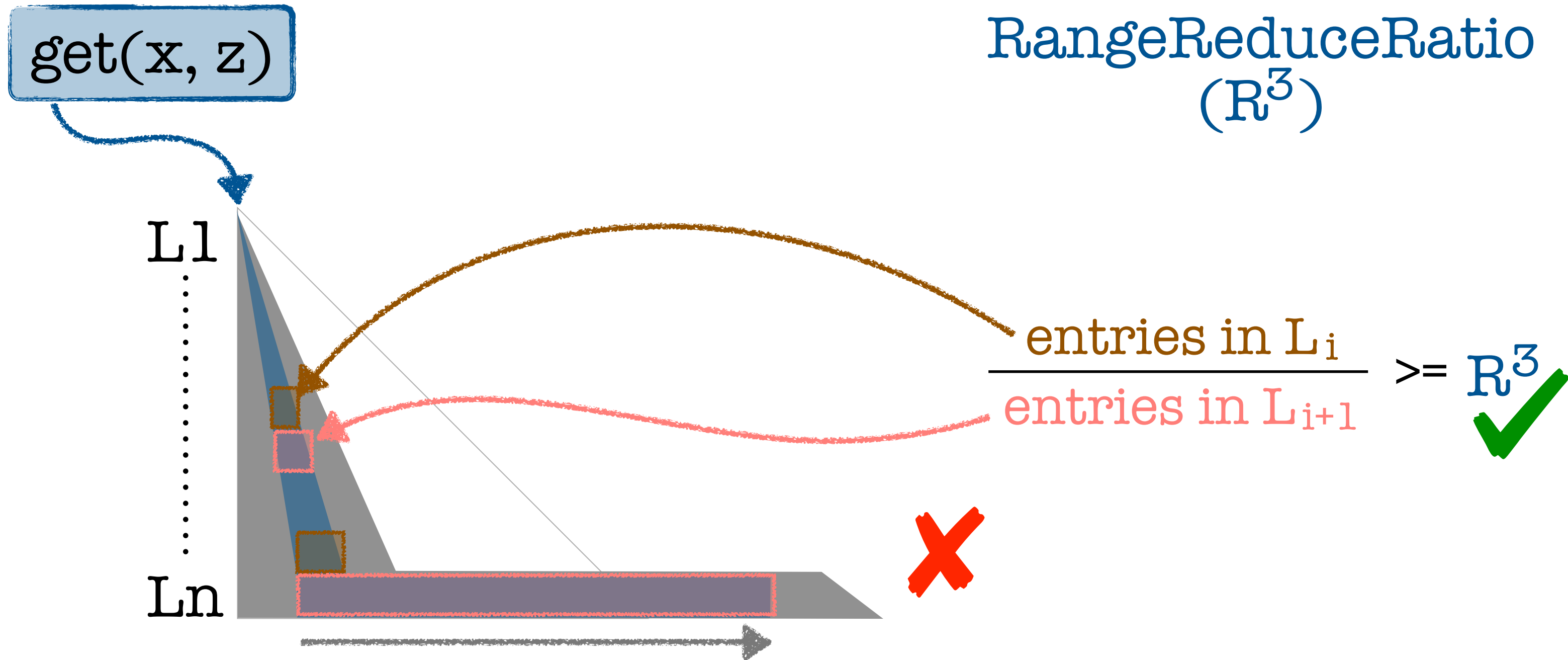
Bounded-Merge



! slower RQ

! increases write amp.

Bounded-Merge



! slower RQ

! increases write amp.

Bounded-Merge

get(x, z)

RangeReduceRatio
(R^3)

L1
⋮
Ln

$$\frac{\text{entries in } L_i}{\text{entries in } L_{i+1}} \geq R^3 \quad \checkmark$$



L1
⋮
Ln



slower RQ



increases write amp.

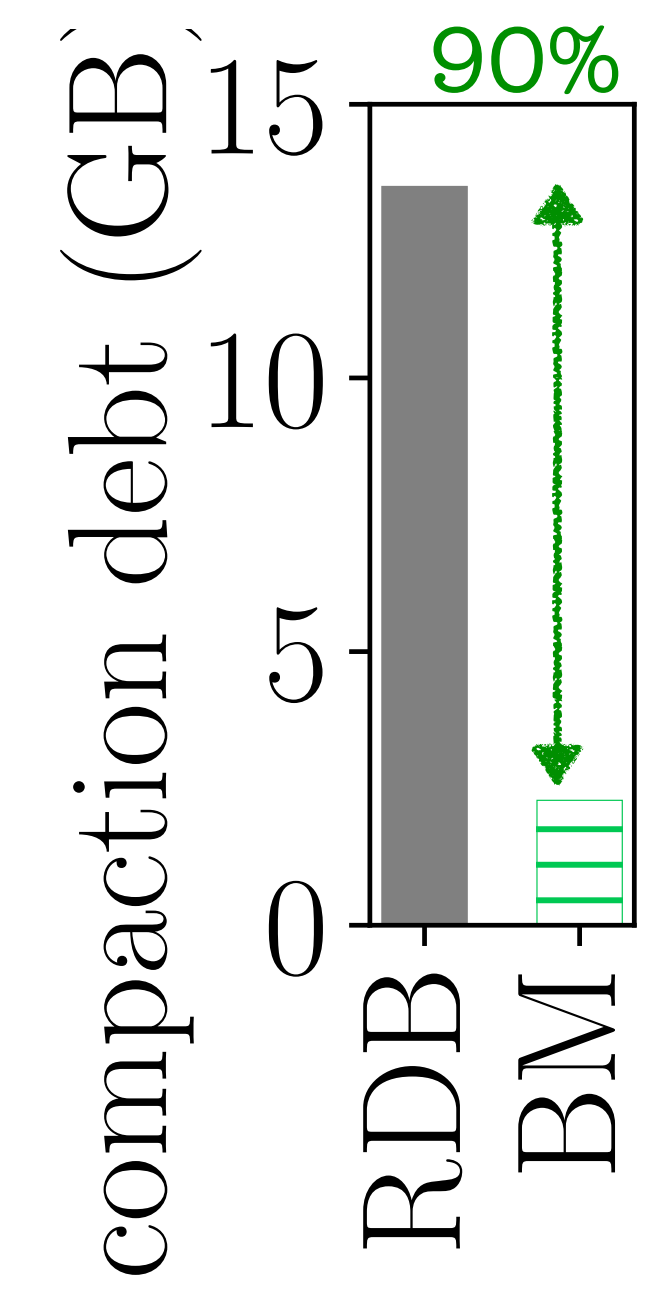
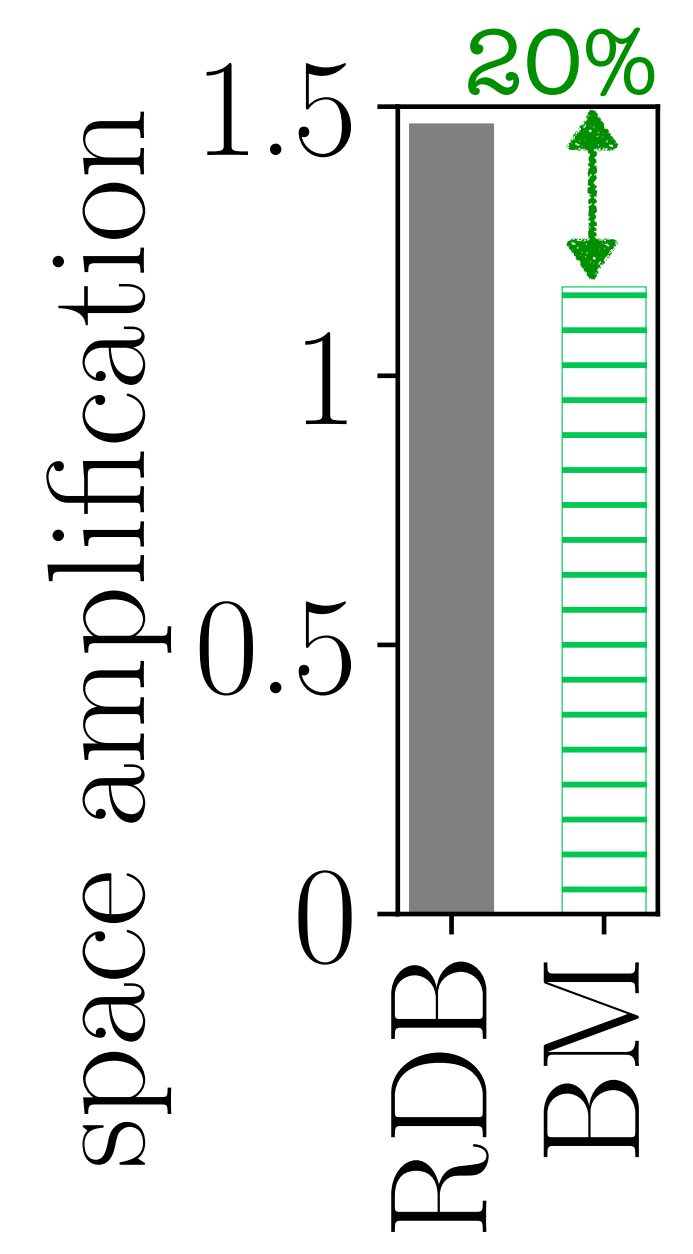
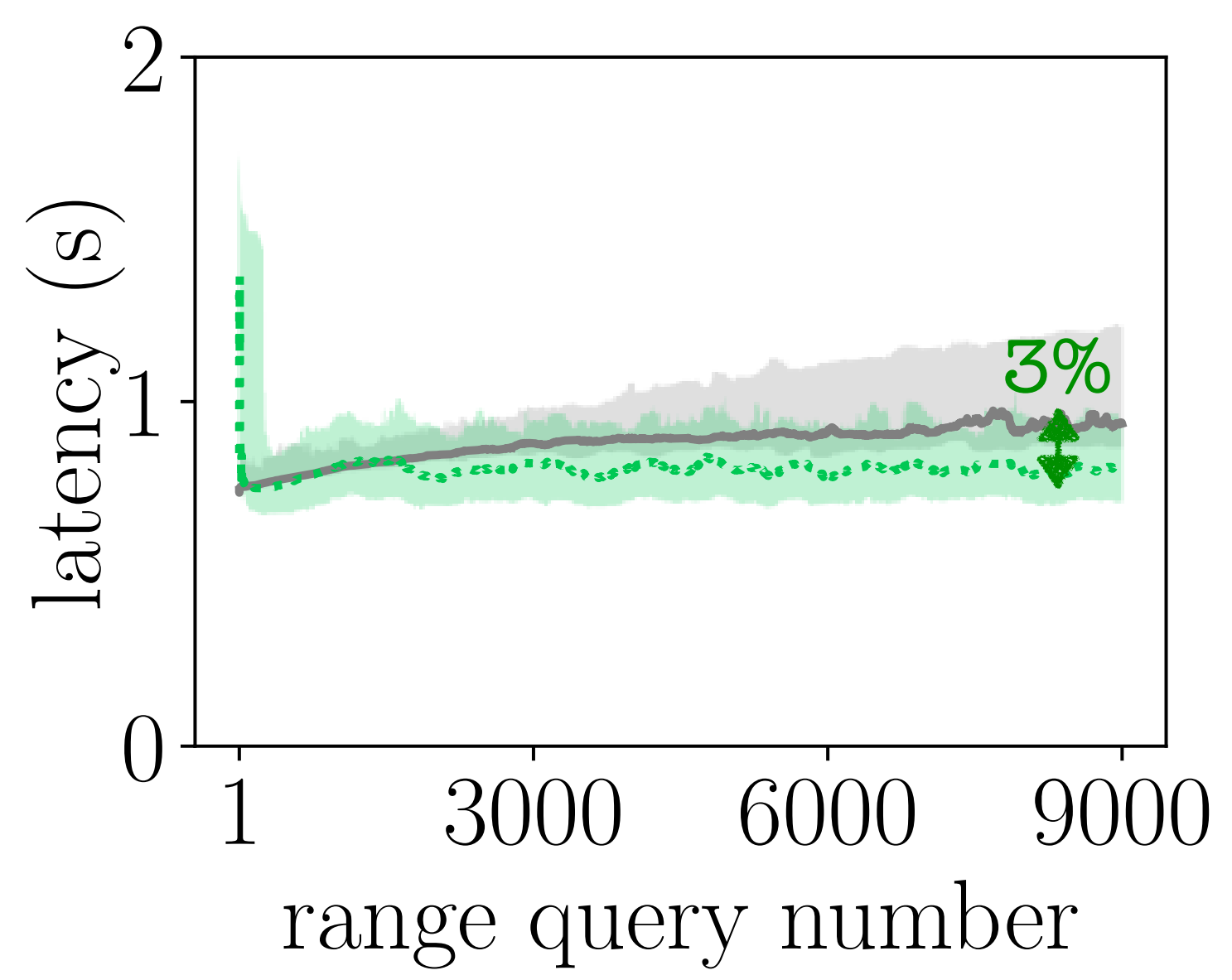
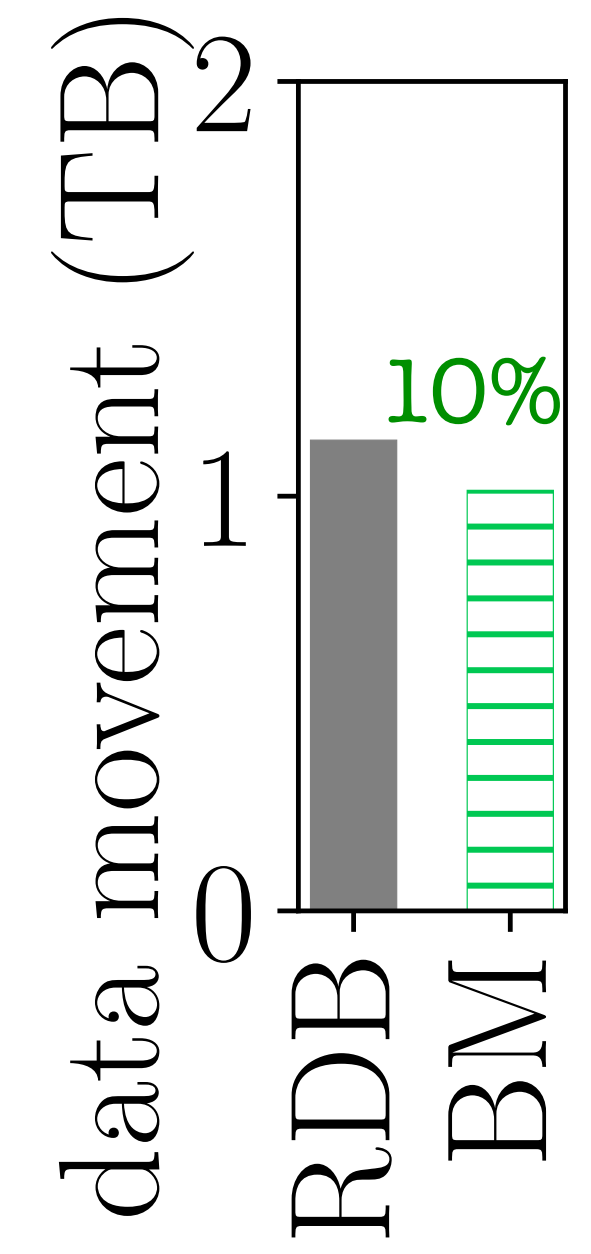
M: buffer size
E: entry size
T: size ratio

Bounded-Merge

I: inserts
U: updates
S: range queries

M=4MB E=128B T=6 I=8.4M U=8.4M S=9K s=0.1

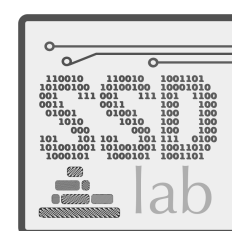
— RocksDB (RDB) Bounded-Merge (BM) ■ RocksDB ▤ Bounded-Merge



BM is moving 10% less data than RocksDB

range queries are 3% faster in BM than RocksDB

BM offers 20% better space amp. & 90% less compaction debt



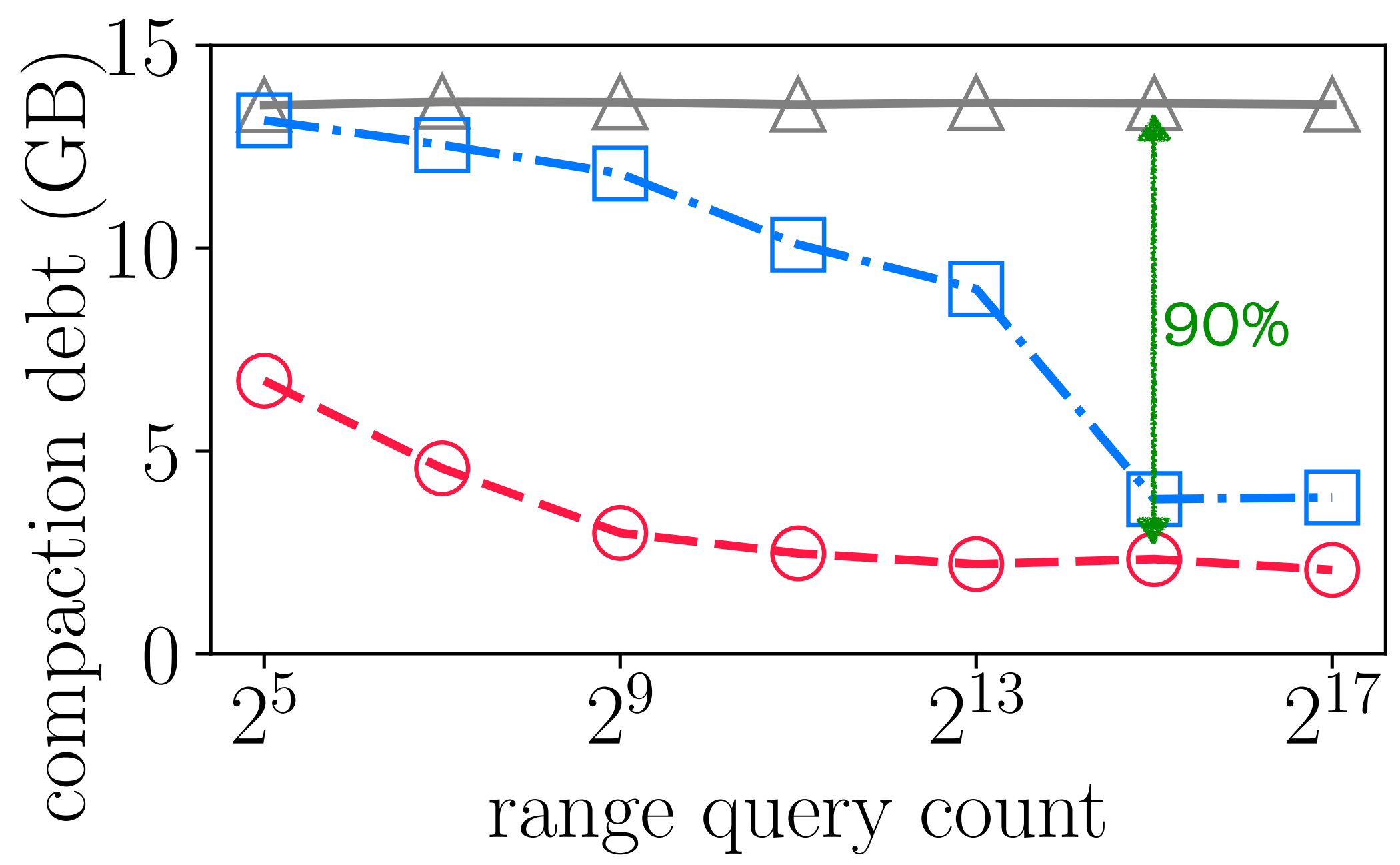
M: buffer size
E: entry size
T: size ratio

Experimental Evaluation

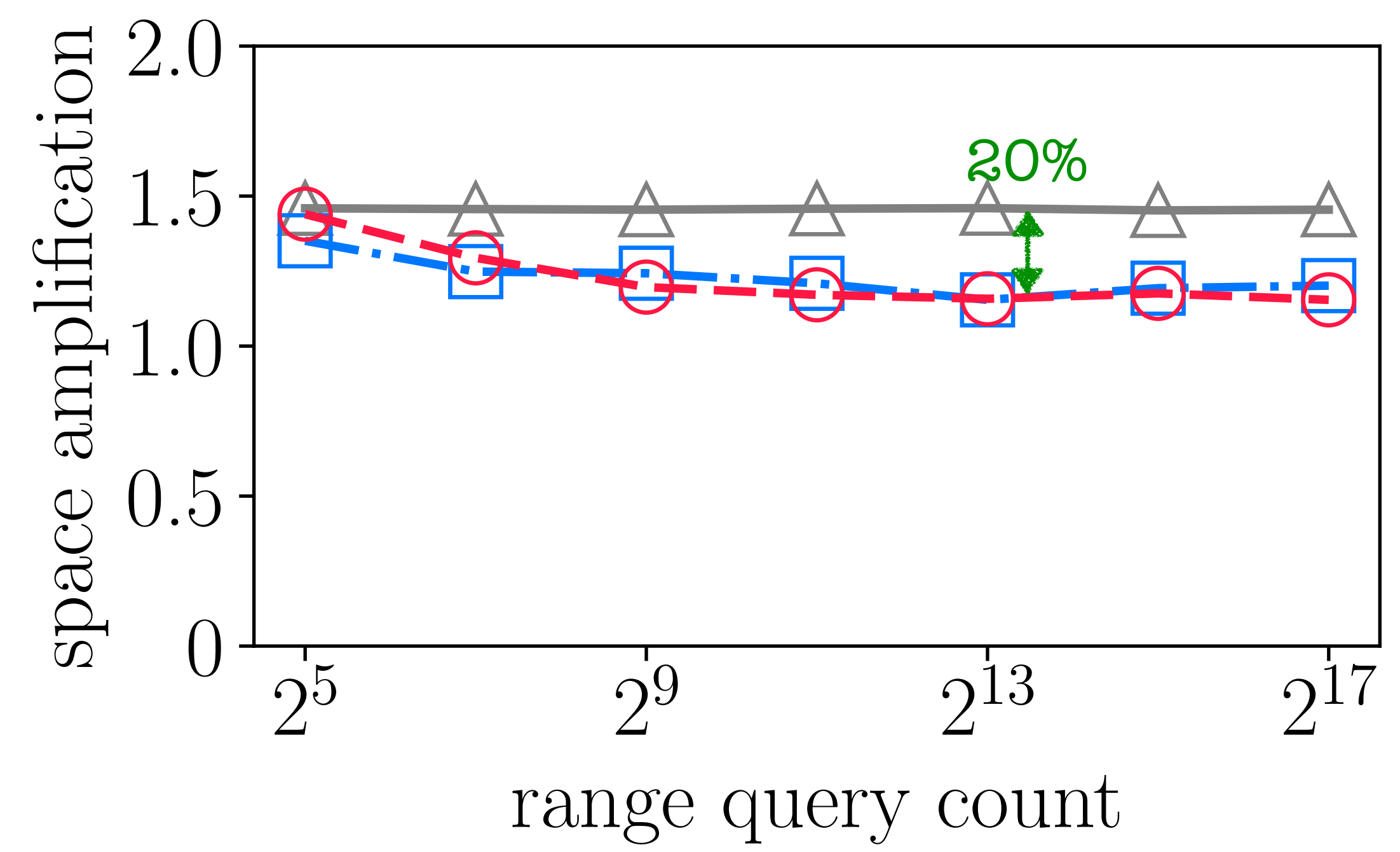
I: inserts
U: updates
S: range queries

M=4MB E=128B T=6 I=8.4M S+U=8.4M s=0.1

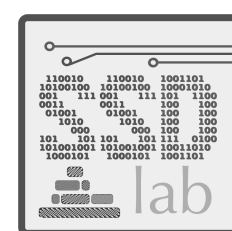
—△— RocksDB —□— SuccinctKV —○— RangeReduce



Reduces the future work of compaction by up to 90%



Improves the space amp. by up to 20% than RocksDB



RangeReduce: Query-Driven LSM Compactions

LSMs perform lots of repetitive work for the RQ-heavy workloads

Compactions reduce the logically invalid key, but not to the full potential

RQs compact data between a requested range, but it is thrown away

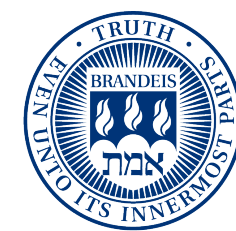
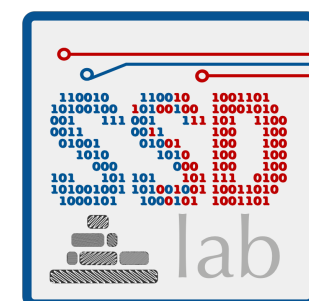
RangeReduce leverages RQs to reduce future work, improving the performance

Shubham Kaushik

<https://shubhamkaushik.com>

Manos Athanassoulis

Subhadeep Sarkar



Brandeis
UNIVERSITY

