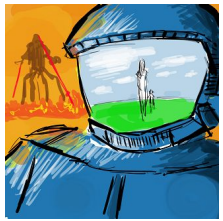
Log Structured Merge Tree



Pinglei Guo

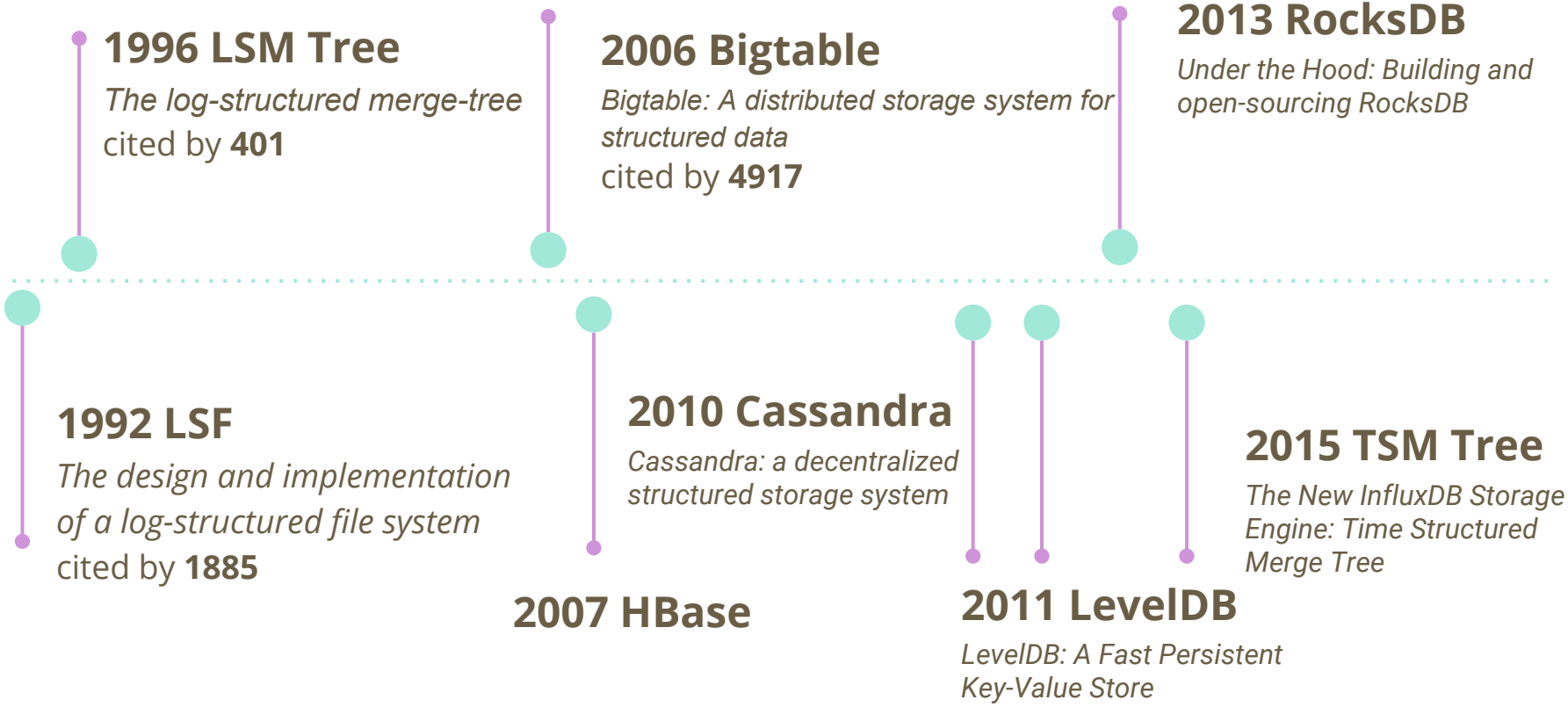
at15

at1510086

Agenda

- History
- Questions after reading the paper
- An example: Cassandra
- The original paper: Why & How & Visualization
- Suggested reading

History of LSM Tree



History of LSM Tree

What's the trend for Database?

- Data become larger, more write
- Non-Relational Databases emerge, HBase, Cassandra
- Database are also used for analysis and decision making

Databases using LSM Tree

- Bigtable
- Cassandra
- HBase
- PNUTS (from Yahoo! 阿里他爸)
- LevelDB && RocksDB
- MongoDB (wired tiger)
- SQLite (optional)
- InfluxDB

Databases using LevelDB/RocksDB

- Riak KV (TS)
- TiKV
- InfluxDB (before 1.0)
- MySQL (in facebook)
- MongoDB (in facebook's dismissed Parse)

Facebook: eat my own Rocks

Questions after reading the paper

- Do I still need WAL/WBL when I use log structured merge tree
- Is LSM Tree a data structure like B+ Tree, is there a textbook implementation
- Can someone explain the rolling merge process in detail
- Databases using LSM Tree often have the concept of column family, is it an alias for Column Database

Quick Answers

- Do I still need WAL/WBL when I use log structured merge tree
  Yes
- Is LSM Tree a data structure like B+ Tree, is there a textbook implementation
  No
- Can someone explain the rolling merge process in detail
  I will try
- Databases using LSM Tree often have the concept of column family, is it an alias for Column Database

  No, JavaScript != Java + Script

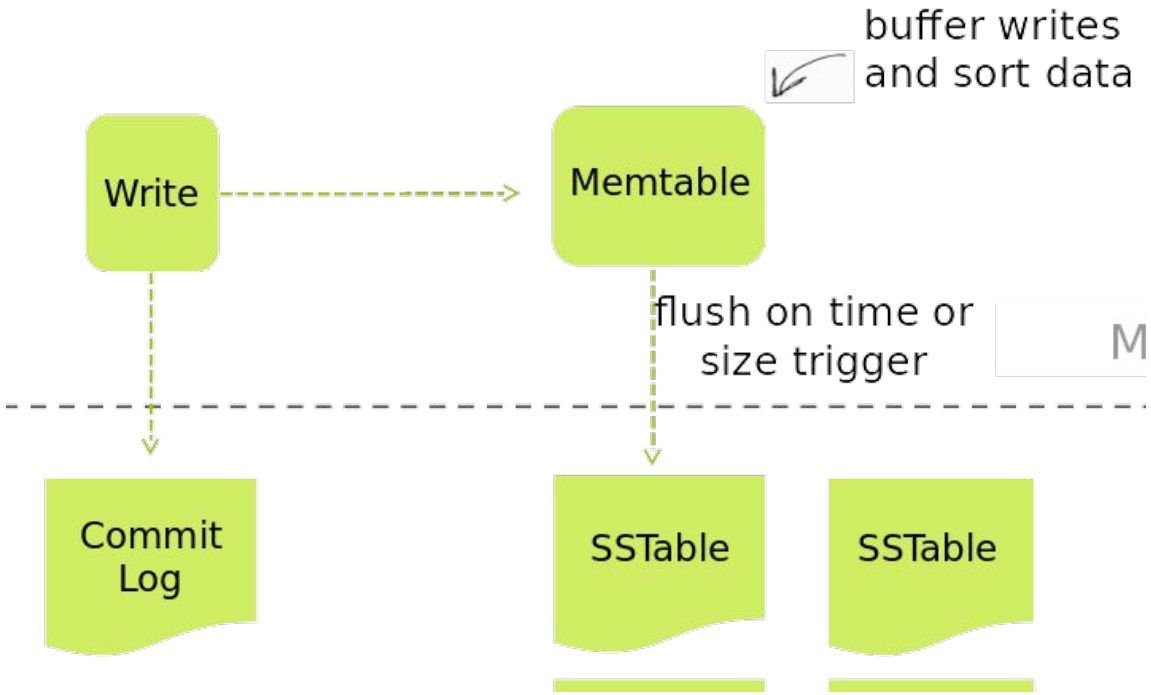
Cassandra as first example

Why? (not O'Neil 96, Bigtable, LevelDB)

why we pick Cassandra as first example?

1. It give us a high level overview of a **full** real system
2. It is easier to understand than original paper
3. It is battle tested
4. It is open source

Cassandra Write

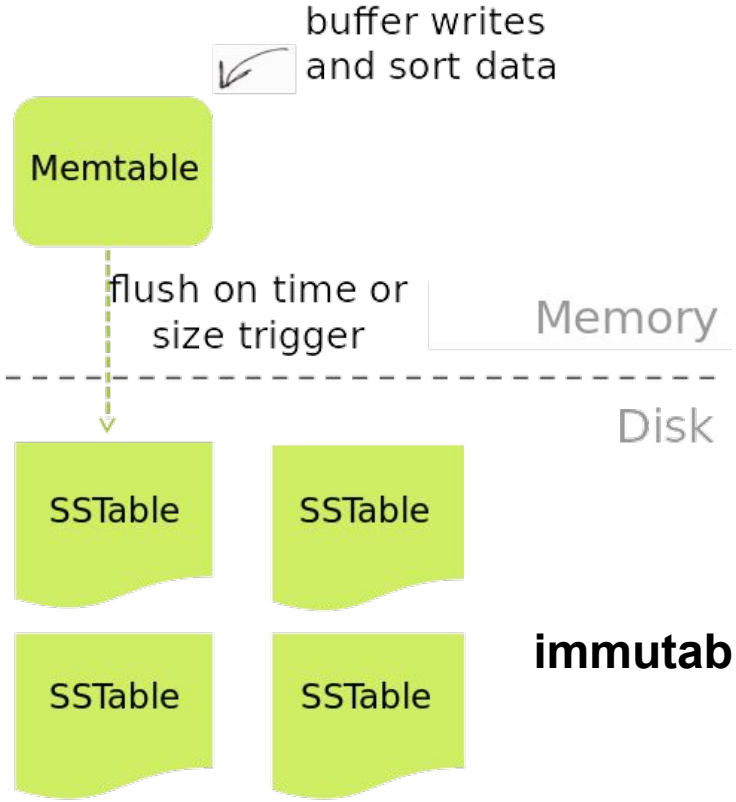


Write goes to

- Commit Log (WAL)
- Memtable (C0 in paper)

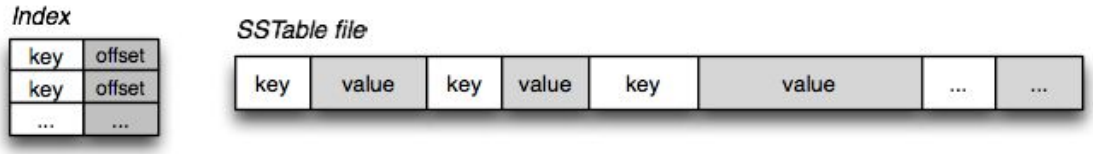
Operations return before the data is written to disk (Fast)

Cassandra ‘Merge’



- Memtable are dumped to disk as SSTable
- SSTable are merged by background process

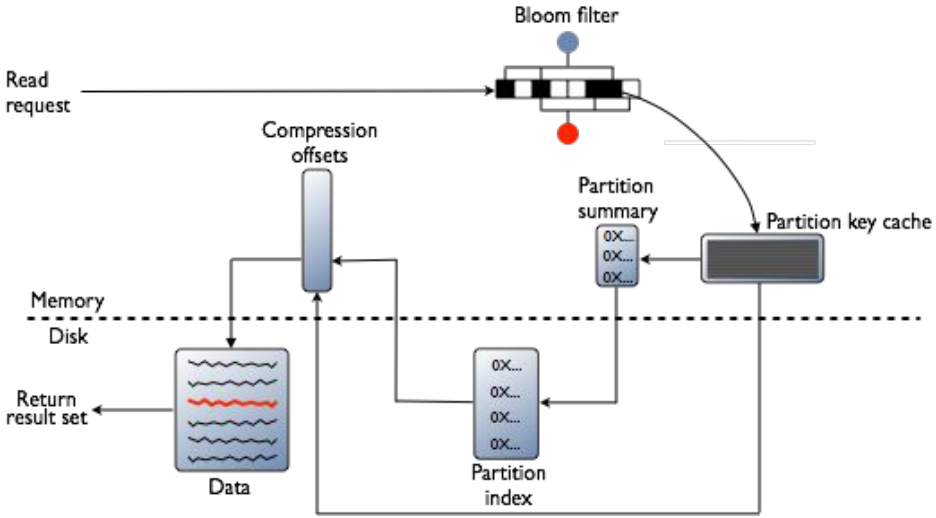
SSTable: Sorted String Table



immutable

- Bloom Filter
- Index
- Data

Cassandra Read (simplified)



- Read from MemTable
- use Bloom filter to identify SSTables
- Load SSTable index
- Read from multiple SSTables
- Merge the result and return

O'Neil 96 The LSM tree

Its name leads to confusion

- **Log** structured merge tree is not log like WAL
- **Log** comes from log structured file system
- LSM **Tree** is a concept than a concrete implementation
- **Tree** can be replaced by other data structure like map
- More intuitive name could be buffered write, multi level storage, write back cache for index

Log is borrowed, Tree can be replaced, Merge is the king

O'Neil 96 The LS Merge tree

Let's talk about Merge

Merge is the subtle part (that I don't understand clearly)

Two Merges

- Post-Write: Merge fast (small) level to slow (big) level
- Read: Read from both fast level and slow level and return the merged result

Merge Sort

- A new array need to be allocated
- Two sub array must be sorted before merge

O'Neil 96 The LS Merge tree

Q1: Why we need to Merge?

A : Because we put data on different media

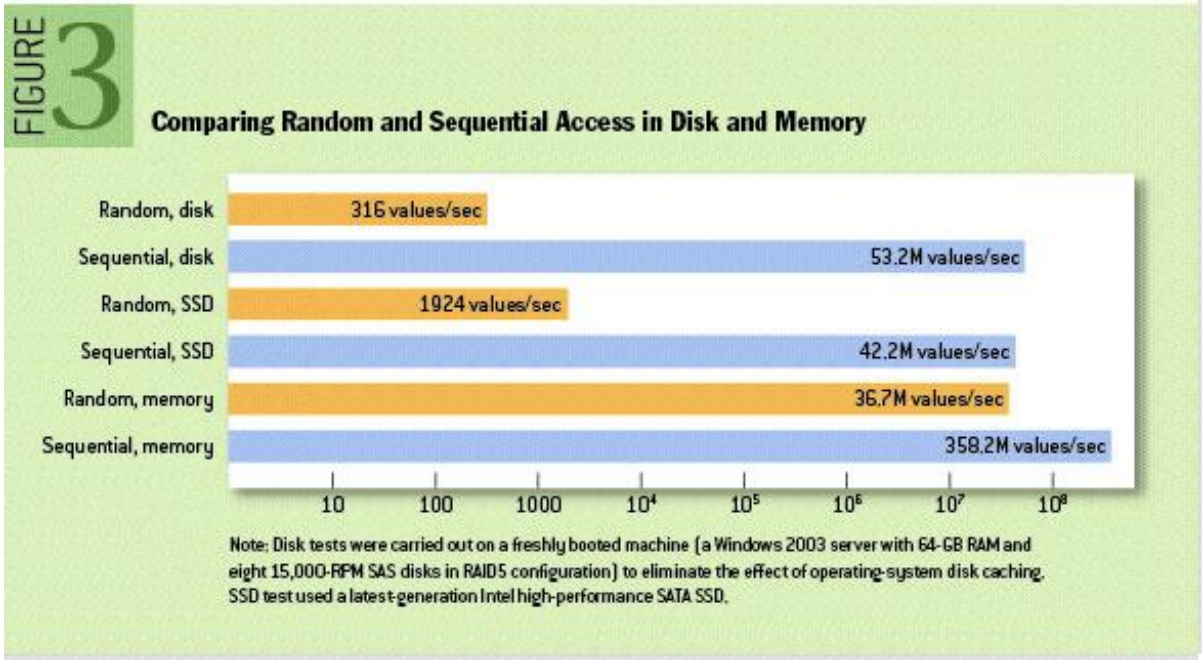
O'Neil 96 The LS Merge tree

1. Merge is needed because we put data on different media

Q2: Why put data on different media?

1. Speed & Access pattern

The 5 minutes rule



O'Neil 96 The LS Merge tree

1. Merge is needed because we put data on different media

Q2: Why put data on different media?

1. Speed & Access pattern
2. Price
3. Durability

- Tape
- HDD
- SSD
- RAM

O'Neil 96 The LS Merge tree

1. Merge is needed because we put data on different media

Q2: Why put data on different media?

1. Speed & Access pattern
2. Price
3. Durability

- Tape
- HDD
- SSD
- RAM

Other media? NVM?

O'Neil 96 The LS Merge tree

1. Merge is needed because we put data on different media

Q2: Why put data on different media?

1. Speed & Access pattern
2. Price
3. Durability

- Tape
- HDD
- SSD
- RAM

Other media? Distributed system is also ‘media’

O'Neil 96 The LS Merge tree

1. Merge is needed because we put data on different media

Q2: Why put data on different media?

1. Speed & Access pattern
2. Price
3. Durability

Distributed systems -> media that resist larger failure

- Natural disasters
- Human misbehave

- Fail of one machine
- Fail of entire datacenter
- Fail of a country
- Fail of planet earth

O'Neil 96 The LS Merge tree

1. Merge is needed because we put data on different media
2. Put data on different media to gain

    1. Faster Speed
    2. Lower Price
    3. Resistance to various level of Failures

Q3: How to merge?

O'Neil 96 The LS Merge tree

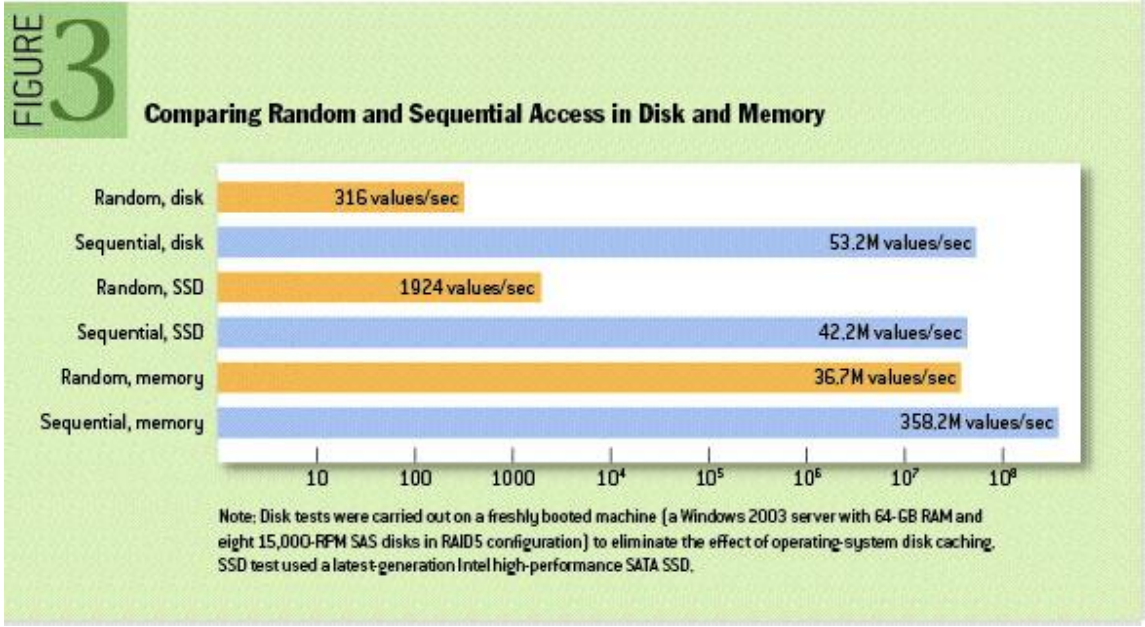
How to Merge is important

Principle: You don't write to the next level until you have to, and you write in the fastest way

- Batch
- Append

↓

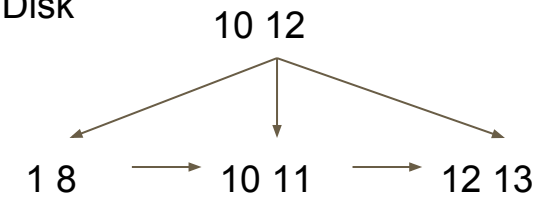
- speed up
- more efficient space usage



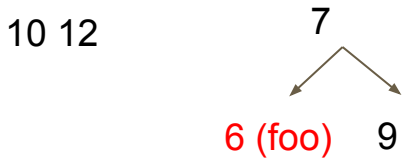
Client: Write <6, "foo">

Mem

Disk

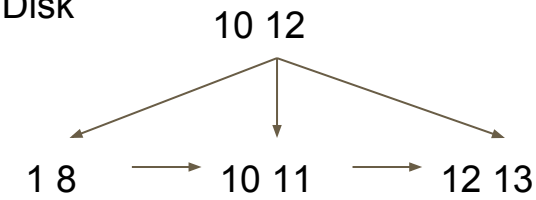


Before



Mem

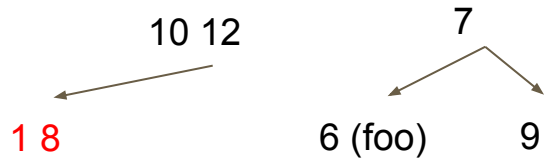
Disk



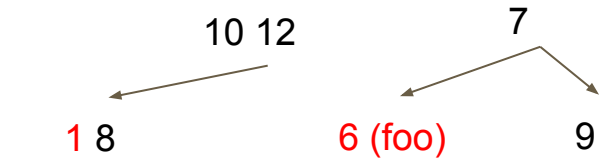
After

O' Neil 96

DB: I need to merge

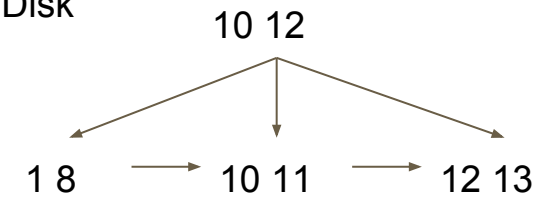


Mem

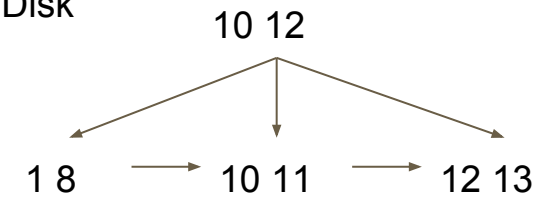


Mem

Disk



Disk

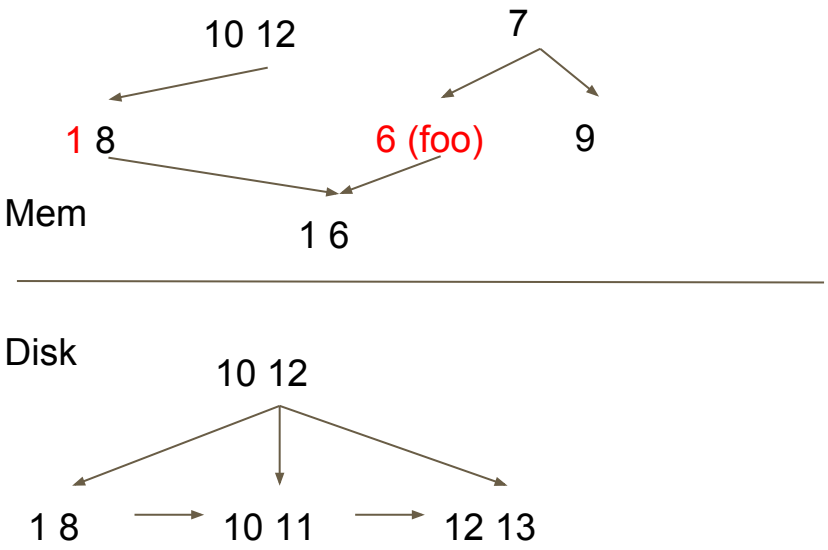


load leaf node into memory

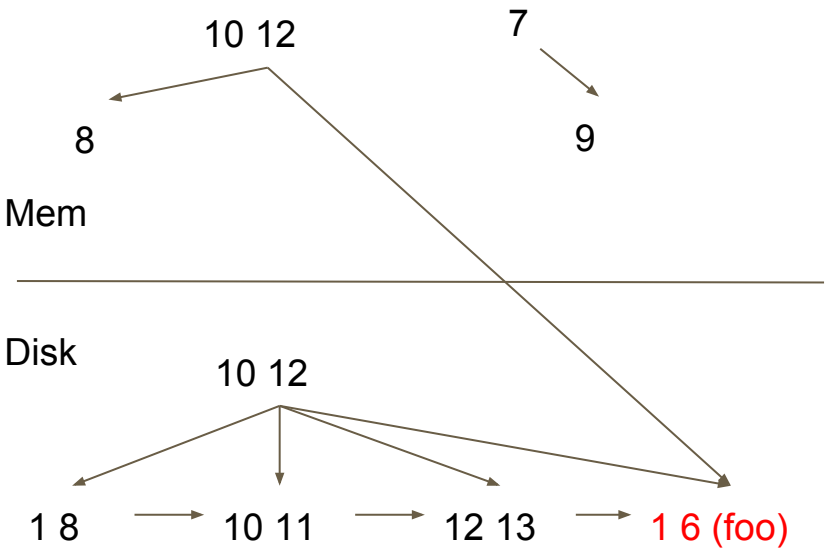
emptying, pick node

O' Neil 96

DB: I need to merge



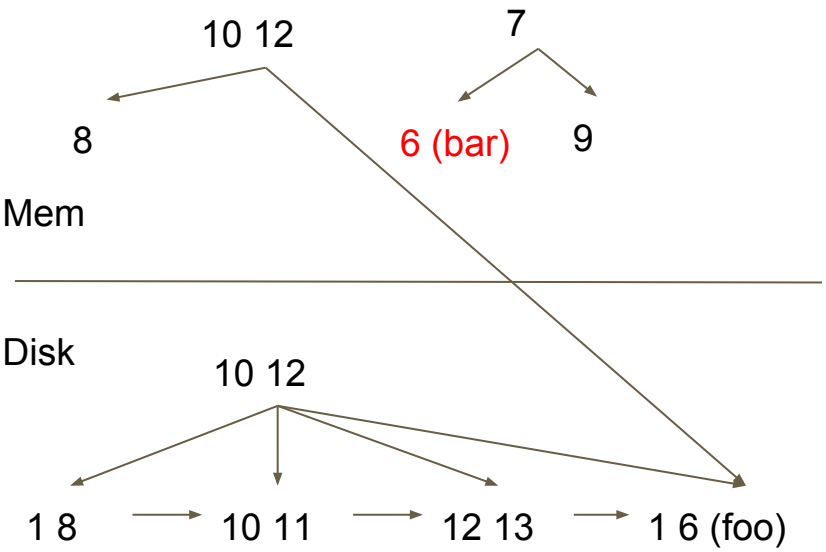
filling



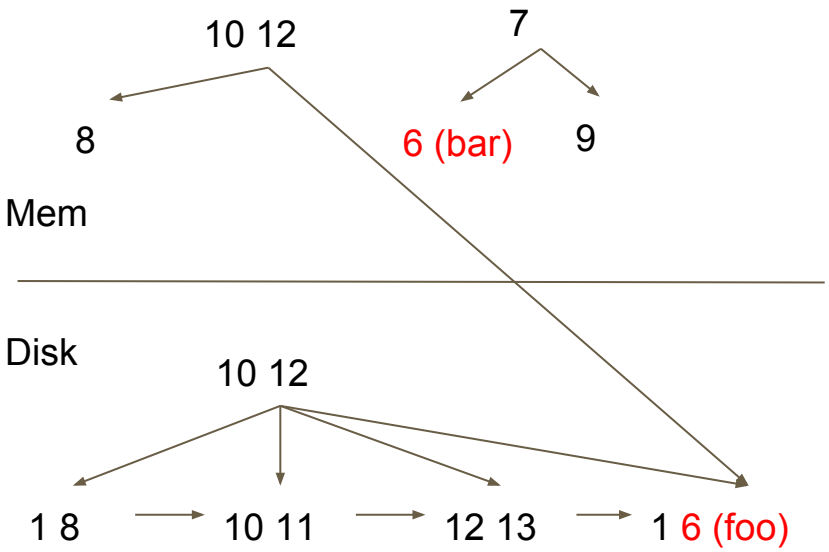
append to disk

O' Neil 96

Client: Write <6, "bar">



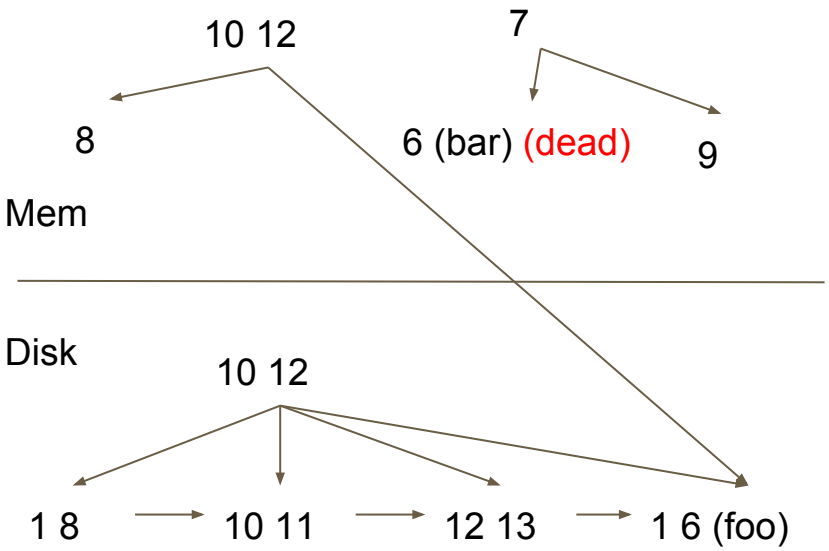
Client: Read <6, ?> [foo, bar]



Fetch from both level and return merged result

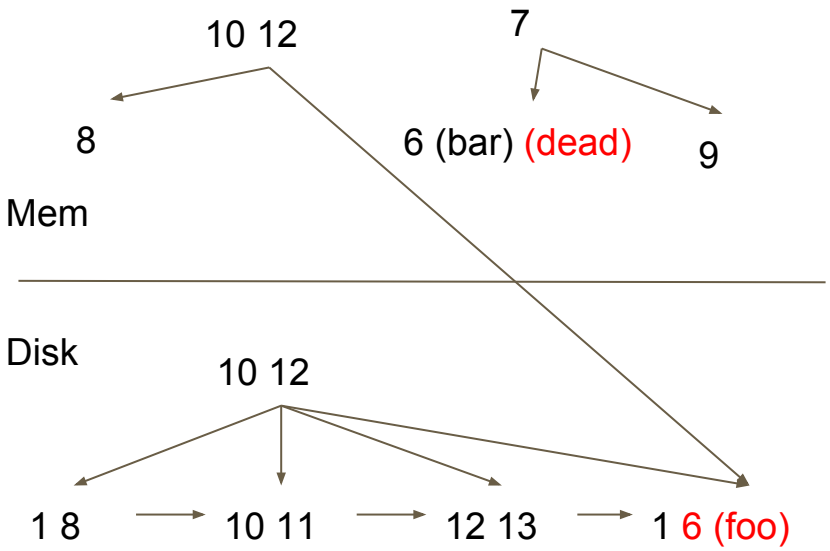
O' Neil 96

Client: Delete <6, "bar">



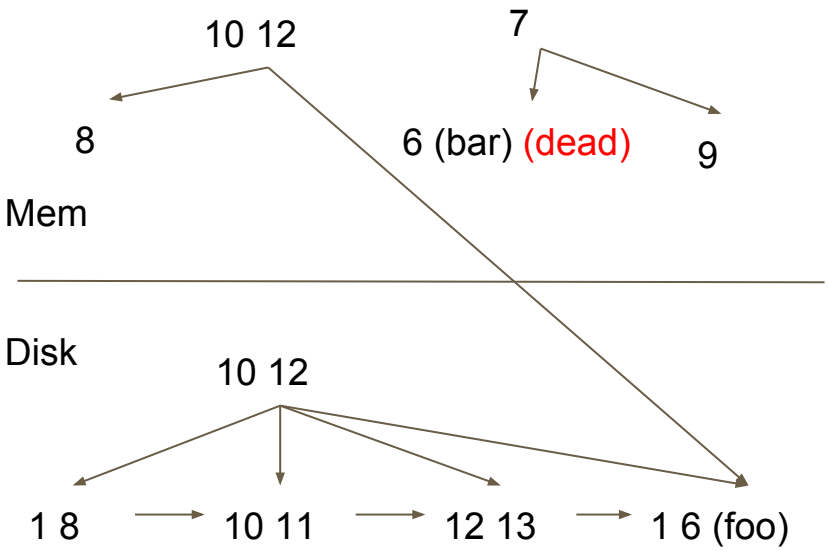
Client: Read <6, ?>

[foo]

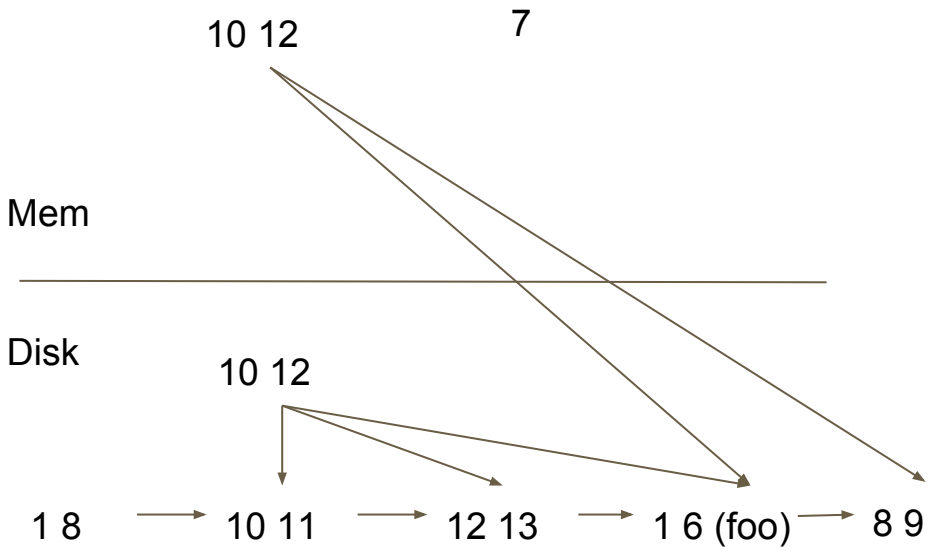


O' Neil 96

DB: I need to merge



Before



After

O' Neil 96

Client: Write <6, "I am foo">

7	Ha Ha
13	Excited

Mem

Disk

1	This	8	is	9	radom	10	gen	11	text	12	!

Before

6	I am foo
7	Ha Ha
13	Excited

Mem

Disk

1	This	8	is	9	radom	10	gen	11	text	12	!

After

Cassandra

DB: I need to dump

Mem

6	I am foo
7	Ha Ha
13	Excited

Disk

1	This	8	is	9	radom	10	gen	11	text	12	!

Before

Mem

Disk

6	I am foo	7	Ha Ha	13	Excited

1	This	8	is	9	radom	10	gen	11	text	12	!

After

Cassandra

DB: I need to compact

Disk

Before

6	I am foo	7	Ha Ha	13	Excited

1	This	8	is	9	radom	10	gen	11	text	12	!

After

1	This	6	I am foo	7	Ha Ha	8	is	9	radom	10	gen	11	text	12	!	13	Excited

Cassandra

Compare of O'Neil 96 and Cassandra

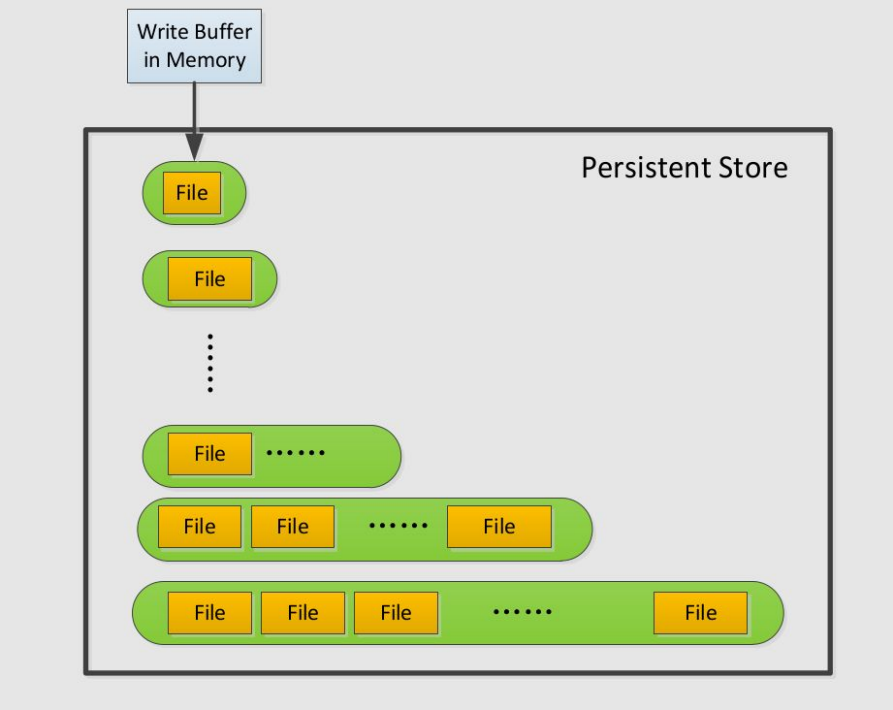
	O'Neil 96	Cassandra
in memory structure	AVL/2-3 Tree	Map
on disk structure	B+ Tree	SSTable, Index, Bloomfilter
level (component)	C_0, C_1 .... C_n	Memtable, SSTable
flush to disk when	Memory can’t hold	Memory can’t hold and/or timer
persist to disk by	Write new block (append)	dump new SSTable from Memtable (append)
merge is done at	Memory (empty, filling block)	Disk (Compaction in background)
concurrency control	Complex	SSTable is immutable, data have (real world) timestamp for versioning, updating value does not bother dump or merge
delete	Tombstone, delete at merge	Tombstone, delete at merge

O'Neil 96 The LS Merge tree

Summary

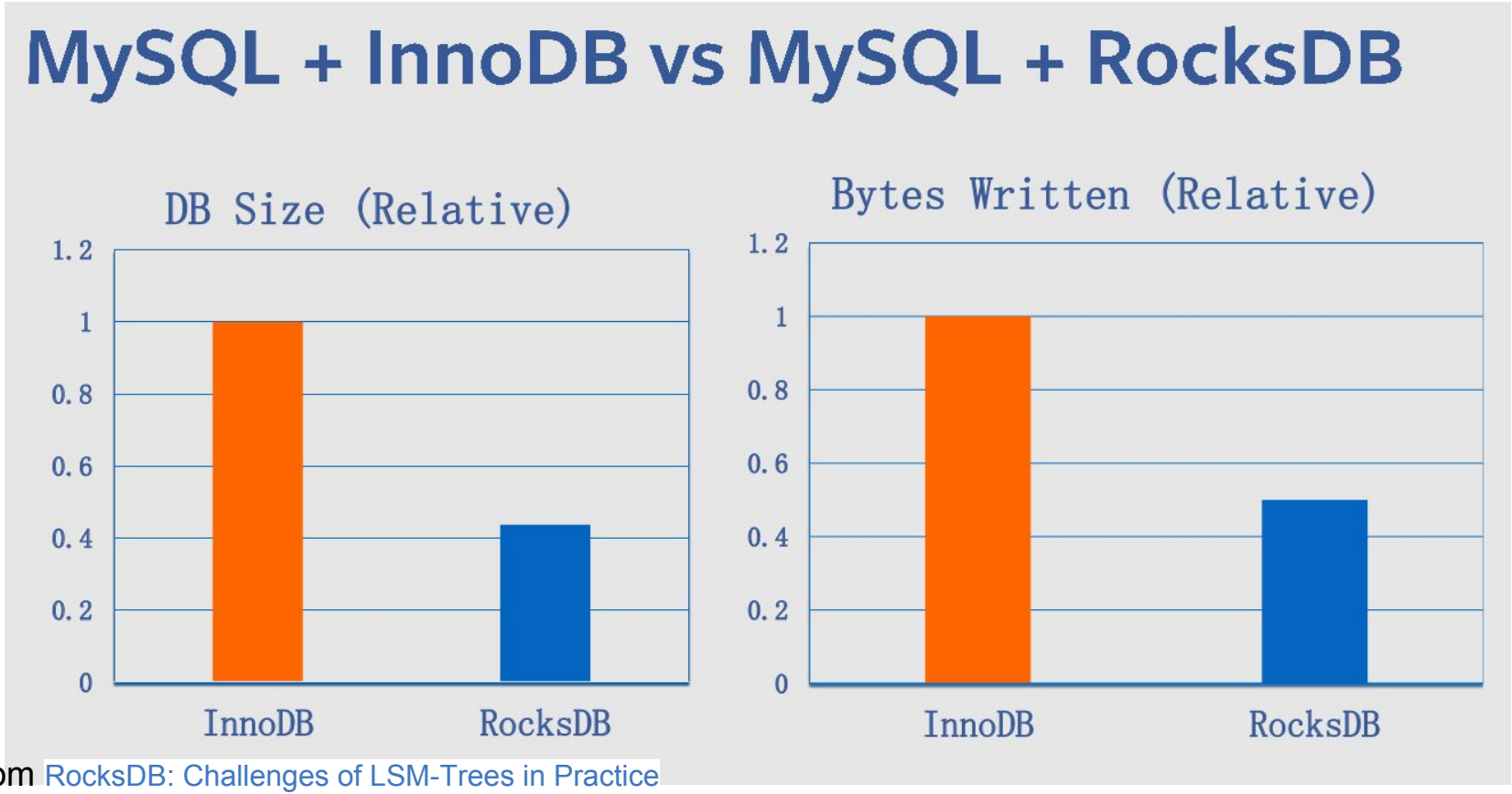
- Write to fast level
- Read from both fast and slow.
- Data is flushed from fast level to slow level when they are too big
- Real delete is defered to merge

LevelDB & RocksDB



From RocksDB: Challenges of LSM-Trees in Practice

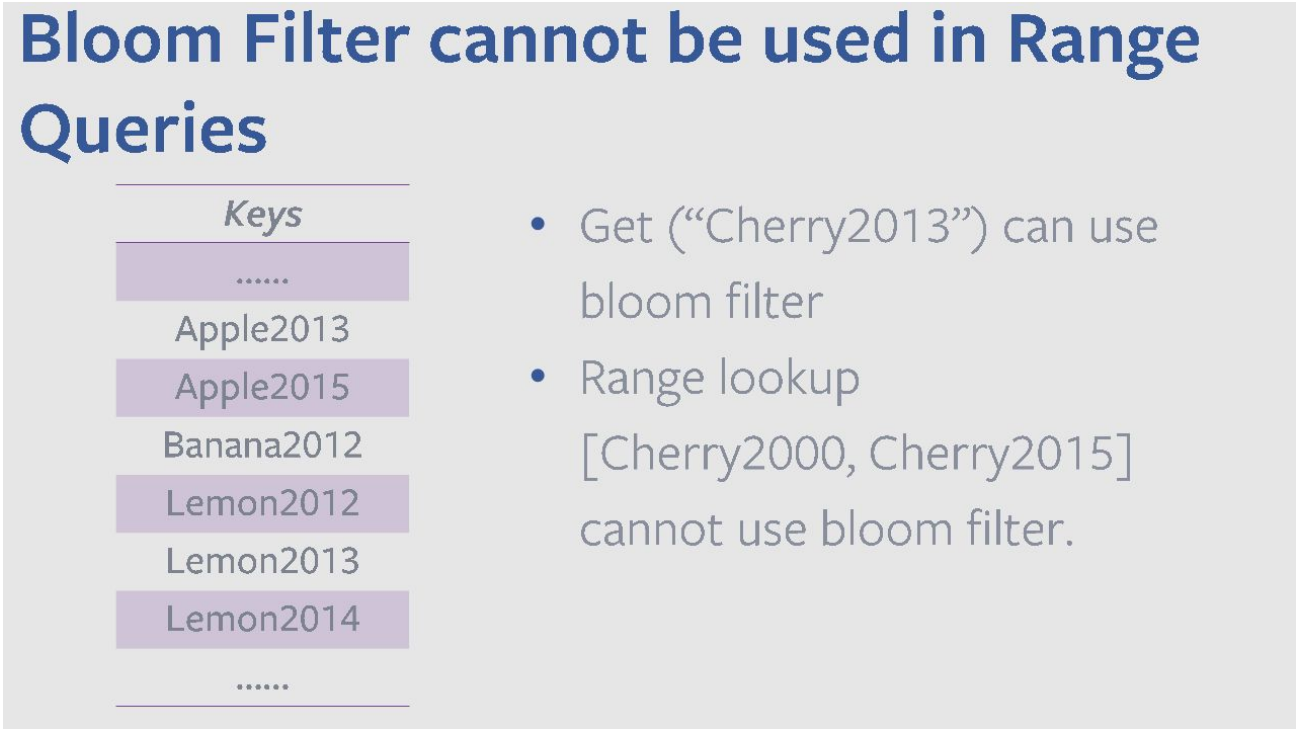
LevelDB & RocksDB



From RocksDB: Challenges of LSM-Trees in Practice

LevelDB & RocksDB

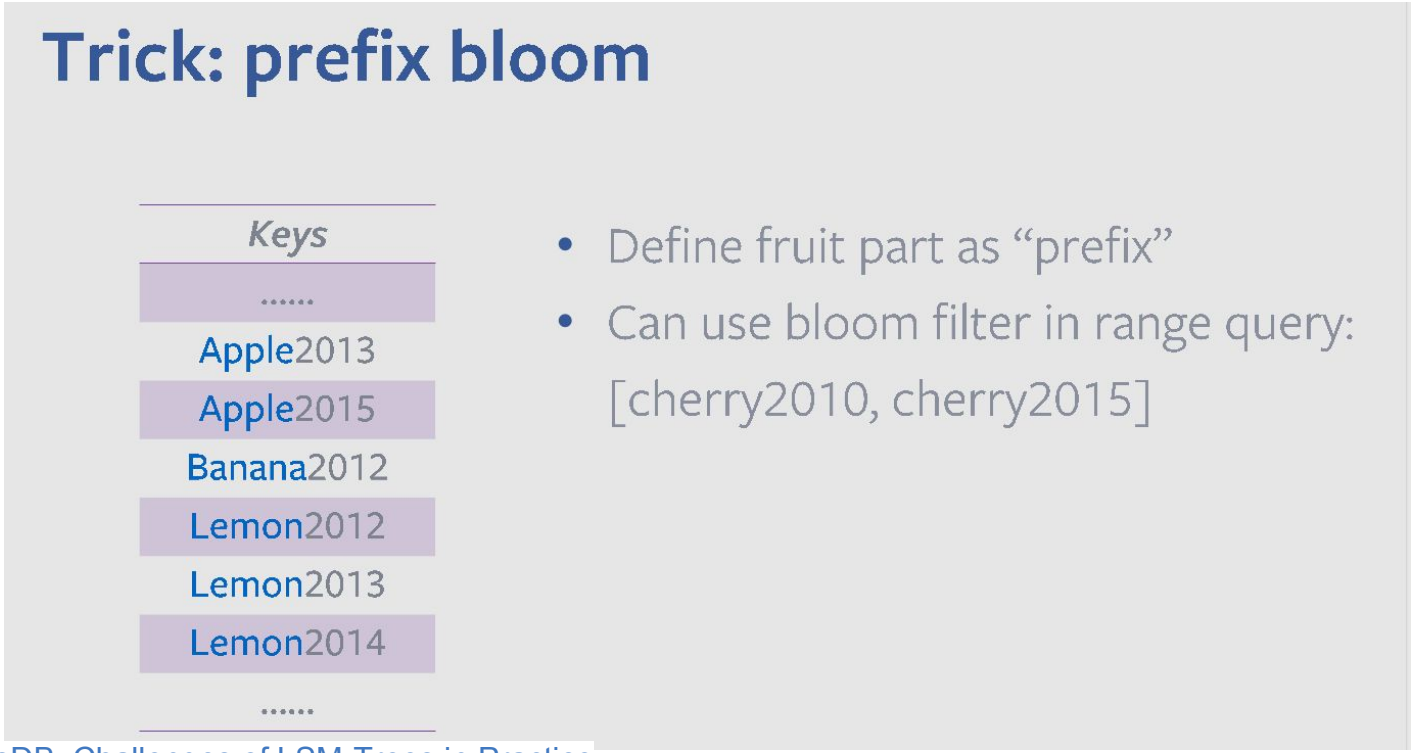
Bloom Filter for range queries



From RocksDB: Challenges of LSM-Trees in Practice

LevelDB & RocksDB

Bloom Filter for range queries



From RocksDB: Challenges of LSM-Trees in Practice

Full Answers

- Do I still need WAL/WBL when I use log structured merge tree
  Yes
- Is LSM Tree a data structure like B+ Tree, is there a textbook implementation
  No, it's how you use different data structure in different storage media
- Can someone explain the rolling merge process in detail
  I tried
- Databases using LSM Tree often have the concept of column family, is it an alias for Column Database

  No, see *Distinguishing Two Major Types of Column-Stores*

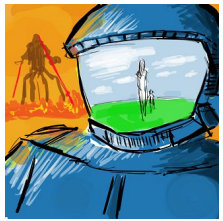
Reference & Suggested reading

1. SSTable and log structured storage leveldb
2. Notes for reading LSM paper
3. Cassandra: a decentralized structured storage system
4. Bigtable: A distributed storage system for structured data
5. RocksDB Talks
6. Pathologies of Big Data
7. Distinguishing Two Major Types of Column-Stores
8. Visualization of B+ Tree
9. Time structured merge tree
10. Code: Cassandra, LevelDB, RocksDB, Indeed LSM Tree, InfluxDB (Talk is cheap, show me the code)

code is cheap, show me the proof; proof is cheap, I just want to sleep

Thank You!

Happy weekend and Lunar New Year!



Pinglei Guo

at15

at1510086