Full Compactions in YugabyteDB & How to Schedule Them

John Meehan
Friday, Jan 27, 2023

YugabyteDB Friday Tech Talks

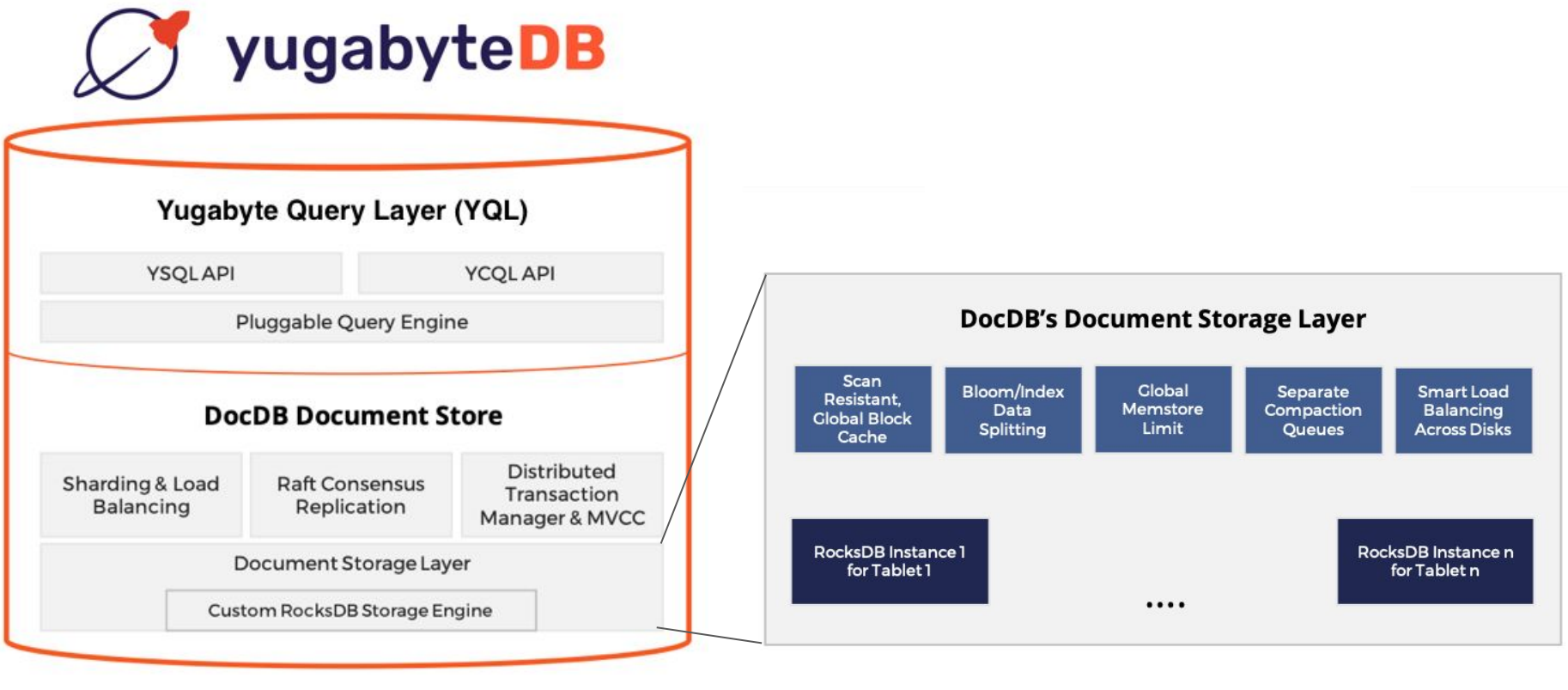
YFTT

Talks by Engineers for Engineers

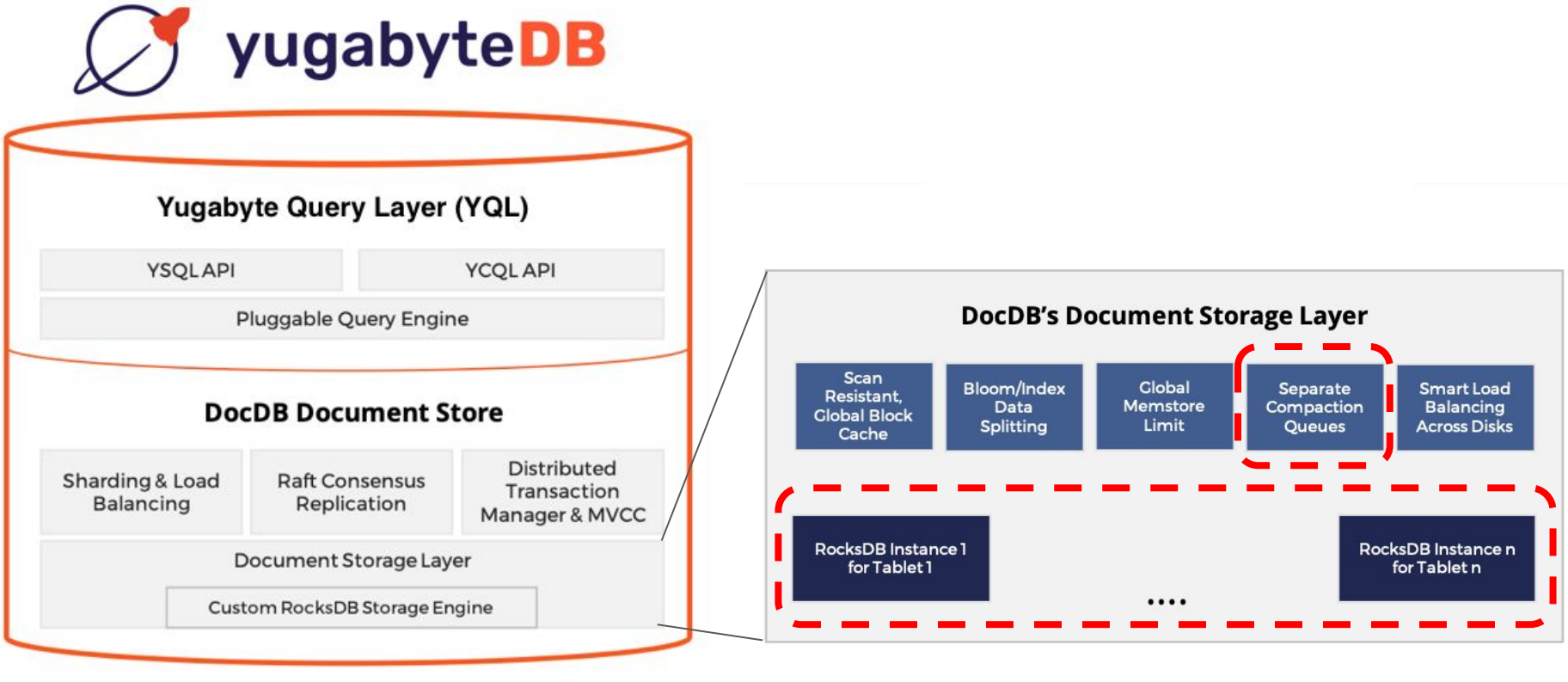
yugabyteDB

Layered architecture of YugabyteDB

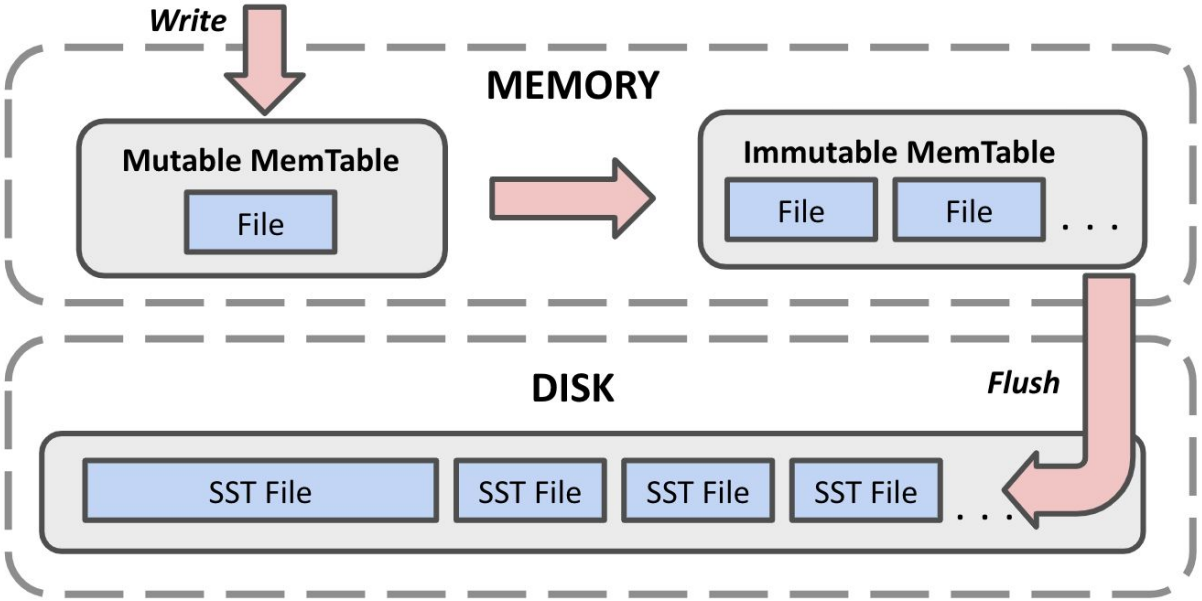
Storage layer of YugabyteDB



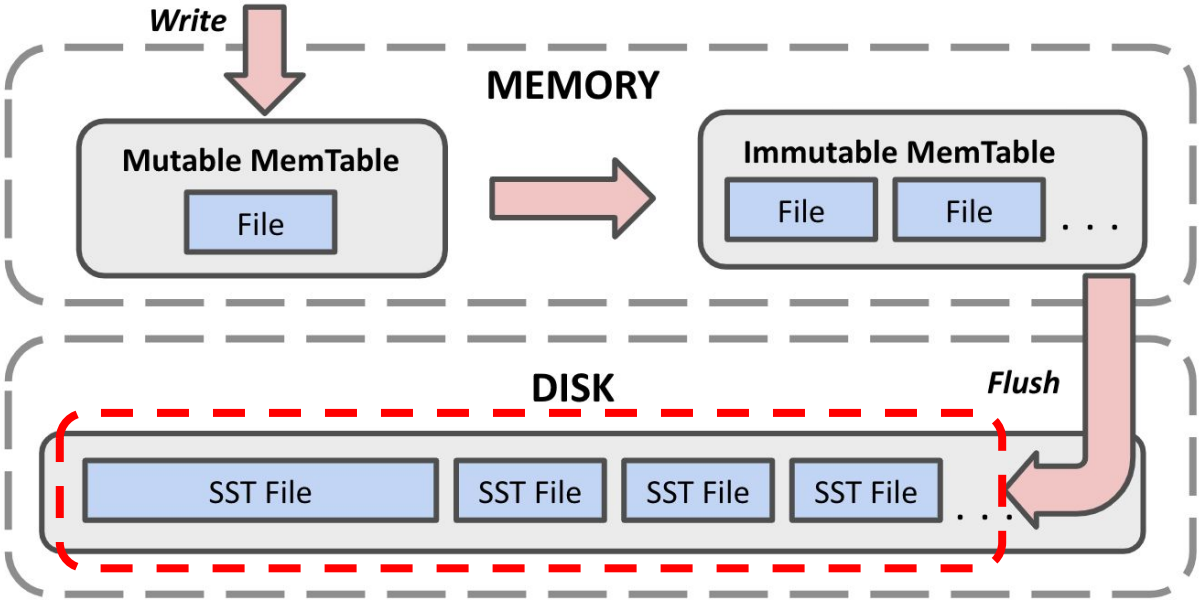
Storage layer of YugabyteDB



Writing to DocDB



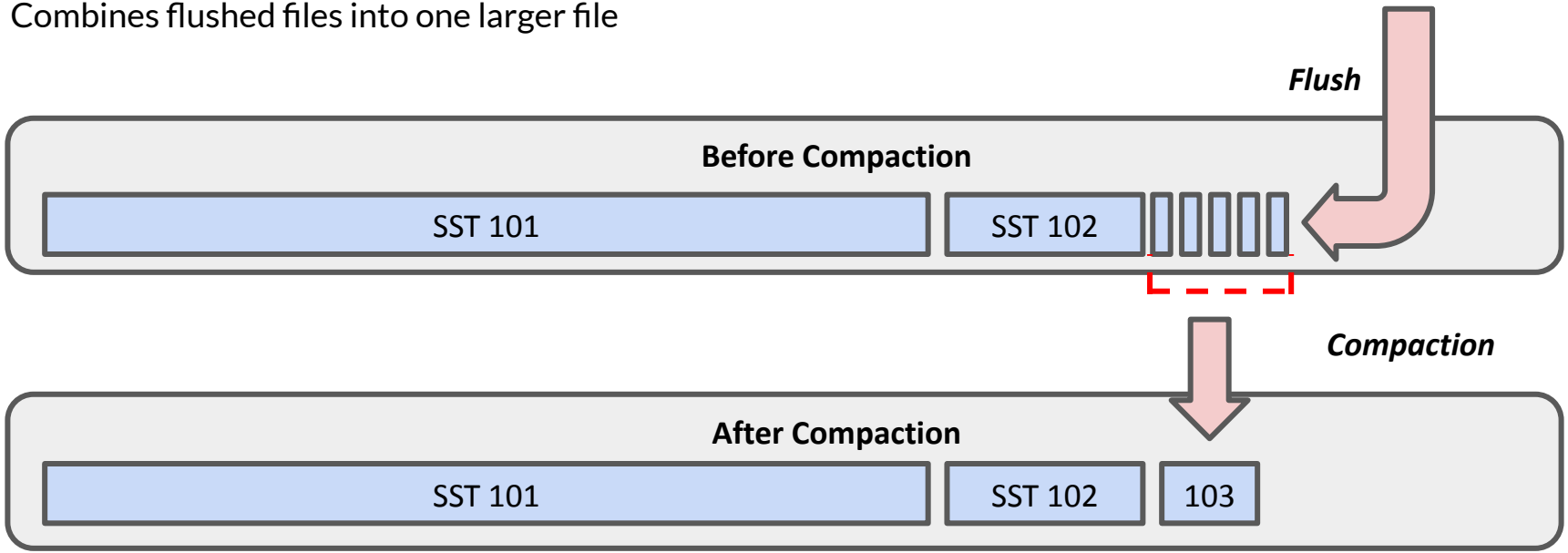
Writing to DocDB



Background Compactions

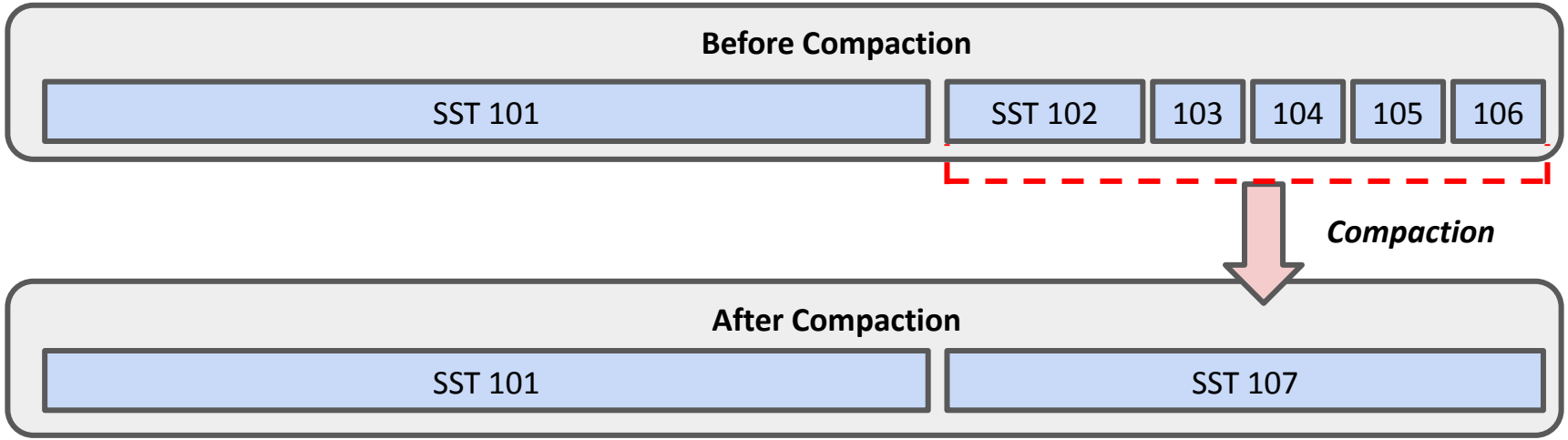
Universal compactions in DocDB

Combines flushed files into one larger file



Universal compactions in DocDB

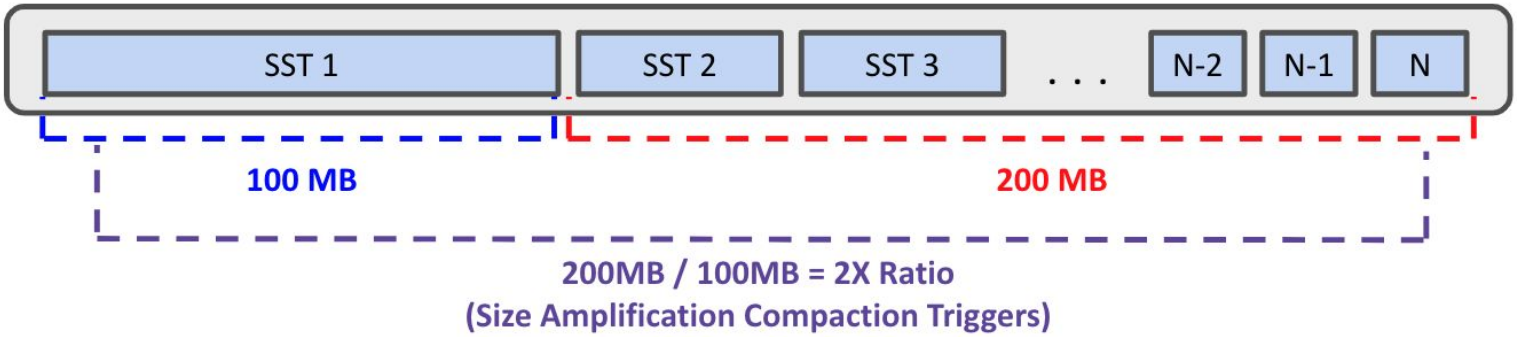
Combines smaller files into one larger file, garbage collecting any deleted/expired data



Algorithm 1: Size amplification

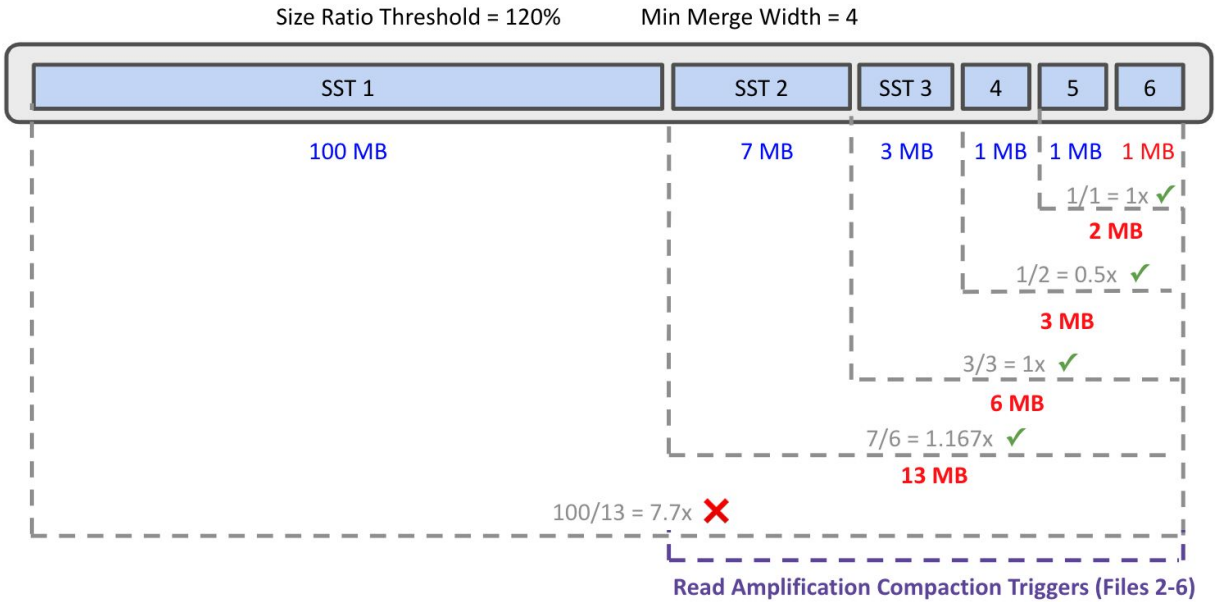
- Looks for *N* consecutive files such that the size of the first is less than **half** the combined size of the other *N-1*
  - i.e. size amplification threshold = **2x**

size amplification ratio = (size(R_N) + size(R_{N-1}) + ... size(R_2)) / size(R_1)

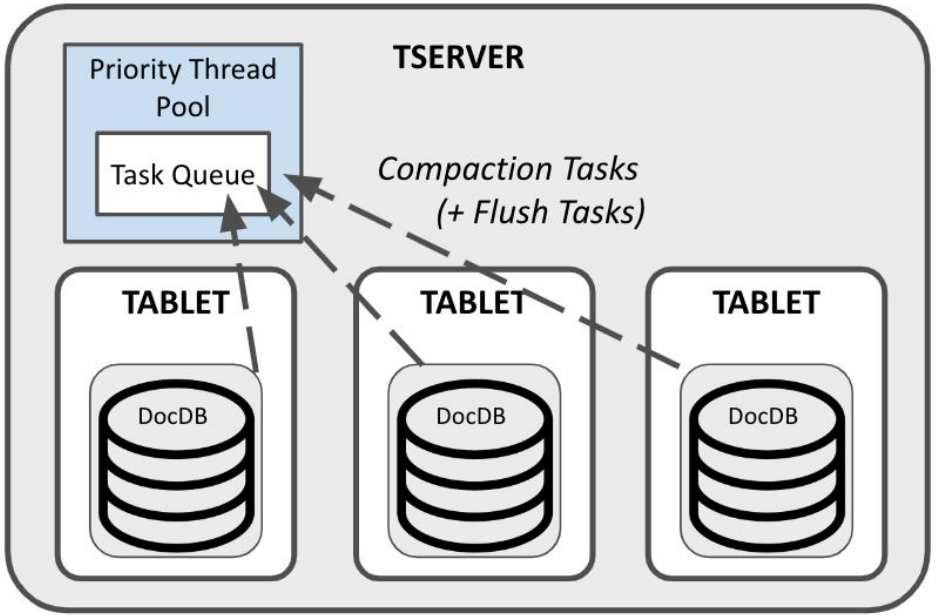


Algorithm 2: Read amplification

- Looks for at least ***[min merge width]*** files (as many as possible) such that the combined size of the latter *N-1* files are less than ***[size ratio threshold]*** from the first file's size
  - Default min merge width = **4** (***rocksdb_universal_min_merge_width*** = 4)
  - Default size ratio threshold = **120%** (***rocksdb_universal_compaction_size_ratio*** = 20)



Compaction prioritization



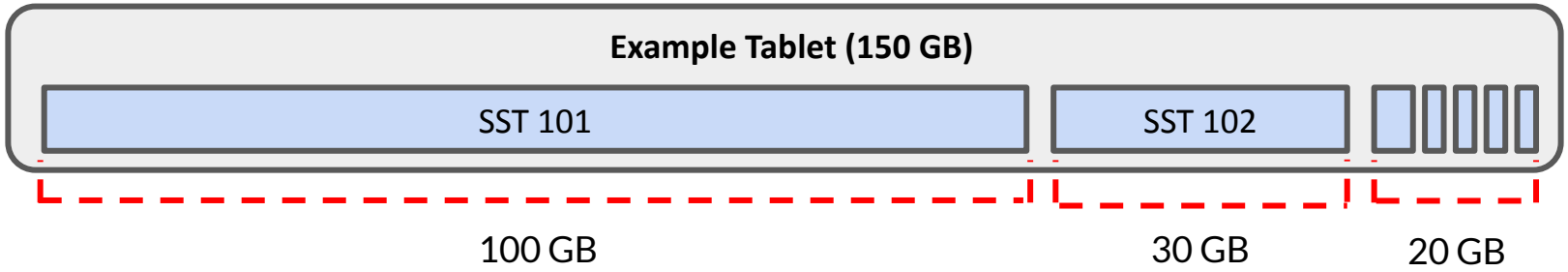
- Shared set of resources (CPU/IO) for all compactions in a TServer
- Managed by a priority thread pool
  - Max threads scale with # of CPUs
- Priorities (in general):
  - Flushes > compactions (if same pool is used)
  - Small compactions > large ones
  - Background compactions > full compactions

Background compaction TServer flags

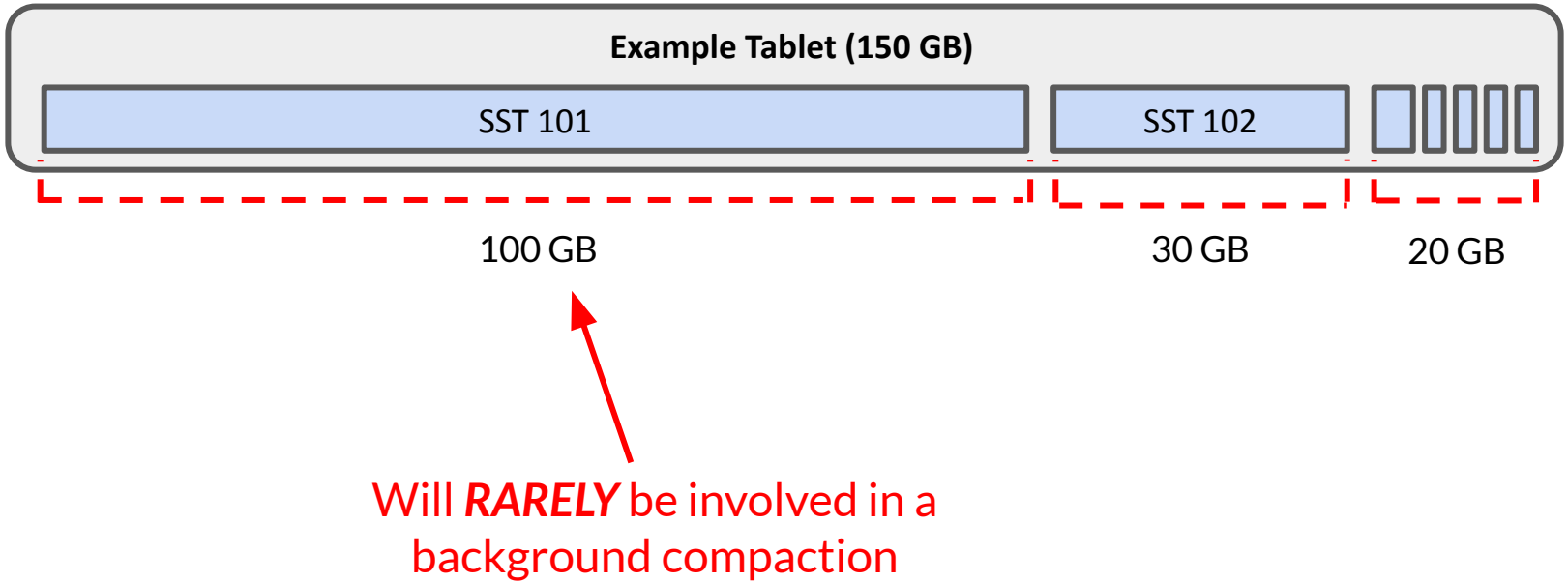
- ***rocksdb_max_background_compactions***
  - Maximum number of simultaneous running compactions allowed per TServer
  - Default: -1 / auto (scales per CPU)
- ***rocksdb_compact_flush_rate_limit_bytes_per_sec***
  - The write rate limit for flushes and compactions (per TServer by default)
  - Default: 1 GB
- ***rocksdb_compaction_size_threshold_bytes***
  - Threshold beyond which a compaction is considered “large”
  - Default: 1 GB
- ***sst_files_soft_limit / sst_files_hard_limit***
  - The soft and hard limits of the number of SSTs allowed per tablet
    (i.e. if reached, the tablet is behind on compactions!)
  - Writes will be throttled when the soft limit is reached, and stopped at the hard limit
  - Default: 24 / 48

Motivation

Example large tablet after many compactions



Example large tablet after many compactions



Times when background compactions may not be enough

Update- or Delete- Heavy Workloads

- Older data versions or deleted data remain in files for extended periods of time
- Read and space amplification

TTL (Time-To-Live) Workloads

- Older data expires over time, and remains in large files for extended periods of time

Deletion Compliance

- Deleted data must be removed from the system by a certain date

Example table

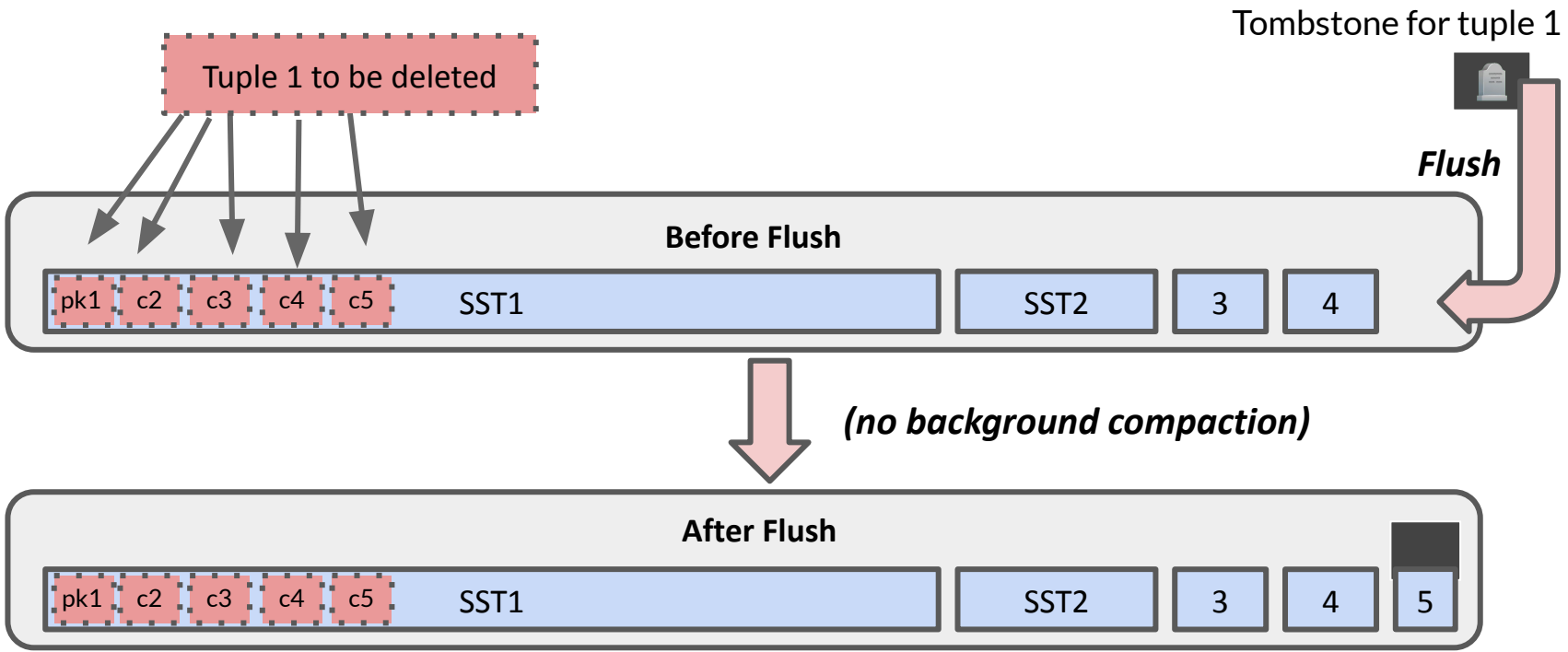
```
CREATE TABLE example (
    pk1  bigint NOT NULL,
    c2   varchar(255),
    c3   varchar(255),
    c4   varchar(255),
    c5   varchar(255),
    PRIMARY KEY (id1));
```

pk1 (PK)	c2 (varchar)	c3 (varchar)	c4 (varchar)	c5 (varchar)

```
INSERT INTO example (pk1, c2, c3, c4, c5)
  SELECT s_id, md5(random()::text), md5(random()::text),
         md5(random()::text), md5(random()::text)
   FROM generate_series(1,1000000) AS s_id;
```

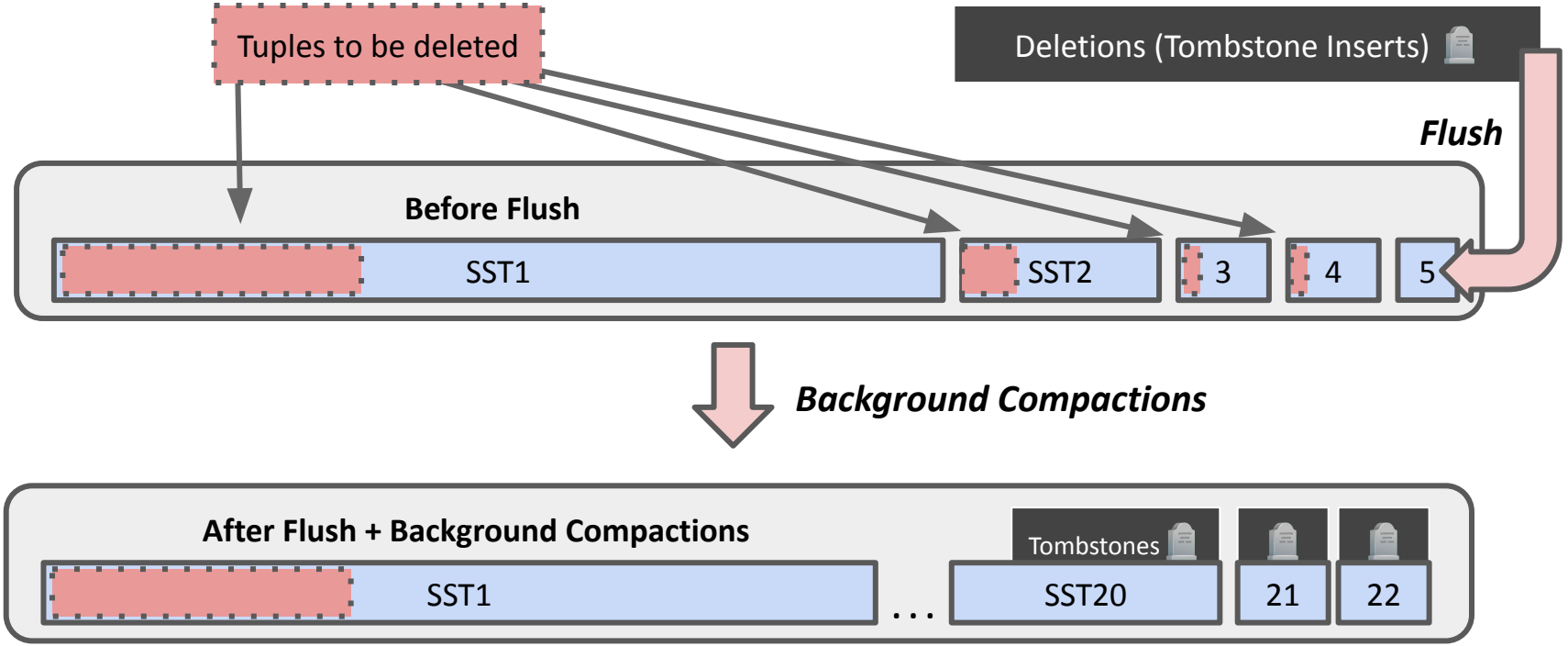
Tuple deletion

```
DELETE FROM example WHERE pk1 = 1;
```



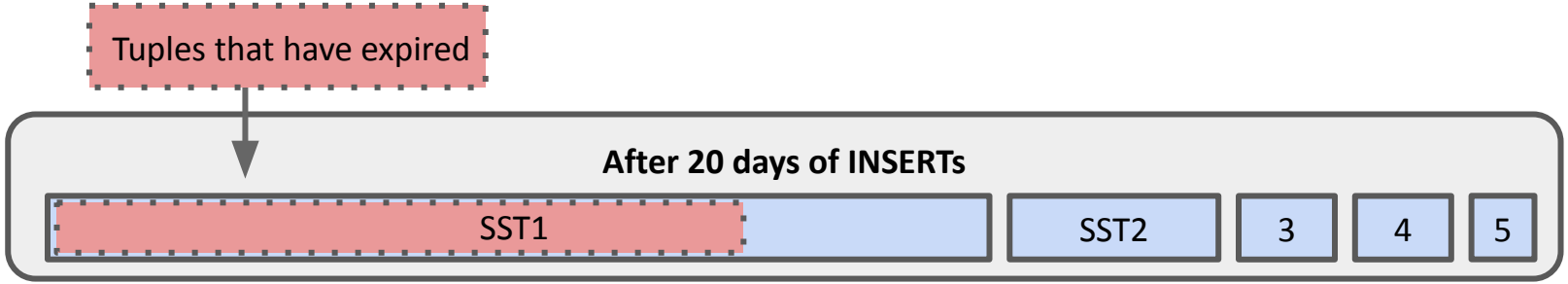
Tuple deletion

```
DELETE FROM example WHERE pk1 >= 500000 AND pk1 < 700000;
```



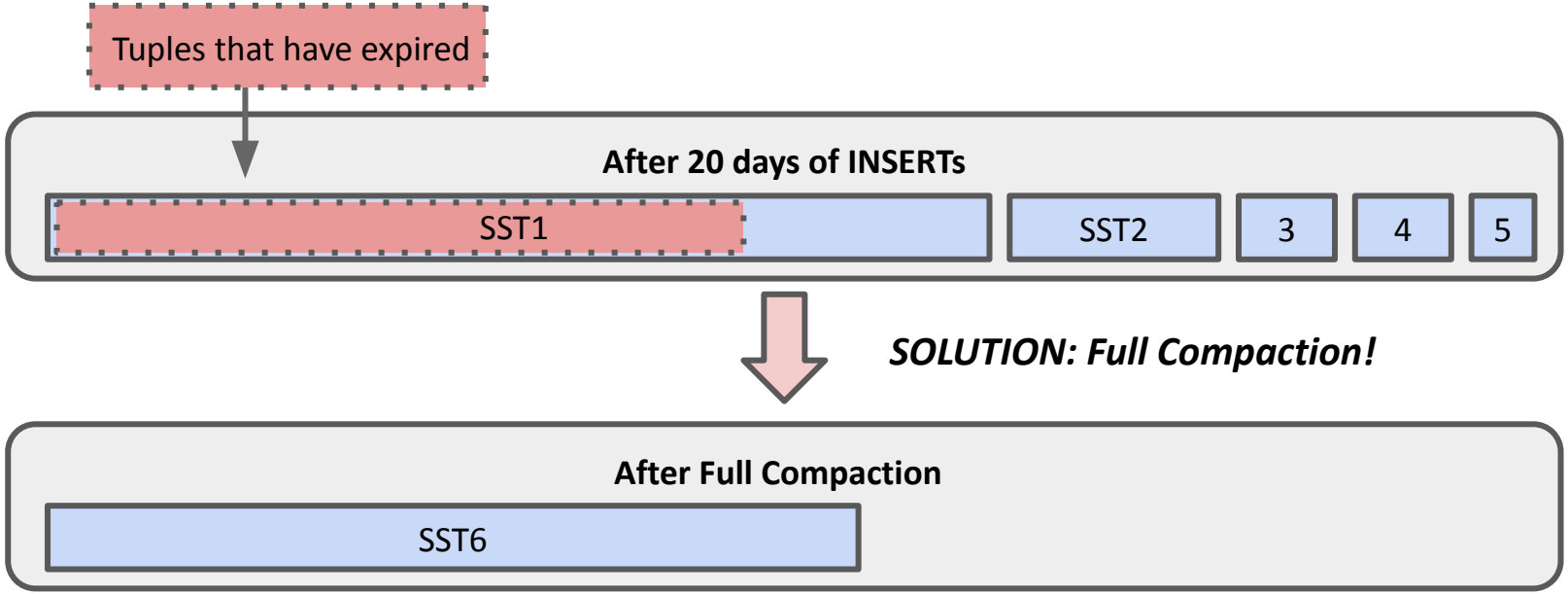
Default TTL (Time-to-Live)

```
ALTER TABLE example WITH default_time_to_live = 10*24*60*60; // 10 days
```



Default TTL (Time-to-Live)

```
ALTER TABLE example WITH default_time_to_live = 10*24*60*60; // 10 days
```



Scheduled Full Compactions

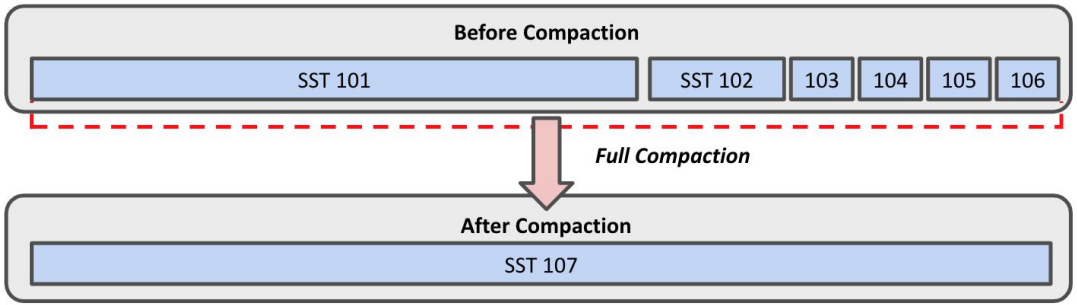
Full compactions

Compaction that includes all SSTs for a tablet

- All files will be compacted into one single file
- Iterates through all data items of all files, ignores deleted/expired data

Resources required (same as background compactions)

- CPU cycles on threads dedicated to compactions (same thresholds apply)
- Disk IO for copying to a new file
- Disk space for temporary copy of all live data in tablet (capped if tablet splitting enabled)
- **NO LOCKS REQUIRED**



How to use scheduled full compactions (TServer flags)

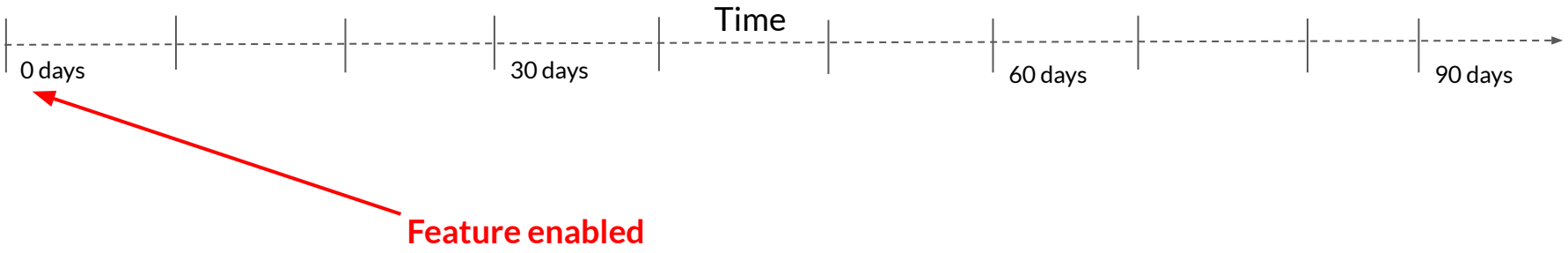
Compaction Frequency (*scheduled_full_compaction_frequency_hours*)

- Upper bound for how frequently full compaction should be run
- Default value: 0 (i.e. feature turned off)
- Recommended values:
  - Every 30 days (720 hours), or every 2 weeks (336 hours)
  - Not less than every 7 days (168 hours)

Jitter Factor (*scheduled_full_compaction_jitter_factor_percentage*)

- Determines the “jitter” introduced into the scheduling
- Reduces likelihood of many tablets scheduled simultaneously
- Default value: 33 (recommended)
  - i.e. 33% of the frequency of schedule is allotted for jitter.
  - Example: if compactions are scheduled every 30 days with 33% jitter, each tablet will be scheduled for compaction pseudorandomly every 20 to 30 days

Scheduling with Jitter Factor



1 Tablet 1 Full Compaction

2 Tablet 2 Full Compaction

3 Tablet 3 Full Compaction

Full Compaction Frequency: **30 days**

Jitter Factor Percentage: **33%**

Jitter Factor: **10 days**

Scheduling with Jitter Factor

1 Tablet 1 Full Compaction

2 Tablet 2 Full Compaction

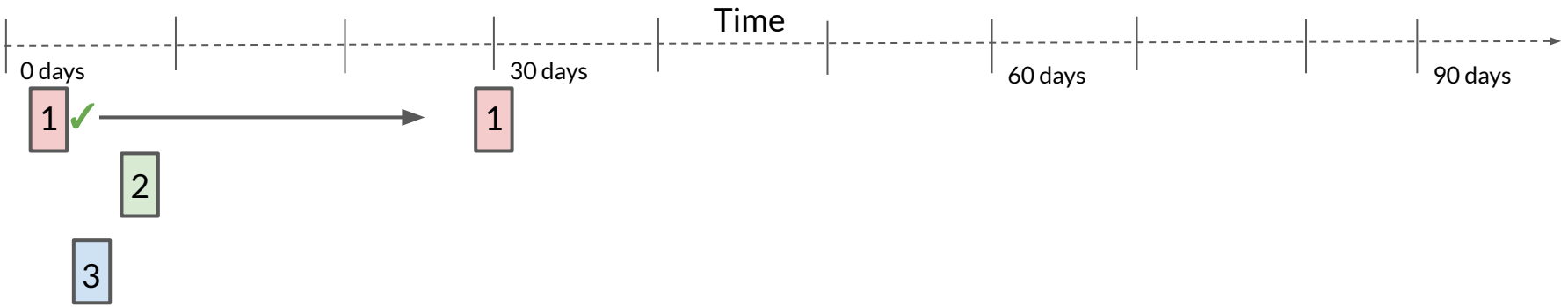
3 Tablet 3 Full Compaction

Full Compaction Frequency: **30 days**

Jitter Factor Percentage: **33%**

Jitter Factor: **10 days**

Scheduling with Jitter Factor



1 Tablet 1 Full Compaction

2 Tablet 2 Full Compaction

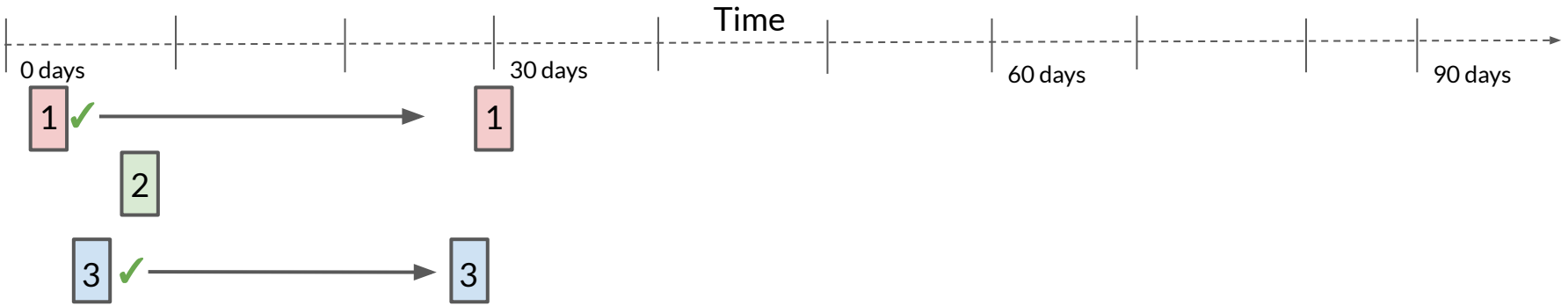
3 Tablet 3 Full Compaction

Full Compaction Frequency: **30 days**

Jitter Factor Percentage: **33%**

Jitter Factor: **10 days**

Scheduling with Jitter Factor



1 Tablet 1 Full Compaction

2 Tablet 2 Full Compaction

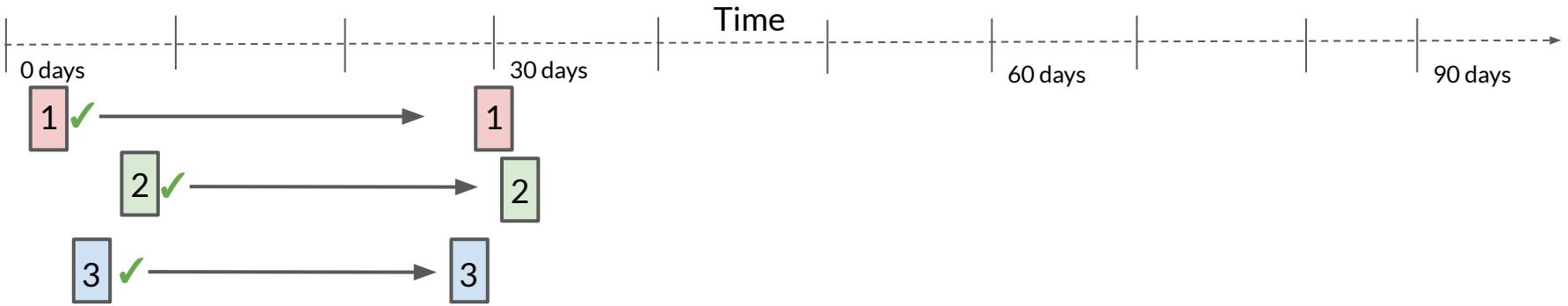
3 Tablet 3 Full Compaction

Full Compaction Frequency: **30 days**

Jitter Factor Percentage: **33%**

Jitter Factor: **10 days**

Scheduling with Jitter Factor



1 Tablet 1 Full Compaction

2 Tablet 2 Full Compaction

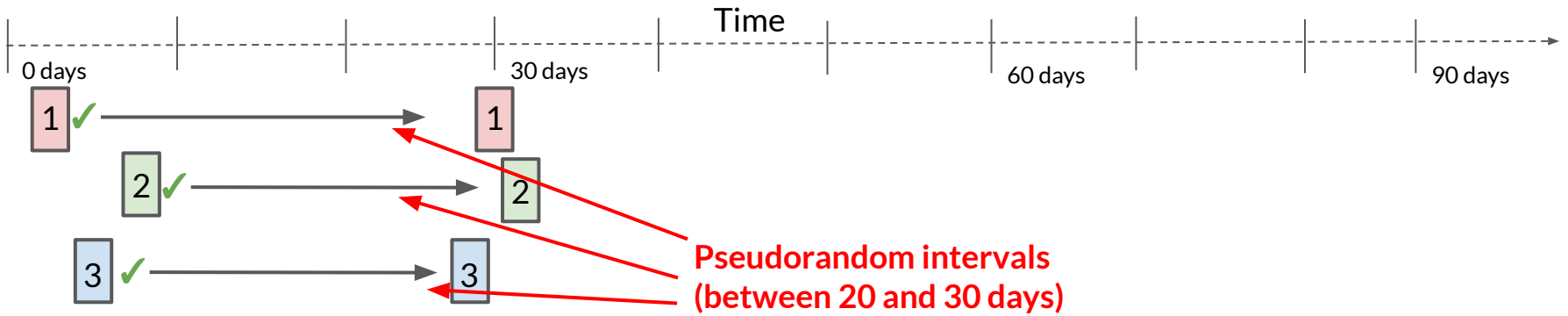
3 Tablet 3 Full Compaction

Full Compaction Frequency: **30 days**

Jitter Factor Percentage: **33%**

Jitter Factor: **10 days**

Scheduling with Jitter Factor



1 Tablet 1 Full Compaction

2 Tablet 2 Full Compaction

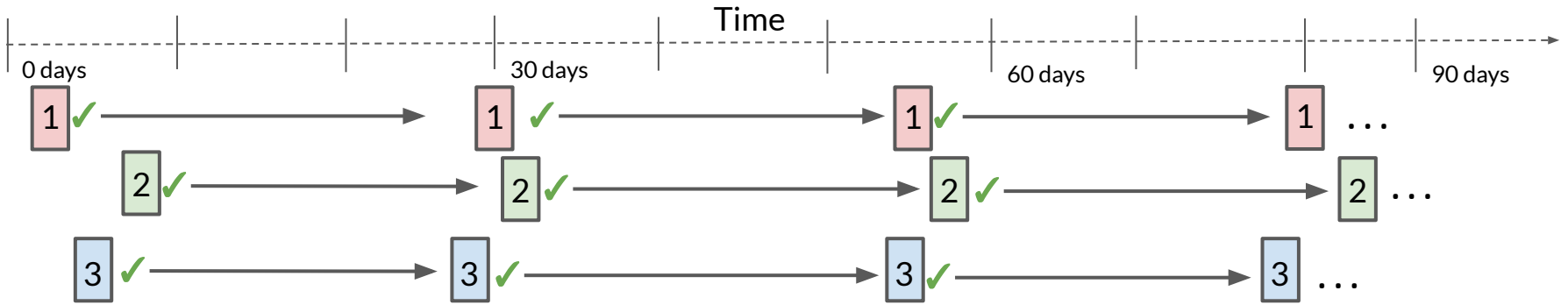
3 Tablet 3 Full Compaction

Full Compaction Frequency: **30 days**

Jitter Factor Percentage: **33%**

Jitter Factor: **10 days**

Scheduling with Jitter Factor



1 Tablet 1 Full Compaction

2 Tablet 2 Full Compaction

3 Tablet 3 Full Compaction

Full Compaction Frequency: **30 days**

Jitter Factor Percentage: **33%**

Jitter Factor: **10 days**

Demo

Future feature improvements

Auto-trigger full compactions when needed

- Automatically improve the performance of some range queries via a full compaction when needed

Specify off-peak “preferred” compaction times

- Schedule more full compactions at workload off-peak hours/days

Client-side full compaction status reporting/monitoring

- Improved visibility into full compaction status

Thank You

Join us on Slack: yugabyte.com/slack (#yftt channel)

Star us on Github: github.com/yugabyte/yugabyte-db

YugabyteDB Friday Tech Talks

YFTT

Talks by Engineers for Engineers

yugabyteDB