Don’t Let Tombstones Affect Scan Performance

Premkumar Thangamani
Feb/24/2023

YugabyteDB Friday Tech Talks

YFTT

Talks by Engineers for Engineers

yugabyteDB

Topics of discussion

- Effect of Large Range Deletes
- Point Lookups vs Range Scan
- Simple Overview
  - Log Structured Merge Tree (LSM)
  - Sorted Strings Table (SST)
- Possible Workarounds
- Future Work

Effect of Large Deletes

Events Scenario

Events / Job Queue / Ticketing

```
Events (
    id int,
    ts timestamp,
    data text,
    PRIMARY KEY (id)
);
```

Earliest event

```
SELECT * FROM events ORDER BY ts ASC LIMIT 1
```

Helpful Index

- `CREATE INDEX idx_ts on events(ts ASC)`

Earliest event - usually very fast (~2 ms)

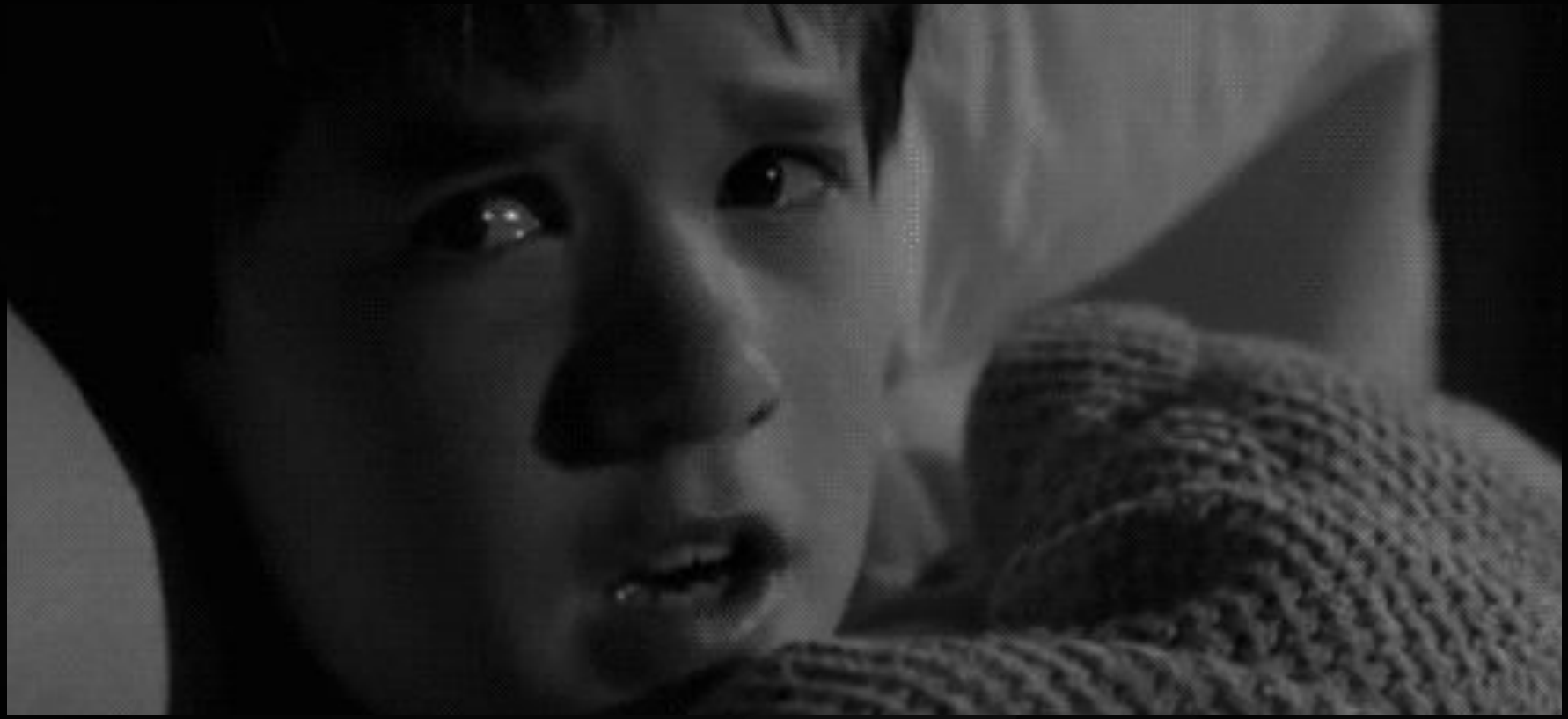
```
SELECT * FROM events ORDER BY ts ASC LIMIT 1
```

```
 id |          ts           |    data
----+-----------------------+-------------
  1 | 2021-03-29 14:25:00 | eventdata-1


Time: 2.257 ms
```

After a lot of deletes … (~2000 ms)

```
SELECT * FROM events ORDER BY ts ASC LIMIT 1
```

```
    id |          ts           |    data
-------+-----------------------+-------------
500001 | 2022-03-11 18:45:00 | eventdata-500001

Time: 2271.798 ms
```



I see dead records!

On Deletes

- Records not removed immediately
- Tombstone Markers will be placed
- **LSM**: Log Structured Merge-Tree
  - Multi-Level Tree like storage

- **MVCC**: Multi Version Concurrency Control
  - Consistent View w/o Locking for reads

Simple visual of SST

```
# our index is [ts ASC]
2021-03-29 14:25:00,1       <DEL>  <= Actual search start position
2021-03-29 14:26:00,2       <DEL>
2021-03-29 14:27:00,3       <DEL>
. . .
. . .
2022-03-11 18:44:00,500000 <DEL>

2022-03-11 18:45:00,500001         <= Preferred search start position
2022-03-11 18:46:00,500002
```

Compaction

- Fixes the problem
- Removes the Tombstones

- But,
  - Resource Intensive
  - To support MVCC, some records may stay on

Simple Trick

Last processed timestamp

- The problem: not knowing where to start.

- Let’s help the Query Executor
  - `ts > :last_processed_ts`

Use last processed timestamp as a hint

```
SELECT * FROM events
   WHERE ts > '2022-03-11 18:44:00'
   ORDER BY ts ASC LIMIT 1

  id    |          ts           |       data
--------+-----------------------+------------------
500001 | 2022-03-11 18:45:00 | eventdata-500001


Time: 2.850 ms
```

DEMO

Storing last timestamp

- Start with a base timestamp of zero
  - Store it as a local variable

- Store the last_processed_ts in a separate table
  - Background thread updates it every ~30s
  - Useful in a multiple worker scenario

Follow up

Future Work

- Identifying scenarios of dense deletes
- Automatically triggering compaction

Further Watching

- Check out John Meehan's Tech Talk on Compaction
- Full Compactions in YugabyteDB

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