Automatic Data Deletion

Aravind Nallan

Friday 27 May - 9:30am - 10am PT

YFTT YUGABYTEDB FRIDAY TECH TALKS

yugabyteDB

Time-To-Live

- Use cases for which you know that the data has a limited lifetime
  - Limited lifetime one-time passwords
  - Coupons valid for a certain time period
  - Time-series data which needs to be in an OLTP database while its relevant
- Currently only available in our Apache Cassandra inspired YCQL API

Time-To-Live - mechanics

- All data stored in YugabyteDB has a timestamp
  - At row level, column level, even at element level for collection types
  - Inserts/Updates change this timestamp
- Tables can have a default TTL
  - Gets applied to all data inserted into the table
  - Stored in the system catalog, not with data
- YCQL rows, columns and collection elements can have a TTL associated with them
  - Stored in the SST files

Time-To-Live - implementation

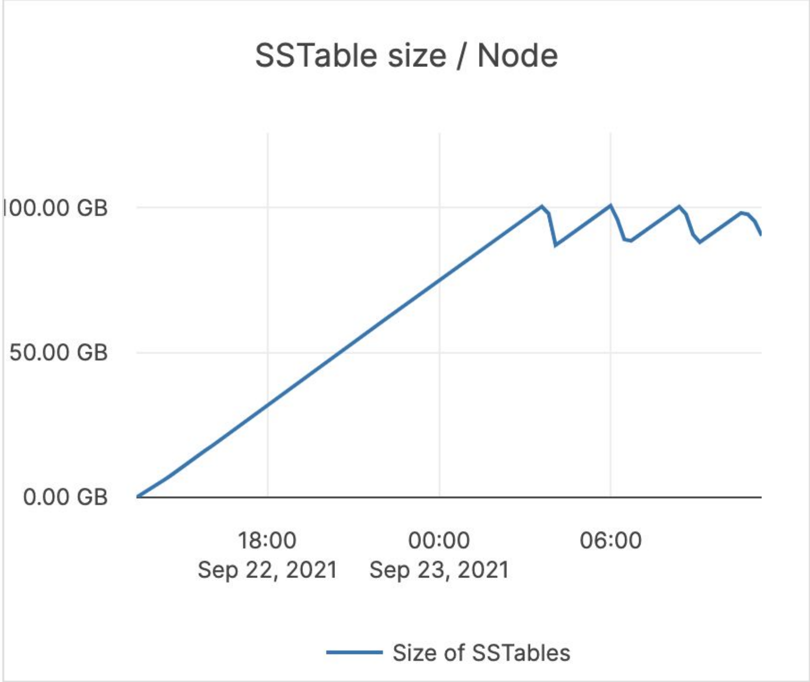
- Implemented in DocDB layer
  - Using the compaction filter feature of RocksDB
  - Compaction reclaims space
  - No need to drop tombstones for expired data like in Apache Cassandra
- TTL and indexes can not be used together
  - Expired data deleted during compaction
  - Compaction is per tablet
  - Difficult to guarantee deletion of related index data transactionally

Managing time-series data - bulk expiration

- Entire SST files can be deleted when all the data in them expires
- Similar to Apache Cassandra's TimeWindowCompactionStrategy
- We added a feature to RocksDB where SST files larger than a threshold are not compacted
  - For minor compactions only
- We added a feature called the compaction file filter similar to compaction filter feature of RocksDB

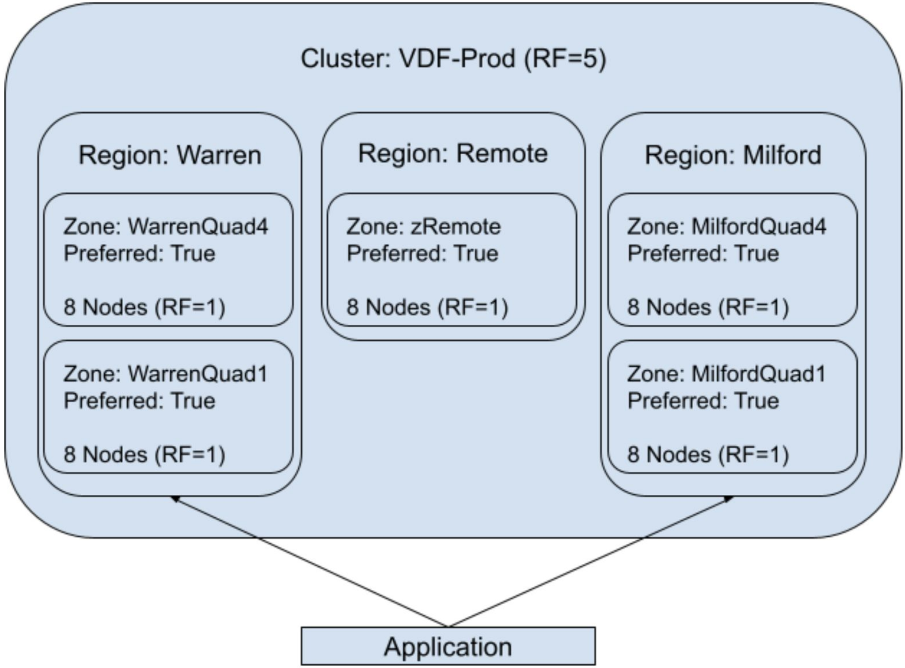
Managing time-series data - simulating TWCS

- Estimate the size of SST for 1 TWCS period
- Set tablet_enable_ttl_file_filter to true
- Set rocksdb_max_file_size_for_compaction to that value
  - Or a multiple thereof based on how much space amplification you can tolerate
- May also need to adjust upwards the following flags
  - sst_files_soft_limit
  - sst_files_hard_limit



Managing time-series data - implementation at GM

- Vehicle telemetry data collection including for OnStar
- 40 node cluster across 3 regions and 5 zones
- 360TB of data managed at RF5
- 550K writes per sec, sized for >2M writes/sec
- Each node expiring about 100GB of data using TTL based file expiry



References

- https://docs.yugabyte.com/preview/develop/learn/ttl-data-expiration-ycql/
- https://docs.yugabyte.com/preview/api/ycql/dml_insert/#insert-a-row-with-expiration-time-using-the-using-ttl-clause
- https://docs.yugabyte.com/preview/api/ycql/dml_update/#update-with-expiration-time-using-the-using-ttl-clause
- https://docs.yugabyte.com/preview/api/ycql/expr_fcall/#ttl-function

Thank You

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

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

YFTT YUGABYTEDB FRIDAY TECH TALKS

yugabyteDB