Запрос на проект по реализации отложенного обновления вторичных индексов на LSM деревьях в базах данных

Составил:

Шпилевой В.Д. 621гр.

Москва, 2018.

Актуальность

LSM (Log-Structured Merge) деревья были разработаны в 1990-х годах для задач с интенсивной записью, и использовались в файловых системах и для резервного копирования. Но их применение в СУБД (Система Управления Базой Данных) было ограничено из-за *скрытых чтений*. Скрытыми называют чтения, которые выполняются СУБД при обновлении данных, чтобы либо найти старые данные и удалить их, либо чтобы проверить ограничения уникальности при вставке новых данных. Значительная часть чтений в СУБД - скрытая, так как почти любое обновление данных требует проверки различных ограничений и удаления старых данных, и это почти ничего не стоит в B-деревьях, где для обновления данных в любом случае нужно читать. Но на LSM деревьях скрытые чтения значительно снижают производительность, поскольку лишают их преимуществ версионности данных, когда можно сохранять новые данные не читая и не удаляя старые явно.

С появлением SSD (Solid-State Drive) дисков абсолютная скорость любых чтений и записи возросла на порядки по сравнению со старыми механическими дисками, но существенно увеличился разрыв между скоростями последовательной записи и последовательного чтения. Благодаря тому, что LSM дерево всегда выполняет запись на диск последовательно, а чтения из него на SSD по скорости мало отличаются от B-дерева, LSM дерево стало одной из стандартных структур данных для СУБД.

Однако SSD хоть и делает LSM дерево более конкурентноспособным, но не решает проблему существования скрытых чтений на любое обновление при наличии вторичных индексов, что не позволяет использовать все возможности LSM деревьев, когда в таблице в БД (База Данных) больше одного индекса.

Когда в таблице только один индекс на LSM дереве, то любые изменения, не требующие знания старых данных (такие как *REPLACE*, *DELETE*),

возможны без скрытых чтений. Например, в случае *REPLACE* запись просто попадает в дерево с новой версией. Тоже самое при *DELETE* - ключ (набор индексируемых колонок и их значений), по которому производится удаление, попадает в дерево с новой версией и пометкой, что это именно удаление, а не вставка. Скрытых чтений не выполняется. Но при появлении вторичного индекса даже *REPLACE* и *DELETE* вынуждены читать старые данные из первичного индекса, чтобы узнать, какой у старой записи был вторичный ключ, и вставить его *DELETE* в LSM дерево вторичного индекса.

Это обычная процедура для индексов на B-деревьях, где нет версионности данных, и на классических LSM деревьях ее тоже нельзя избежать. Это происходит из-за того, что удаление старых версий данных в LSM дереве работает так, что записи считаются разными версиями одних и тех же данных, только если они равны по ключу, по которому сортируется дерево. И если некоторый запрос меняет этот ключ в уже существующей записи, не читая и не удаляя ее явно, то новая запись становится никак не связанной со старой, и старая не удалится никогда — LSM дерево видит их как разные ключи.

Таким образом, задача борьбы со скрытыми чтениями в таблицах с индексами на LSM деревьях становится актуальной.

Цели проекта

Необходимо разработать и реализовать алгоритм отложенного обновления вторичных индексов на LSM-деревьях. Для достижения заданной цели необходимо решить следующие подзадачи:

- исследовать существующие способы уменьшения влияния скрытых чтений и ускорения записи;
- разработать и реализовать алгоритм отложенного обновления вторичных индексов на LSM-деревьях;
- провести экспериментальную апробацию реализации.

Разработка должна вестись на языке C на основе архитектуры БД Tarantool для движка хранения Vinyl (Vinyl — реализация LSM деревьев в Tarantool).

Реализация нового алгоритма должна ускорить запись в таблицу в БД Tarantool созданную на движке хранения Vinyl минимум в два раза.

Новый алгоритм должен быть формализован и донесен до научной общественности через выступления на конференциях, публикации в сборниках, книгах, журналах.

В результате выполнения проекта должны быть получены:

- формализованное описание алгоритма;
- реализация в виде патча на языке C для Tarantool версии 1.7;
- результаты экспериментов, доказывающие увеличение скорости не менее чем в два раза, в виде графиков.

Взаимосвязи и ограничения

Проект связан с классическими LSM деревьями, B деревьями и их реализациями, с оценками сложностей, с существующими способами откладывания скрытых чтений, ускорения записи. С устройством SSD дисков, операционными системами Linux.

Сроки выполнения всех поставленных целей проекта — с 01.09.2017 до 20.04.2018.

Заинтересованные лица

Проект инициирован техническим директором Tarantool Константином А. Осиповым и программистом Tarantool — Владиславом Д. Шпилевым. Проект финансируется юр. лицами Mail.Ru Group и Tarantool. Контроль выполнения проекта ведется Константином А. Осиповым, Дмитрием Ю. Волкановым.