

АЛГОРИТМ ОТЛОЖЕННОГО ОБНОВЛЕНИЯ ВТОРИЧНЫХ ИНДЕКСОВ НА LSM-ДЕРЕВЬЯХ

Шпилевой Владислав Дмитриевич¹

Волканов Дмитрий Юрьевич²

Осипов Константин Аркадьевич³

1: *Магистр, факультет ВМК МГУ имени М. В. Ломоносова, Москва, Россия*

2: *К.ф.-м.н., ассистент, факультет ВМК МГУ имени М. В. Ломоносова,
Москва, Россия*

3: *CTO Tarantool, Mail.Ru Group*

E-mail: vshpilevoi@mail.ru, volkanov@lvk.cs.msu.su, kostja@tarantool.org

В настоящее время растет популярность Log-Structured-Merge-деревьев (LSM) [1], как замены В-деревьям, для применения в задачах с интенсивной записью. Это вызвано тем, что в SSD-дисках издержки В-деревьев, вызванные случайной записью, стали доминировать над издержками LSM-деревьев, вызванными чтениями [2].

Существенная часть чтений, которые выполняет современная СУБД - скрытые. Это чтения при выполнении запросов обновления БД для проверки консистентности и для поиска и удаления старых данных во вторичных индексах [3]. Эти чтения лишают LSM-таблицу превосходства над В-деревом в применении к задачам с интенсивной записью из-за необходимости явно искать и удалять старые записи. Скрытые чтения приводят к значительному снижению скорости выполнения запросов из-за обращений к диску.

В рамках данной работы разработан и реализован алгоритм, который позволяет не делать скрытых чтений для некоторых операций обновления таблицы на LSM-индексах. Ключевая идея алгоритма в том, что скрытое чтение и удаление старых данных на *REPLACE* и *DELETE* запросы может быть в некоторых случаях отложено до выполнения слияния уровней LSM-дерева. Можно показать, что откладывание обновлений дает более чем двухкратный прирост скорости. Пусть

N — общее количество записей в таблице,

b — фактор ветвления В⁺-дерева на нулевом уровне LSM-дерева,

r — отношение размеров соседних уровней LSM-дерева,

K — общее число индексов в таблице.

Теорема 1. *Откладывание операций *REPLACE* и *DELETE* ускоря-*

ет их работу в $[2 + \frac{O(\log_r(N-m))}{O(\log_b m) * K}]$ и $[K + 1 + \frac{O(\log_r(N-m))}{O(\log_b m)}]$ раз соответственно.

На практике скорость доступа к данным на диске существенно ниже, чем к данным в памяти, за счет чего даже на таблицах размером до миллиона записей можно получить почти десятикратное ускорение за счет того, что отложенные обновления не читают диск вовсе.

Иллюстрации

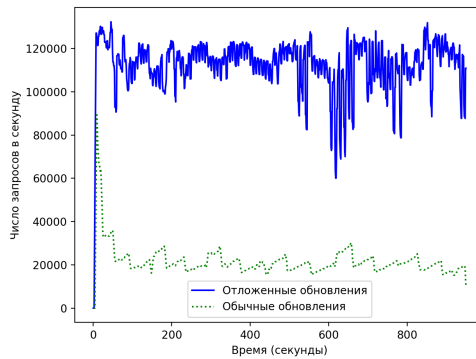


Рисунок 1. Практическая разница в скорости обычных и отложенных обновлений

В результате выполнения работы разработан и реализован алгоритм отложенного обновления индексов на LSM-деревьях, получена математическая и экспериментальная оценки его эффективности.

Литература

1. O'Neil P. et al. The log-structured merge-tree (LSM-tree) //Acta Informatica. – 1996. – Т. 33. – No. 4. – С. 351-385.
2. Agrawal N. et al. Design Tradeoffs for SSD Performance //USENIX Annual Technical Conference. – 2008. – Т. 8. – С. 57-70.
3. Ren K. et al. SlimDB: a space-efficient key-value storage engine for semi-sorted data //Proceedings of the VLDB Endowment. – 2017. – Т. 10. – №. 13. – С. 2037-2048.