

УДК 004.657

ОПТИМИЗАЦИЯ ВЫПОЛНЕНИЙ UPDATE-ОПЕРАЦИЙ НАД JSONB ДАННЫМИ В POSTGRESQL

Тарасов И. С.¹

Научный руководитель – преподаватель Николаев В. В.¹

¹Университет ИТМО

iwan.tarasow2013@gmail.com

Введение

Для работы с данными JSON и с другими слабоструктурированными данными часто используются NoSQL системы. Однако многие из таких СУБД ограниченно поддерживают ACID-транзакции либо требуют сложной настройки для обеспечения целостности данных, что делает реляционные СУБД более предсказуемыми и безопасными решениями для гибридных сценариев.

В PostgreSQL для быстрого поиска строк в таблице по JSONB (так называется тип для JSON-данных в PostgreSQL) можно использовать индексы на отдельные пути (ключи), например *name*. Однако при изменении столбца, участвующего в индексе, все индексы приходится перестраивать, что влечёт за собой дополнительную нагрузку на систему, особенно если индексов много. Для минимизации накладных расходов в PostgreSQL был добавлен механизм HOT обновлений (Heap Only Tuple), которые применяются, когда изменяются неиндексируемые столбцы. При HOT обновлении создаётся новая версия строки без перестройки индексов. Однако HOT не работает с индексами на выражения. Например, обновление ключа *age* в JSONB-столбце приводит к отключению HOT, даже если индекс существует только на ключе *name*, который не изменился.

За последние годы было предложено несколько подходов по оптимизации механизма HOT. Такие решения, как WARM (Write Ahead Reduction for Mutable indexes) [1] и PHOT (Partial Heap Only Tuple) [2], направлены на снижение количества обновлений за счёт предоставления разрешений индексам ссылаться на различные версии строк, что позволяет обновлять только те индексы, которые в действительности были затронуты. Однако данные подходы были отклонены сообществом PostgreSQL, поскольку их реализация требует глубокой переработки архитектуры PostgreSQL и модификации значительного числа компонентов.

Такие решения, как SFI (Surjective Functional Indexes) [3] и прототип Грега Бёрда [4], ориентированы на частичные индексы и индексы на выражения и не предполагали кардинальных изменений в архитектуре, как в случае с PHOT и WARM. Подход SFI не был принят разработчиками по причине того, что он базируется на доверии к пользователю: неверная аннотация индекса как «сюръективный» может привести к неконсистентности данных. Прототип Грега Бёрда на данный момент остается наиболее актуальным предложением и до сих пор развивается, но пока что так и не был принят в официальный релиз из-за больших накладных расходов – для принятия решений о HOT-обновлении требуется двухкратное вычисление значений всех индексов. Хотя в патче Грега Берда реализовано кэширование результатов вычислений, избавляя их от повторного расчета, он все равно требует полного прохода по всем индексам и десериализации JSON-документа. Именно эта вычислительная стоимость стала причиной создания нового подхода.

Основная часть

Все рассмотренные решения предполагают создание универсального механизма для увеличения производительности HOT-обновлений. В рамках работы предлагается решение, которое предназначено для работы только с JSONB данными. В отличие от подходов Берда Грега и SFI, предлагаемое решение предсказывает HOT-обновления на

основе анализа структуры запроса. Оно не вычисляет значения индексных выражений, а анализирует пути, модифицируемые функцией `jsonb_set`. Решение состоит из следующих этапов:

1. Система анализирует структуру оператора UPDATE, содержащего функции модификации JSONB, и определяет, затрагивает ли обновление пути, участвующие в функциональных индексах. Если обновление касается только неиндексируемых путей, система разрешает HOT.
2. При анализе UPDATE система делит запросы на две категории: простые, основанные на прямом извлечении значений по фиксированному пути и сложные – зависящие от всего содержимого столбца, например, оператор приведения `::text`. Оптимизация применится только к простым запросам.
3. Все пути JSON, участвующие в каких-либо индексах, хранятся в хеш-таблице. Если в операции UPDATE изменяется какой-либо ключ JSON, то сначала выполняется проверка на предмет того, входит ли этот ключ в индекс. Если да, то HOT обновление запрещается.
4. Отключение оптимизации через GUC-параметр.

Выводы

В ходе работы проанализирована проблема высокой стоимости операций обновления данных в PostgreSQL, связанная с необходимостью перестройки всех индексов таблицы при изменении хотя бы одного индексируемого столбца. Мною были рассмотрены предложения по снижению накладных расходов, включая механизмы PHOT, WARM и surjective functional indexes, а также причины, по которым данные решения не были внедрены в основную ветку PostgreSQL.

В рамках исследования предложен альтернативный подход, направленный на снижение нагрузки при обновлении строк с индексами на выражения, при сохранении корректности MVCC. Предлагаемое решение ориентировано на уменьшение числа операций обновления индексных структур и может быть использовано в высоконагруженных OLTP-сценариях с частыми UPDATE-операциями и большим количеством индексов.

Литература

1. Deolasee P. Heap WARM Tuples - Design Draft / Pavan Deolasee. — Текст : электронный // PostgreSQL : [сайт]. — URL: https://www.postgresql.org/message-id/CABOikdMop5Rb_RnS2xFdAXMZGSqcJ-P-BY2ruMd%2BbuUkJ4iDPw%40mail.gmail.com (дата обращения: 16.02.2026).
2. Bossart, N. Partial heap only tuples / N. Bossart. — Текст : электронный // PostgreSQL : [сайт]. — URL: <https://www.postgresql.org/message-id/2ECBBCA0-4D8D-4841-8872-4A5BBDC063D2%40amazon.com> (дата обращения: 16.02.2026).
3. Knizhnik, K Surjective functional indexes / Konstantin Knizhnik. — Текст : электронный // PostgreSQL : [сайт]. — URL: <https://www.postgresql.org/message-id/4d9928ee-a9e6-15f9-9c82-5981f13ffca6%40postgrespro.ru> (дата обращения: 16.02.2026).
4. Burd, G. Expanding HOT updates for expression and partial indexes / G. Burd. — Текст : электронный // PostgreSQL : [сайт]. — URL: <https://www.postgresql.org/message-id/78574B24-BE0A-42C5-8075-3FA9FA63B8FC%40amazon.com> (дата обращения: 16.02.2026).