17.05.2017

Почему использовать Postgres (обновлено за последние 5 лет)

Типы данных, включая JSONB и диапазонные типы

Postgres всегда имел открытое и дружественное отношение к добавлению новых типов данных. Уже давно в нем присутствуют массивы, геопространственные данные и другие. Несколько лет назад были добавлены два типа данных, на которые стоит обратить внимание:

JSONB

JSONB - это двоичное представление JSON. Он может быть проиндексирован с помощью индексов GIN и GIST. Вы также можете выполнять запросы в полном JSON-документе для быстрого поиска данных.

Диапазонные типы

Хотя они не получили такой популярности, как JSONB, диапазонные типы могут быть особенно полезны, если они подходят для вашей задачи. В пределах одной колонки вы можете иметь диапазон значений - это особенно удобно для работы с временными промежутками. Если вы создаете приложение для работы с календарем или часто используете интервалы времени, то диапазонные типы позволяют хранить это в одной колонке. Реальное преимущество состоит в том, что затем вы можете установить ограничения на то, что определенные временные метки не могут пересекаться, или другие ограничения, которые могут иметь смысл для вашего приложения.

Расширения

Очень сложно говорить о Postgres, не упоминая всю экосистему вокруг него. Расширения становятся все более важными для сообщества и развития Postgres. Расширения позволяют вам легко подключаться к Postgres, не требуя их включения в основной код Postgres. Это означает, что они могут добавлять расширенную функциональность, не завися от выпусков и обзорного цикла Postgres. Некоторые отличные примеры:

Citus

Citus (компания, в которой я работаю) превращает Postgres в распределенную базу данных, позволяя легко распределить вашу базу данных по нескольким узлам. Для вашего приложения она все равно выглядит как одна база данных, но под капотом она распределена по нескольким физическим машинам и экземплярам Postgres.

HyperLogLog

Это моя личная любимица, которая позволяет легко получать приближенные уникальные счетчики, предварительно агрегированные, а также выполнять различные операции с ними на протяжении нескольких дней, таких как объединения, пересечения и другие. HyperLogLog и другие алгоритмы набросков могут быть чрезвычайно распространены в больших наборах данных и распределенных системах, и особенно здорово, что они почти готовы к использованию в Postgres "из коробки".

PostGIS

PostGIS не новый, но его стоит еще раз подчеркнуть. Он обычно считается самой продвинутой геопространственной базой данных. PostGIS добавляет новые расширенные геопространственные типы данных, операторы и упрощает выполнение многих операций на основе местоположения, которые вам понадобятся при работе с картографией или маршрутизацией.

Логическая репликация

В течение многих лет самой большой проблемой Postgres была сложность настройки репликации. Изначально это касалось любой формы репликации, но затем появилась стриминговая репликация (это стриминг бинарного журнала WAL или формата журнала записи). Инструменты, такие как wal-e, помогают

использовать механизмы Postgres для таких вещей, как восстановление после сбоя.

Затем мы получили основу для логической репликации в последних версиях, хотя она все равно требовала расширения для Postgres, поэтому она не была 100% готова к использованию из коробки. И, наконец, у нас появилась полноценная логическая репликация. Логическая репликация позволяет отправлять фактические команды, что означает, что вы можете реплицировать только определенные команды или определенные таблицы.

Масштабирование

В дополнение ко всем функциональным возможностям, которые мы видели, Postgres становится все лучше и лучше в плане производительности. В частности, у нас уже есть основы для параллелизма, и на некоторых запросах вы увидите намного лучшую производительность. Если вам нужно еще большее масштабирование, чем у одиночного узла Postgres (например, 122 или 244 ГБ оперативной памяти на RDS или Heroku), у вас есть варианты, такие как упоминавшаяся ранее Citus, которая поможет вам масштабироваться в ширину.

Более богатая индексация

У Postgres уже были некоторые мощные индексы, такие как GIN и GiST, которые теперь стали полезны для JSONB. Но мы также видели появление индексов KNN и Sp-GiST, и у нас еще больше новинок в этой области.

Upsert

Upsert был работой в процессе в течение нескольких лет. Это была одна из тех функций, с которыми большинство людей имели дело с помощью CTE (общих таблиц выражений), но это могло создавать гонки. Это была также одна из нескольких функций, которую MySQL имела и которой не было в Postgres. И немного больше года назад мы получили официальную поддержку upsert.

Внешние обертки данных (Foreign Data Wrappers)

Да, внешние обертки данных существовали уже много лет назад. Если вы не знакомы с внешними обертками данных (foreign data wrappers), они позволяют сопоставить внешнюю систему данных с таблицами непосредственно в Postgres. Это означает, что, например, вы можете взаимодействовать и запрашивать вашу базу данных Redis непосредственно из Postgres с помощью SQL. Они продолжают совершенствоваться все больше и больше по сравнению с тем, что у нас было более 5 лет назад. В частности, у нас появилась поддержка записываемых внешних оберток данных, что означает, что вы можете записывать данные в другие системы непосредственно из Postgres. Теперь также есть официальная внешняя обертка данных для Postgres, которая идет в комплекте с Postgres и сама по себе довольно полезна при запросах на различных экземплярах Postgres.

Еще больше

И если вы пропустили более ранние версии, пожалуйста, ознакомьтесь с ними. Краткое изложение их содержания включает:

Оконные функции
Функции
Пользовательские языки программирования (PLV8, к примеру?)
Типы данных NoSQL
Пользовательские функции
Общие таблицы-выражения
Параллельное создание индексов
Транзакционное DDL
Внешние обертки данных
Условные и функциональные индексы
Слушание и уведомления
Наследование таблиц
Транзакционная синхронная репликация