

Руководство пользователя: pg_pathman расширение

Центр разработки PostgreSQL

Exported on 05/29/2020

Table of Contents

1 Руководство для сотрудника сопровождения.....	5
2 Руководство для администратора безопасности.....	6
3 Руководство для разработчика прикладных сервисов.....	7

В состав PostgreSQL Sber Edition входит pg_pathman расширение, выполняющее секционирование больших и распределённых баз данных.

Секционированием данных называется разбиение одной большой логической таблицы на несколько меньших физических секций. Секционирование может принести следующую пользу:

- В определённых ситуациях оно кардинально увеличивает быстродействие, особенно когда большой процент часто запрашиваемых строк таблицы относится к одной или лишь нескольким секциям. Секционирование может сыграть роль ведущих столбцов в индексах, что позволит уменьшить размер индекса и увеличит вероятность нахождения наиболее востребованных частей индексов в памяти.
- Когда в выборке или изменении данных задействована большая часть одной секции, последовательное сканирование этой секции может выполняться гораздо быстрее, чем случайный доступ по индексу к данным, разбросанным по всей таблице.
- Массовую загрузку и удаление данных можно осуществлять, добавляя и удаляя секции, если это было предусмотрено при проектировании секционированных таблиц. Операция ALTER TABLE DETACH PARTITION или удаление отдельной секции с помощью команды DROP TABLE выполняются гораздо быстрее, чем массовая обработка. Эти команды также полностью исключают накладные расходы, связанные с выполнением VACUUM после DELETE.
- Редко используемые данные можно перенести на более дешёвые и медленные носители.

pg_pathman автоматически создает все секции, не использует триггеры и по умолчанию, при добавлении записи, выходящей за рамки текущей сессии, pg_pathman автоматически создаст новую секцию.

pg_pathman предоставляет ряд оптимизаций, улучшающих производительность:

1. Вставка новой записи эффективнее в отсутствии пользовательских триггеров;
2. Список секций кешируется, и поиск по ним осуществляется за $\log(N)$ (актуально при большом количестве секций);
3. Из плана запроса исключается сканирование родительской таблицы.

pg_pathman можно использовать чтобы:

- Секционировать большие базы данных, не прерывая их работу.
- Ускорять выполнение запросов с секционированными таблицами.
- Управлять существующими и добавлять новые секции на лету.
- Добавлять в качестве секций сторонние таблицы.
- Соединять секционированные таблицы для операций чтения и записи.

Установка расширения

1. Необходимо добавить pg_pathman в настроечный параметр shared_preload_libraries в конфигурационном файле PostgreSQL.conf.

❗ Важно: pg_pathman может конфликтовать с другими расширениями, которые используют те же функции для перехвата управления. В связи с этим необходимо всегда добавлять это расширение **в конец списка** shared_preload_libraries.

2. Создать расширение в БД:
CREATE EXTENSION pg_pathman;

Преимущества по отношению к секционированию с помощью наследования

То что выполняет pg_pathman можно реализовать путем создания дочерних таблиц вручную.

Последовательность действий следующая:

1. Создать родительскую таблицу:

- a. `CREATE TABLE measurements (log_date date, data text);`
2. Создать набор дочерних таблиц (секций):
 - a. `CREATE TABLE measurement_y2020m02 (CHECK (logdate >= DATE '2020-02-01' AND logdate < DATE '2020-03-01')) INHERITS (measurement);`
 - b. `CREATE TABLE measurement_y2020m03 (CHECK (logdate >= DATE '2020-03-01' AND logdate < DATE '2020-04-01')) INHERITS (measurement);`
 - c. ...
3. Для того, что бы перенаправлять INSERT в нужную секцию, необходимо на таблицу measurements навесить триггер:
 - a. `CREATE OR REPLACE FUNCTION measurement_insert_trigger()
 RETURNS TRIGGER AS $$
 BEGIN
 IF (NEW.logdate >= DATE '2006-02-01' AND
 NEW.logdate < DATE '2006-03-01') THEN
 INSERT INTO measurement_y2006m02 VALUES (NEW.*);
 ELSIF (NEW.logdate >= DATE '2006-03-01' AND
 NEW.logdate < DATE '2006-04-01') THEN
 INSERT INTO measurement_y2006m03 VALUES (NEW.*);
 ...
 ELSIF (NEW.logdate >= DATE '2020-01-01' AND
 NEW.logdate < DATE '2020-02-01') THEN
 INSERT INTO measurement_y2008m01 VALUES (NEW.*);
 ELSE
 RAISE EXCEPTION 'Date out of range. Fix the
 measurement_insert_trigger() function!';
 END IF;
 RETURN NULL;
 END;
 $$
 LANGUAGE plpgsql;`

pg_pathman автоматизирует и оптимизирует этот процесс. Например так выглядит все вышеперечисленное в pg_pathman:

```
SELECT create_range_partitions('measurements', 'logdate', '2006-02-01'::date, '1
month'::interval);
```

1 Руководство для сотрудника сопровождения

Реализованная функциональность не содержит специальных инструкций для сотрудника сопровождения.

2 Руководство для администратора безопасности

Реализованная функциональность не содержит специальных инструкций для администратора безопасности.

3 Руководство для разработчика прикладных сервисов

Реализованная функциональность не содержит специальных инструкций для разработчика прикладных сервисов.