

PGCONF.RUSSIA 2024

pgpro_redefinition – расширение
для онлайн-манипуляций с
большими таблицами

Попов Александр

a.popov@postgrespro.ru



План доклада

Предпосылки создания

Возможности pgpro_redefinition

Основная идея!

Зависимости

Сравнение с другими решениями

Реализация

Демо

Вопросы

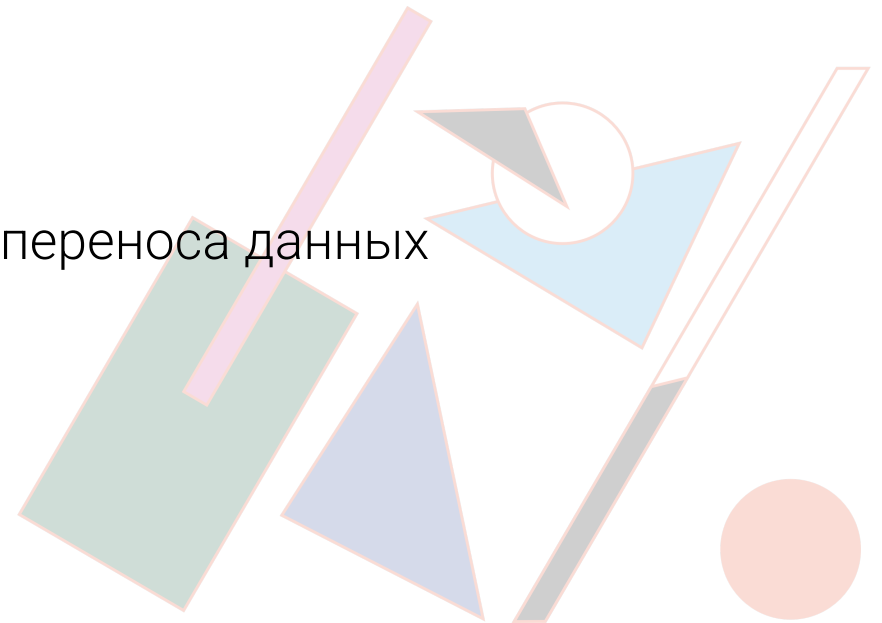
Зачем

- Большие базы
- Большие таблицы
 - Долгий vacuum
 - Разрастание таблиц
- Частая проблема функционирования большой таблицы online
- Возможность перенесения таблиц на другую СУБД, например
 - Postgres Pro Enterprise -> PostgreSQL
 - Postgres Pro Enterprise -> Postgres Pro Shardman
 - Postgres Pro Enterprise -> citus



Возможности pgpro_redefinition

- Перестроение большой таблицы в секционированную таблицу
- Разбиение таблицы на несколько таблиц
- Любые преобразования данных в процессе перестройки
- Обновление колонки
- Логическая репликация таблиц в стороннюю БД
- Callback функция запускается на каждую строку во время переноса данных
- Копирование (изменение) данных маленькими порциями
- Простой API



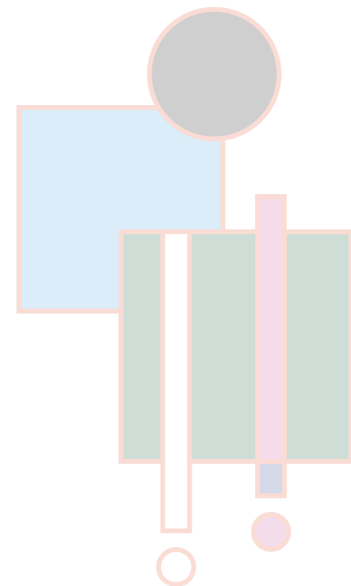
Текущая реализация

- Для захвата изменений используются триггера
- Написано на SQL, PL/PGSQL
- Использует pgpro_scheduler
- Использует автономные транзакции



Режимы работы

- **Онлайн-перенос данных**
 - Данные попадают в таблицу приемник немедленно
- **Отложенный перенос данных**
 - Данные попадают в промежуточную таблицу и переносятся после



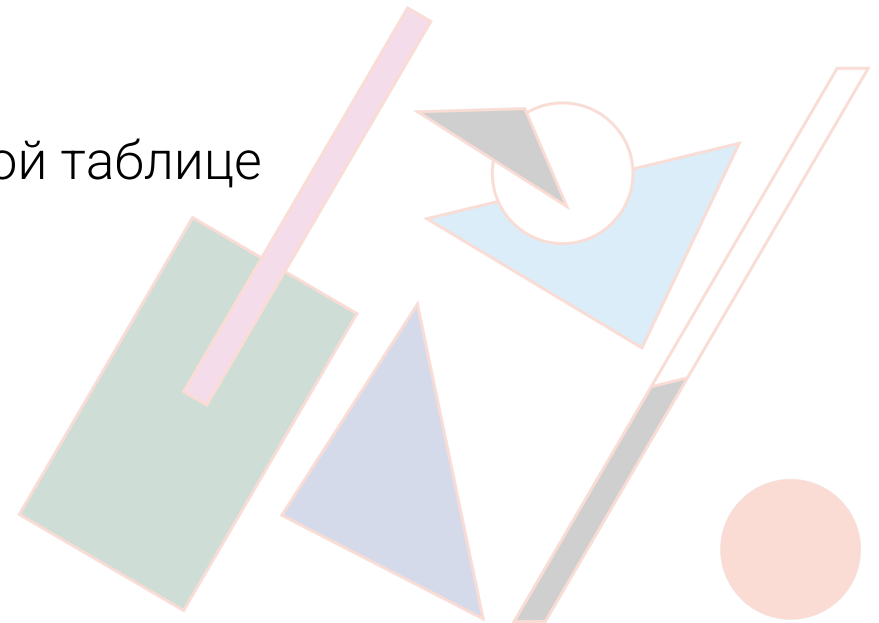
Онлайн-преобразования данных (online)

- Применение
 - Легковесные вычисления
- Ограничения
 - В случае ошибки в callback функции клиент получит ошибку, и транзакция будет прервана

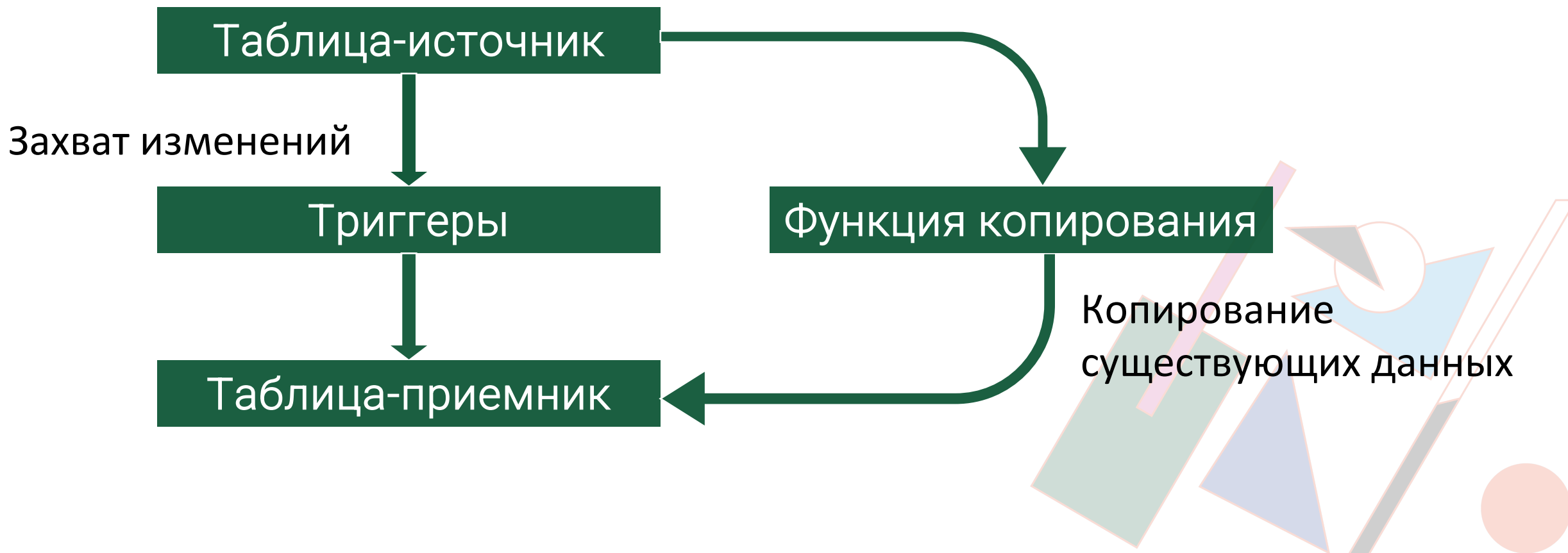


Отложенные преобразования (deferred-mlog)

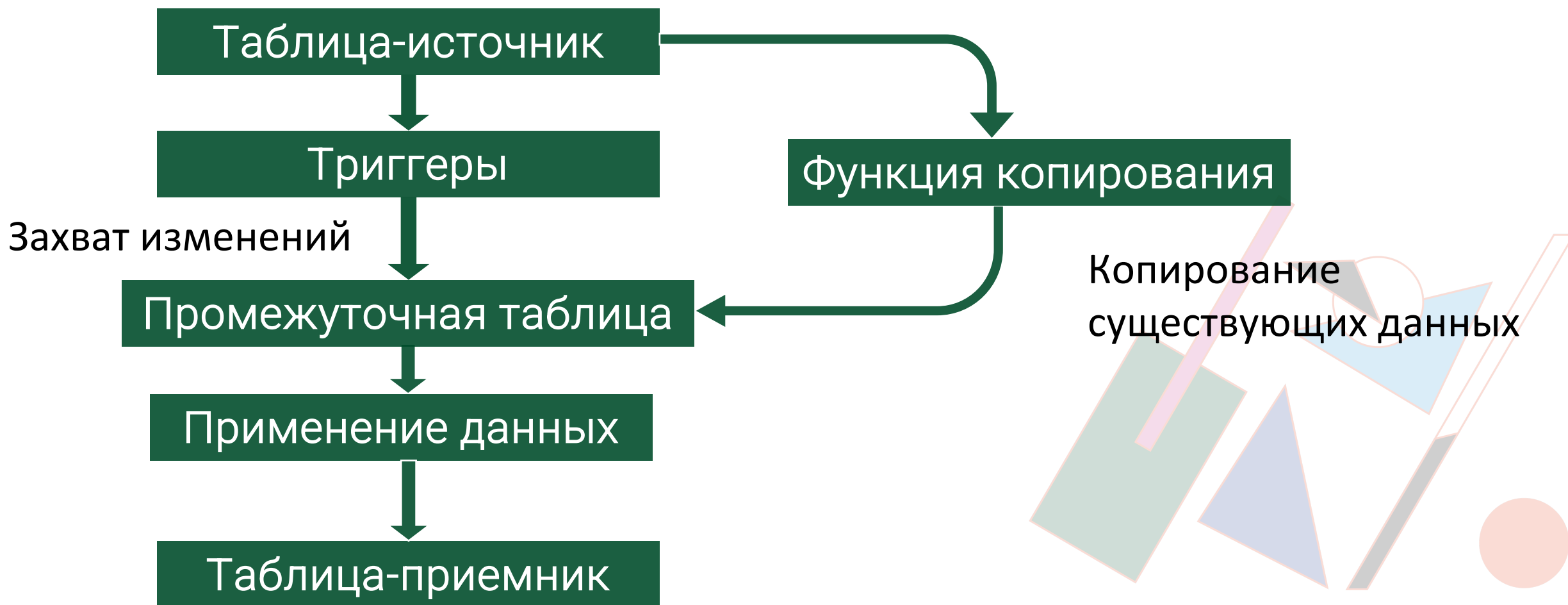
- Применение
 - Тяжелые вычисления
 - Возможность переносить данные в другие СУБД – логическая репликация
 - Устойчивость к сбоям
- Ограничения
 - Перед переносом данные сохраняются в промежуточной таблице



Онлайн-преобразования данных (online) – схематично

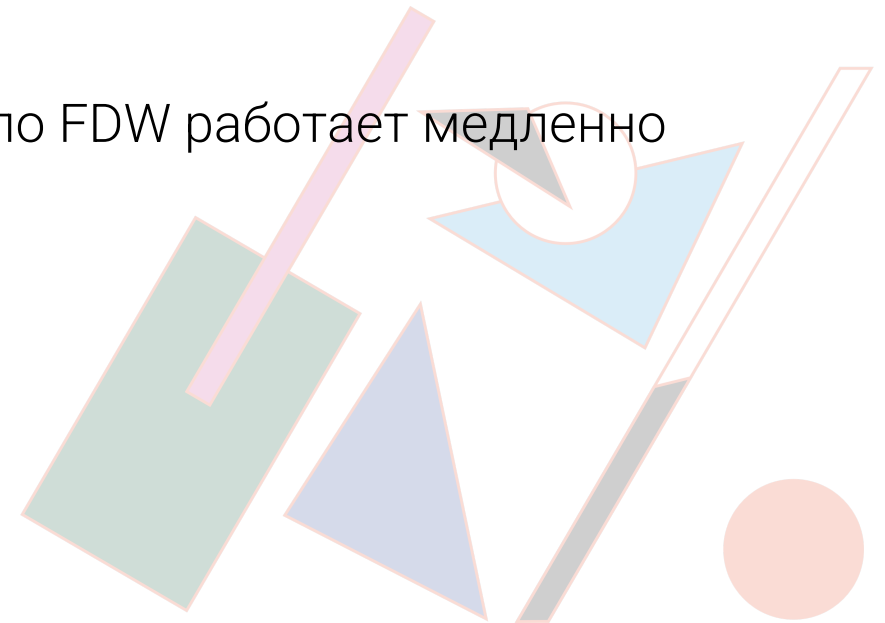


Отложенное преобразование данных (deferred) – схематично



Ограничения и недостатки

- Повышенная нагрузка на источник
- Применение данных осуществляется в один поток
- При отложенном преобразовании данных дополнительно данные пишутся во вспомогательную таблицу
- Внимание: применение изменений на стороннем сервере по FDW работает медленно
- Необходимо следить за работой pgpro_scheduler



Сравнение с другими решениями

- Простой запрос
- Свои скрипты и запросы
- pg_repack



Простой запрос

- Плюсы
 - Простота написания
 - Работает в одной транзакции
- Минусы
 - Работает в одной транзакции
 - Блокировки на таблицы
 - Применимо только к таблицам маленького и среднего размера
 - В случае прерывания запроса – запрос выполняется заново



Свои скрипты и запросы

- Плюсы
 - Полный контроль над данными
 - Применимо для больших таблиц
- Минусы
 - Сложность написания
 - Необходимость создавать и поддерживать триггера на таблицы
 - Требуется внешнее средство для копирования данных

pg_repack

- Плюсы
 - Прекрасный протестированный инструмент
 - Основная задача – избавление от распухания таблиц
 - Работает быстро
- Минусы
 - Не работает с секционированными таблицами
 - Начальное заполнение данных одним запросом
 - В случае прерывания работы – запуск pg_repack по новой
 - Вопросы при создании индексов



Сравнение с другими решениями

	Сложность реализации	Online-работа	Скорость работы	Перестроение таблицы
Простой запрос	Легко	Нет	Быстро	Да
Свои скрипты и запросы	Сложно	Да	Средне	Да
pg_repack	Легко	Да	Быстро	Нет
pgpro_redefinition	Средне	Да	Средне	Да

API

- `register_table`
 - Регистрация таблицы
- `start_capture_data`
 - Начало захвата изменений
- `start_redef_table (pause_redef_table)`
 - Начало копирования существующих данных
- `start_apply_mlog (pause_apply_mlog)`
 - Начало применения данных
- `finish_table`
 - Окончание работы
- `abort_table`
 - Отмена всех изменений



Демо – партиционирования простой таблицы

```
create table table_online_redef (  
    id    bigint primary key  
    , type varchar(1)  
    , data text  
);
```

Демо – создание промежуточной таблицы

```
create table interim_table_online_redef (  
    id    bigint primary key  
    , type varchar(1)  
    , data text  
    ) partition by hash (id);
```

```
create table interim_table_online_redef_part01  
partition of interim_table_online_redef  
for values with (modulus 2, remainder 0);
```

```
create table interim_table_online_redef_part02  
    partition of interim_table_online_redef  
for values with (modulus 2, remainder 1);
```

Демо – регистрация таблицы

```
call pgpro_redefinition._start_service_job();
```

```
call pgpro_redefinition.register_table(  
    configuration_name    => 'config_table_online_redef'  
, type                  => pgpro_redefinition._type_online()  
, kind                  => pgpro_redefinition._kind_redef()  
, source_table_name     => 'table_online_redef'  
, source_schema_name    => 'public'  
, dest_table_name       => 'interim_table_online_redef'  
, dest_schema_name      => 'public'  
);
```

Демо – callback функция

```
create function _redef_callback_6d3d427e8269679a5889c164e6027be1(  
    source_old          table_online_redef  
,  
    source_new          table_online_redef  
,  
    OUT dest_old        interim_table_online_redef  
,  
    OUT dest_new        interim_table_online_redef  
) returns record language plpgsql  
as  
$$  
declare  
begin  
    dest_old.id := source_old.id;  
    dest_old.type := source_old.type;  
    dest_old.data := source_old.data;  
  
    dest_new.id := source_new.id;  
    dest_new.type := source_new.type;  
    dest_new.data := source_new.data;  
end;  
$$;
```

Демо – начало захвата изменений

```
call pgpro_redefinition.start_capture_data(  
    configuration_name => 'config_table_online_redef'  
);
```

```
insert into table_online_redef  
select id, id%10, md5(id::text)::text from generate_series (100001,100010) as id;
```

```
select * from interim_table_online_redef;
```

id	type	data
100001	1	e2a6a1ace352668000aed191a817d143
...		
100005	5	efd1a2f9b0b5f14b1fac70a7f8e8a9e7
100009	9	795202367b2120e77b231d4d2b98e2b9

(10 строк)

Демо – начало копирования данных

```
call pgpro_redefinition.start_redef_table(  
    configuration_name => 'config_table_online_redef'  
);
```

```
select *  
from pgpro_redefinition.inc_stat  
where configuration_name = 'config_table_online_redef'  
    and job_type = pgpro_redefinition._job_type_redef_data()  
order by ts_finish desc limit 7;
```

configuration_name	job_type	dest_selected	dest_inserted	ts_start
config_table_online_redef	redef_data	1000	999	2024-03-13 16:05:10.034155
config_table_online_redef	redef_data	1000	1000	2024-03-13 16:05:09.832635
config_table_online_redef	redef_data	1000	1000	2024-03-13 16:05:09.630828

Демо – окончание работы

```
call pgpro_redefinition.finish_table(  
    configuration_name => 'config_table_online_redef'  
);  
  
alter table table_online_redef rename to table_online_redef_tmp;  
  
alter table interim_table_online_redef rename to table_online_redef;
```



Результаты тестов

CPU - 12 x "Intel(R) Xeon(R) Gold 6338 CPU @ 2.00GHz"
Memory - 64300 MB
DISK - 300GB

postgres=# \d+ source

Column	Type
id	bigint
a1	character(100)
a2	character(100)
a3	character(100)
a11	timestamp without time zone
a12	json
a13	bigint
a4	character(100)
...	
a10	character(100)

Indexes:

"source_pkey" PRIMARY KEY, btree (id)



Результаты тестов

	Широкая таблица 100ГБ	Узкая таблица (4 поля bigint) 6.7ГБ	Широкая таблица 100ГБ
vacuum full	21 минута	4 мин	
pg_repack	28 минут		
pg_repack – с нагрузкой	Не прошел		
pgpro_redefinition	С посторонней нагрузкой 78 минут	С посторонней нагрузкой 100 минут	60 минут

Ближайшие планы

- Оформить как расширение
- Регулировать нагрузку в зависимости от времени
- Многопоточное применение данных



Ссылки

- pgpro_redefinition – https://github.com/postgrespro/pgpro_redefinition
- https://docs.oracle.com/en/database/oracle/oracle-database/19/arpls/DBMS_REDEFINITION.html#GUID-2BA796C4-8B4D-49B4-8A35-4C6F789CD374
- <https://postgrespro.ru/docs/enterprise/16/pgpro-scheduler>
- <https://postgrespro.ru/docs/enterprise/16/atx>



Вопросы?



Спасибо за внимание

- Попов Александр
- a.popov@postgrespro.ru

