Программа логической репликации данных из СУБД Postgres в СУБД Postgres

Pg2PgSync

Руководство пользователя

Оглавление

Термины и определения

БД - База данных.

СУБД - Система управления базами данных.

Обратная инкрементальная миграция - автоматическая синхронизация изменений данных из новой системы на СУБД PostgreSQL в старую систему на СУБД PostgreSQL, например, если в СУБД Postgres выполнена DML-команда (добавление/ изменение/ удаление записи в таблице), то аналогичная команда и действие автоматически выполняется в соответствующей таблице СУБД Postgres.

DDL- операции/команды - (Data Definition Language), SQL-команды для описания и создания структуры БД.

DML-операции/команды – (Data Manipulation Language), SQL-команды для работы с данными в БД: добавление, изменение, удаление записей данных в таблице.

Файлы WAL – (Write-Ahead Logging) журнал предзаписи изменений в БД. Ведение файлов WAL - это стандартный метод обеспечения целостности данных в СУБД Postgres, при котором изменения в файлах БД записываются только после того, как эти изменения были занесены в журнал WAL, что обеспечивает возможность восстановить БД методом REDO (восстановление с воспроизведением) после сбоев.

1. Назначение программы Pg2PgSync

Программа Pg2PgSync - https://forstelecom.ru/pg2pgsync - предназначена для передачи изменений данных из исходной БД PostgreSQL в целевую БД PostgreSQL.

Программа может быть применена в случае необходимости интеграции различных баз данных.

Pg2PgSync предоставляет следующие возможности:

- Синхронизация данных как для небольших наборов таблиц, так и для множества схем
- Реализация многопоточной буферизованной обработки потока сообщений логической репликации СУБД PostgreSQL. Поддерживается работа с пулом слотов репликации.
- Разнесение таблиц в разные слоты репликации позволяет повысить скорость передачи данных.
- Поддержка пакетного (batch) режима, что ускоряет обработку массовых операций с отдельными таблицами
- Настройка параметров режима работы утилиты в конфигурационных файлах формата json
- Декодирование сообщений из файлов логической репликации PostgreSQL в операторы Postgres SQL
- Выполнение преобразованных Postgres SQL команд изменения данных
- Возможность многопоточной репликации данных с разделением множества таблиц на группы
- Оптимизация потока операций. Выполняется агрегация последовательности операций над одной строкой таблицы в пределах транзакции. Это позволяет сократить количество операций изменения данных, исполняемых в целевой БД, тем самым увеличивается скорость обработки и сокращается нагрузка на сборщик мусора (VACUUM).
- Вывод сообщений об ошибках репликации
- Возможность ручного и автоматического пропуска ошибок. Пропущенные ошибки журналируются.
- Устойчивость к сбоям - возможность продолжения репликации после устранения причин ошибки

- Возможность работы в составе кластера. При этом поток сообщений репликации получается с ведомого сервера (реплики) кластера Postgres. Это позволяет снизить нагрузку на основной сервер БД.
- Монитор процесса синхронизации данных
- Ведение файла журнала выполненных действий.

Применение программы Pg2PgSync позволяет выполнять быструю репликацию больших потоков данных без потерь, например, в целях синхронизации. Программа позволяет осуществлять мониторинг процесса репликации, отображение ошибок, возобновление репликации после исправления ошибок или пропуск ошибок. Модуль оптимизации (агрегации) позволяет сократить объем операций изменения данных, исполняемых в целевой БД.

2. Настройка сервера Postgres – источника данных

Настройка сервера СУБД Postgres заключается в выполнении следующих предварительных действий:

- Настроить потоковую репликацию данных

- Выделить достаточно места для хранения файлов WAL

- Создать публикации таблиц и слотов репликации

- Настроить поддержку Batch режима

- Учесть ограничения поддержки операции truncate table.

2.1. Активация потоковой репликации

Для активации потоковой репликации необходимо добавить следующие настройки в конфигурационный файл сервера Postgres (postgresql.conf):

wal_level = logical

max_wal_senders = 72

max_replication_slots = 36

wal_sender_timeout = 60s

max_slot_wal_keep_size = -1 (default)

track_commit_timestamp = off (default).

2.2. Хранилище для WAL

Сервер Postgres сохраняет файлы WAL до тех пор, пока все слоты логической репликации не подтвердят получение данных. Таким образом, если процесс репликации отстает от потока изменений сервера, то объем файлов WAL будет расти неограниченно. Необходимо обеспечить достаточный объем дискового хранилища для сохранения файлов WAL.

2.3. Публикации таблиц и слоты репликации

Для осуществления репликации необходимо создать в БД Postgres публикации таблиц и слоты репликации. В публикацию может входить список схем или список таблиц:

CREATE PUBLICATION pg2pg_01 FOR TABLES IN SCHEMA test1;

CREATE PUBLICATION pg2pg _02 FOR TABLE test2.table_a, test2.table_b;

Если имеются сегментированные таблицы, то необходимо добавить фразу WITH (publish_via_partition_root=true).

Теперь создать слоты репликации:

SELECT PG_CREATE_LOGICAL_REPLICATION_SLOT('pg2pg _01', 'pgoutput');

SELECT PG_CREATE_LOGICAL_REPLICATION_SLOT('pg2pg _02', 'pgoutput');

Слоты репликации могут быть явно описаны в конфигурационном файле (параметр slots). Однако, при большом количестве слотов удобнее пользоваться шаблоном (pg_slots_template). Программа автоматически подключит все слоты, удовлетворяющие данному шаблону. Например, при значении шаблона pg2pg _%, будут подключены слоты pg2pg _01 и pg2pg _02, приведенные выше. При этом для каждого слота должна существовать публикация, имя которой совпадает с именем слота.

2.4. Batch режим

Программа Pg2PgSync поддерживает режим пакетной передачи однотипных операций DML в целевую БД (insert или update или delete) с одной таблицей. Например, если есть таблица, в которую происходит постоянная массовая вставка записей, то такую таблицу необходимо выделить в отдельную пару публикация/слот. Pg2PgSync обнаруживает в очереди серию одинаковых операций с одной таблицей и обрабатывает их в пакетном режиме. Это значительно ускоряет выполнение массовых операций в целевой БД. Запросы из очереди выбираются в пакет до тех пор, пока не произойдет одно из:

- больше нет запросов в очереди;

- следующая операция в очереди отличается от операции пакета (в том числе операция commit);

- достигнуто максимальное кол-во записей в пакете (параметр настройки tdb_max_batch_size).

2.5. Truncate table

Поддерживается операция truncate table.

3. Мониторинг и исправление ошибок

В процессе работы программа получает поток сообщений от БД Postgres, преобразует их в операторы SQL и выполняет в целевой БД Postgres. В случае возникновения ошибки программа останавливает обработку сообщений в слоте, в котором эта ошибка произошла. Средство мониторинга (монитор, см. ниже) позволяет обнаружить возникновение ошибки, а также просмотреть детали ошибки. Ошибка может произойти на стороне исходной БД Postgres, на стороне целевой БД Postgres, или в самой программе.

Если ошибка устранима, то необходимо ее устранить, после чего перезапустить программу для возобновления работы слота.

Если ошибка произошла в программе, необходимо сообщить детали ошибки разработчику.

Следует понимать, что продолжение работы исходной БД Postgres с остановленной программой Pg2PgSync или остановленной репликацией в одном или нескольких слотах, приводит к постоянному росту

объема файлов WAL. Необходимо максимально оперативно устранить ошибку и снова запустить процесс репликации.

Программа Pg2PgSync содержит средства пропуска ошибок репликации, однако использовать их не рекомендуется, т.к. это может привести к нарушению целостности целевой БД и последующему накоплению ошибок.

4. Монитор процессов репликации БД

Программа включает в себя средство наблюдения за процессом репликации – монитор, представляющий собой встроенный HTTP-сервер.

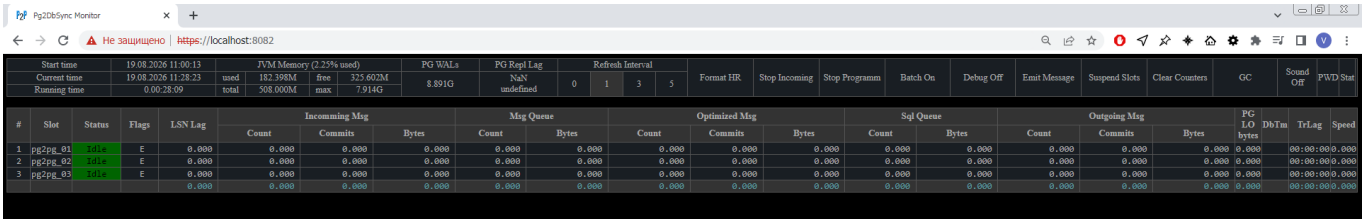

Рисунок 1. Экран монитора процессов репликации

Экран монитора разделен на две части (см. Рисунок 1). В верхней части расположена панель общей информации и кнопок управления. В нижней части – информация по слотам репликации.

4.1. Панель общей информации

Параметры общей информации репликации на верхней панели монитора (см. Рисунок 1):

Start Time – дата/время запуска программы Pg2PgSync.

Current Time – текущее время.

Running Time – время работы программы (количество дней и часы:минуты:секунды).

JVM Memory – информация о потреблении памяти JVM.

PG WALs – текущий объем файлов WAL сервера Postgres.

Refresh Interval – период обновления информации в секундах. 0 – обновление остановлено.

Format Full/HR – выбор формата отображения числовых величин.

- Full – в байтах или штуках
- HR (Human Readable) – отображение значений с суффиксами K, M, G, T, P, E (килобайт, мегабайт, и т.д., или тысяч, миллионов, и т.д.)
- Stop Incoming / Resume Incoming – приостановка (возобновление) получения потока сообщений репликации от исходного сервера Postgres.

Stop Program – остановка программы. При этом слоты репликации закрываются, содержимое очередей удаляется, программа останавливается. Повторный запуск программы осуществляется вручную.

Restart Slot – возобновление работы слота после устранения ошибки.

Clear Counters – Очистка счетчиков сообщений репликации. Не рекомендуется использовать. Предназначена для отладочных целей. Допустимо сбрасывать счетчики в состоянии покоя: очереди пусты, новых сообщений репликации не поступает.

Debug On/Off – Включение/выключение записи в лог отладочной информации. Не рекомендуется использовать. Предназначена для отладочных целей. Вывод отладочной информации существенно замедляет работу программы.

GC – запуск сборщика мусора (Garbage Collection) JVM. Не рекомендуется использовать. Предназначена для отладочных целей.

4.2. Информация по слотам репликации

Параметры по слотам репликации в нижней части монитора (см. Рисунок 1):

Slot – имя слота репликации сервера Postgres.

Status – состояние процесса репликации для данного слота в программе Pg2PgSync:

- Active – нормальный режим, ошибок нет.
- Error – ошибка. При нажатии на ячейку отобразится панель сообщения об ошибке.
- Throttling – превышен максимальный размер очереди в байтах (параметр настройки max_queue_size_mb) или в сообщениях репликации (параметр настройки max_queue_size_msg). При этом получение сообщений репликации от сервера Postgres приостанавливается.

Incoming Msg – полученные от сервера Postgres сообщения репликации.

Msg Queue – очередь полученных от исходного сервера Postgres сообщений репликации до обработки.

Sql Queue – очередь SQL операций, подготовленных для исполнения в целевой БД.

Outgoing Msg - сообщения репликации, успешно переданные в целевую БД.

Count - количество сообщений (включая commit);

Commits - количество сообщений commit;

Bytes – объем полученных сообщений в байтах.

PG LO bytes – Объем переданных данных, хранящихся в БД Postgres как Большие Объекты (Large Object).

Speed – скорость передачи данных килобайт/сек.

5. Запуск программы

Для запуска программы требуется среда выполнения java версии 17 или выше.

Программа может исполняться как на отдельном сервере, так и на одном из серверов СУБД, исходном или целевом.

Структура каталогов:

```
/opt/pg2pgsync/
      config.json
      Pg2PgSync.jar
      start.sh
/opt/pg2pgsync/lib
           gson-2.11.0.jar
           postgresql-42.7.13.jar
           ru.fors.utils.lightsmtp.jar
```

Скрипт запуска программы (start.sh):

```
#!/bin/bash

PATH=PATH:$JAVA_HOME/bin

java -Xmx8192m -Dfile.encoding=UTF-8 -jar Pg2PgSync.jar
```

Программа может быть запущена с указанием конфигурационного файла:

```
java –Xmx32768m -Dfile.encoding=UTF-8 -jar Pg2PgSync.jar –c config.json
```

Если конфигурационный файл не указан, программа пытается открыть файл config.json в текущем каталоге.

Программа создает в целевой БД таблицу зафиксированных транзакций (см. параметр tdb_lsn_table). Например, pg2pg.last_commited_lsn. Необходимо заранее создать схему для данной таблицы и дать пользователю, от имени которого программа подключается к целевой БД, права на создание таблиц в данной схеме.

6. Описание параметров конфигурационного файла

Все параметры программы описываются конфигурационным файлом в формате json, где приведены описание доступных параметров:

monitor_host – IP-адрес или имя хоста для привязки листенера монитора к интерфейсу. Если пустое значение, то листенер работает на всех интерфейсах.

monitor_port – номер порта, на котором будет запущен листенер монитора. Например, 8080.

monitor_props – стартовые параметры монитора:

switch_format – показать (1) или скрыть (0) кнопку переключения режима отображения.

incoming_state - показать (1) или скрыть (0) кнопку приостановки получения сообщений репликации от исходной БД.

stop_programm - показать (1) или скрыть (0) кнопку остановки программы.

clear_counters - показать (1) или скрыть (0) кнопку сброса значений счетчиков.

debug_mode – показать (1) или скрыть (0) кнопку переключения режима отладки (вкл/выкл).

refresh_interval – значение периода (сек) обновления данных монитора при старте программы. Может принимать значения 0, 1, 3, 5.

debug_mode – режим вывода отладочной информации (true/false). Существенно замедляет работу программы . Использовать вывод отладочной информации следует только на этапе тестирования.

stat_refresh_interval – период (сек) обновления статистической информации, получаемой экземплярами монитора и выводимой в файл. Рекомендуется значение 1.

log_file – файл вывода отладочной информации и сообщений об ошибках. Если указана пустая строка, то выводятся на консоль.

stat_file – файл вывода статистической информации. Если указана пустая строка, файл не записывается.

msg_file_dir – каталог для записи файлов сообщений репликации. Если указана пустая строка, файлы не записываются. Экспериментальное, использовать не рекомендуется.

alerts_enable – включить режим отправки сообщений об ошибках на адреса электронной почты и/или Telegram (true/false).

mail_host – IP-адрес или имя хоста сервера SMTP.

mail_port – номер порта сервера SMTP. Например, 465.

mail_tls – использовать протокол TLS (true/false).

mail_ssl – использовать протокол SSL (true/false).

mail_username – имя пользователя (логин) почтового сервера.

mail_password – пароль почтового сервера.

mail_subject – тема сообщений. Например, "pg2pg alert".

mail_sender – отправитель сообщений.

mail_recipients – список получателей почтовых сообщений через запятую".

tlg_bot_token – токен бота Telegram для отправки сообщений.

tlg_recipients – список идентификаторов абонентов Telegram (chat_id) через запятую.

tdb_db_url – строка соединения с целевой БД Postgres в следующем формате: jdbc:postgresql://<host>:5432/<dbname>.

tdb_username – имя пользователя целевой БД Postgres.

tdb_password – пароль пользователя целевой БД Postgres.

tdb_lsn_table – имя таблицы (с указанием схемы) для хранения идентификаторов зафиксированных (commited) транзакций. Например, "pg2pg.last_commited_lsn".

tdb_max_batch_size – максимальное количество сообщений, которые могут быть переданы в целевую БД одним пакетом. Например, 100000.

pg_db_url – строка соединения с исходной БД Postgres в следующем формате: jdbc:postgresql://<host>:5432/<dbname>.

pg_username – имя пользователя БД Postgres.

pg_password – пароль пользователя БД Postgres.

pg_status_interval – максимальный интервал отправки серверу Postgres статуса.

max_queue_size_mb – максимальный размер очереди в мегабайтах (сумма размеров сообщений репликации во всех слотах). Например, 100.

max_queue_size_msg – максимальный размер очереди в сообщениях репликации (суммарное количество сообщений репликации во всех слотах). Например, 200000.

pg_slots_template – шаблон имен слотов репликации. Например, "pg2pg%". Программа извлекает слоты, предварительно настроенные в исходной БД Postgres по заданному шаблону. Так же должны быть предварительно созданы публикации, имена которых совпадают с именами слотов.

slots – перечень слотов и публикаций, которые будут участвовать в репликации. Слоты и публикации должны быть предварительно настроены в исходной БД Postgres. Данный параметр работает только в том случае, если не задано значение шаблона (pg_slots_template). Например:

```
"slots":
[
   {"slot":"pg2pg1", "publications":"pg2pg_repltest1"},
   {"slot":"pg2pg2", "publications":"pg2pg_repltest2"}
],
```

pg_types – описание типов БД Postgres с указанием метода отображения целевую БД.

oid – числовой идентификатор типа БД Postgres.

name – наименование типа БД Postgres.

bind – метод отображения значения при выполнении операции в целевой БД:

S – as String
N – Numeric (as String with replace decimal point)
D – Date
T – Timestamp
TZ – Timestamp with timezone
TI – Interval (не реализовано)
H – Binary (Hex String <= 32767)
B – Binary (Hex String to byte[] conversion)
L – Lagre Object (BLOB by stream).

map – варианты преобразования значений, заданные в виде коллекции. Например, "t:1;f:0;" – преобразует значение "t" в значение "1", а "f" в "0".

func – альтернативная возможность преобразования значения, заданная в виде SQL-выражения. Например, "case ? when 't' then '1' when 'f' then '0' else 'BADVAL' end". В данном случае в SQL оператор помещается приведенное выражение, а исходное значение привязывается к подстановочному символу ? (placeholder). Конечное значение будет вычислено целевым сервером при выполнении SQL-оператора.

7. Пример конфигурационного файла

Пример конфигурационного файла `config.json`

```
{
  "monitor_host":"",
  "monitor_port":8080,
  "monitor_params":{"switch_format":1, "incoming_state":1, "stop_programm":1,
"restart_slot":0, "clear_counters":0, "debug_mode":0, "refresh_interval":1},

  "debug_mode":false,
  "stat_refresh_interval":1,

  "log_file":"",  /*pg2pgsync.log*/
  "stat_file":"", /*pg2pgsync.stat*/
  "msg_file_dir":"",

  "alerts_enable":false,
  "mail_host":"smtp.org.ru",
  "mail_port":465,
  "mail_tls":false,
  "mail_ssl":true,
  "mail_username":"alerter@org.ru",
  "mail_password":"password",
  "mail_subject":"pg2pg alert",
  "mail_sender":"alerter@org.ru",
  "mail_recipients":"ivanov@org.ru,sidorov@org.ru",
  "tlg_bot_token":"1234567890:BBQ-p5mxV1cOtF7CkvnNrsi7a7od56MNgwk",
  "tlg_recipients":"111111,222222",

  "tdb_db_url":" jdbc:postgresql://target_pgserver:5432/postgres ",
  "tdb_username":"postgres",
  "tdb_password":"postgres",
  "tdb_lsn_table":"pg2pg.last_commited_lsn",
  "tdb_max_batch_size":100000,

  "pg_db_url": "jdbc:postgresql:// source_pgserver:5432/postgres",
```

```
  "pg_username":"postgres",
  "pg_password":"postgres",

  "pg_status_interval":100,
  "max_queue_size_mb":100,
  "max_queue_size_msg":200000,
  "pg_slots_template":"pg2pg_%",

  "slots":[],

  "pg_types":
  [
      { "oid":16,   "name":"bool",        "bind":"S", map":"t:1;f:0;"},
      { "oid":17,   "name":"bytea",       "bind":"B" },
      { "oid":18,   "name":"char",        "bind":"S" },
      { "oid":20,   "name":"int8",        "bind":"N" },
      { "oid":21,   "name":"int2",        "bind":"N" },
      { "oid":23,   "name":"int4",        "bind":"N" },
      { "oid":25,   "name":"text",        "bind":"S" },
      { "oid":26,   "name":"oid",         "bind":"L" },
      { "oid":700,  "name":"float4",      "bind":"N" },
      { "oid":701,  "name":"float8",      "bind":"N" },
      { "oid":1042, "name":"bpchar",      "bind":"S" },
      { "oid":1043, "name":"varchar",     "bind":"S" },
      { "oid":1082, "name":"date",        "bind":"D" },
      { "oid":1083, "name":"time",        "bind":"" },
      { "oid":1114, "name":"timestamp",   "bind":"T" },
      { "oid":1184, "name":"timestamptz", "bind":"TZ" },
      { "oid":1186, "name":"interval",    "bind":"TI" },
      { "oid":1700, "name":"numeric",     "bind":"N" },
      { "oid":1266, "name":"timetz",      "bind":"" }
   ]
}
/*
      S  String
      N  Numeric (as String with replace decimal point)
      D  Date
      T  Timestamp
      TZ Timestamp with timezone
      TI Interval (not implemented)
      H  Binary (Hex String <= 32767)
      B  Binary (Hex String to byte[] conversion)
      L  Lagre Object (BLOB by stream)
  */
```
