EDB
Postgres® for the AI Generation

Speeding up logical replication setup

Euler Taveira
Principal Engineer
Oct 25, 2024

Motivation

- Initial data synchronization is slow
  - workload causes backlog while copying big tables
- Disk space
  - big tables + workload
  - WAL retention to wait until COPY is done
- migration from different major versions
- other replication tools
  - pglogical_createsubscriber

Initial data synchronization

- it is done separately for each table
  - tablesync worker (max_sync_workers_per_subscription)
  - one tablesync worker per table
  - COPY … TO STDOUT
  - catchup phase
  - SUBREL_STATE_SYNCDONE -> SUBREL_STATE_READY
  - handle it to apply worker

Proposal

- pg_createsubscriber
- creates a logical replica using a physical replica
- is it worth it?
  - 20 GB: 13x
  - 10 GB: 7x
  - https://www.postgresql.org/message-id/CAHv8RjKYo1Xdkj6WhFZ32Mg4-%2B9EuNgau%3DJZH8U7_NPv2gFmGQ%40mail.gmail.com

Development

- Initial email: 2022-02-21
- Messages: **327**
- Commit: 2024-03-25
- Reviewers: **7**

$ wc -l src/bin/pg_basebackup/pg_createsubscriber.c
2253 src/bin/pg_basebackup/pg_createsubscriber.c

Commit

commit: d44032d0146306971cd5ccf232fe37269717d6f2
author: Peter Eisentraut <peter@eisentraut.org>
date: Mon, 25 Mar 2024 12:30:55 +0100
committer: Peter Eisentraut <peter@eisentraut.org>
date: Mon, 25 Mar 2024 12:42:47 +0100
pg_createsubscriber: creates a new logical replica from a standby server

[...]

Author: Euler Taveira <euler.taveira@enterprisedb.com>
Reviewed-by: Hayato Kuroda <kuroda.hayato@fujitsu.com>
Reviewed-by: Amit Kapila <amit.kapila16@gmail.com>
Reviewed-by: Shlok Kyal <shlok.kyal.oss@gmail.com>
Reviewed-by: Vignesh C <vignesh21@gmail.com>
Reviewed-by: Shubham Khanna <khannashubham1197@gmail.com>
Reviewed-by: Ashutosh Bapat <ashutosh.bapat.oss@gmail.com>
Reviewed-by: Peter Eisentraut <peter@eisentraut.org>
Discussion:
https://www.postgresql.org/message-id/flat/5ac50071-f2ed-4ace-a8fd-b892cffd33eb@www.fastmail.com

Design

- initial physical replica
- run on the target server
  - it is a server application
- all tables in the specified databases are included in the logical replication setup
- pair of publication and subscription objects are created for each specified database
- source and target servers must have the same major version as pg_createsubscriber
- user name (target server) must have privileges to create subscriptions
- ensure the configuration parameters are sufficient for the logical replication setup in both servers

Behind the scenes

- target server must be stopped
- start the server with the specified command-line options
  - avoid connection during transformation steps
- check the primary for configuration parameters
  - wal_level = logical
  - max_replication_slots >= current + number of specified databases
  - max_wal_senders >= current + number of specified databases
- check the standby for configuration parameters
  - max_replication_slots >= current + number of specified databases
  - max_logical_replication_workers >= number of specified databases
  - max_worker_processes >= 1 + number of specified databases
- stop the target server after verification
- create the required objects for each database on publisher
  - publication, replication slot
  - remember the LSN from the latest replication slot (**key point**)

Behind the scenes

- setup the recovery
  - recovery_target_lsn: LSN from previous step
  - unset if there are any recovery_target_* parameters
- start the target server
  - wait until accepting connections
  - don’t start logical replication yet (max_logical_replication_workers=0)
- wait unit the target server to be promoted
  - recovery_timeout
  - if it fails after this point, you must recreate the physical replica
- drop pre-existing subscriptions on the target server
- drop publication from the target server (earlier step)
- create the subscription for each database on the target server
  - create subscription (enabled = false)
  - set replication progress with the LSN from a previous step
  - enable subscription

Behind the scenes

- remove replication slot used by the physical replication, if any
- remove failover replication slot if they exist on the target server
- stop the target server
- modify the system identifier
  - pg_resetwal

Interface

-d, --database=DBNAME database in which to create a subscription
-D, --pgdata=DATADIR location for the subscriber data directory
-n, --dry-run dry run, just show what would be done
-p, --subscriber-port=PORT subscriber port number (default 50432)
-P, --publisher-server=CONNSTR publisher connection string
-s, --socketdir=DIR socket directory to use (default current dir.)
-t, --recovery-timeout=SECS seconds to wait for recovery to end
-U, --subscriber-username=NAME user name for subscriber connection
-v, --verbose output verbose messages
--config-file=FILENAME use specified main server configuration file when running target cluster
--publication=NAME publication name
--replication-slot=NAME replication slot name
--subscription=NAME subscription name

Object names

- name pattern: pg_creatatesubscriber_%u_%x
- %u: database OID
- %x: random integer
- it is used in case you don't specify the –publication, –subscription or –replication-slot option
- the replication slot has a special treatment
  - if not specified, assign subscription name
  - same rule as CREATE SUBSCRIPTION

Run: dry run mode

```
$ pg_createsubscriber --dry-run -d postgres -D /standby -P "host=192.168.0.3 port=5432"
2024-10-20 11:04:49.695 EEST [7367] LOG:  starting PostgreSQL 17.0 on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit
2024-10-20 11:04:49.734 EEST [7367] LOG:  listening on Unix socket "/tmp/.s.PGSQL.50432"
2024-10-20 11:04:49.857 EEST [7370] LOG:  database system was shut down at 2024-10-20 11:03:21 EEST
2024-10-20 11:04:49.894 EEST [7367] LOG:  database system is ready to accept connections
pg_createsubscriber: error: target server must be a standby
```

Run: dry run mode

```
$ pg_createsubscriber --dry-run -d postgres -D /standby -P "host=192.168.0.3 port=5432"
2024-10-20 11:07:12.088 EEST [7462] LOG:  starting PostgreSQL 17.0 on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit
2024-10-20 11:07:12.157 EEST [7462] LOG:  listening on Unix socket "/tmp/.s.PGSQL.50432"
2024-10-20 11:07:12.314 EEST [7471] LOG:  database system was shut down in recovery at 2024-10-20 11:06:50 EEST
2024-10-20 11:07:12.320 EEST [7471] LOG:  entering standby mode
2024-10-20 11:07:12.378 EEST [7471] LOG:  redo starts at 0/F000110
2024-10-20 11:07:12.378 EEST [7471] LOG:  consistent recovery state reached at 0/10000000
2024-10-20 11:07:12.378 EEST [7462] LOG:  database system is ready to accept read-only connections
2024-10-20 11:07:12.462 EEST [7472] LOG:  started streaming WAL from primary at 0/10000000 on timeline 1
pg_createsubscriber: error: publisher requires wal_level >= "logical"
```

Run: dry run mode

```
$ pg_createsubscriber --dry-run -d postgres -D /standby -P "host=192.168.0.3 port=5432" -v
pg_createsubscriber: validating publisher connection string
pg_createsubscriber: validating subscriber connection string
pg_createsubscriber: checking if directory "/tmp/pgdata2" is a cluster data directory
pg_createsubscriber: getting system identifier from publisher
pg_createsubscriber: system identifier is 7429253919236734606 on publisher
pg_createsubscriber: getting system identifier from subscriber
pg_createsubscriber: system identifier is 7429253919236734606 on subscriber
pg_createsubscriber: starting the standby server with command-line options
2024-10-20 11:21:20.477 EEST [7986] LOG:  redirecting log output to logging collector process
2024-10-20 11:21:20.477 EEST [7986] HINT:  Future log output will appear in directory "log".
pg_createsubscriber: server was started
pg_createsubscriber: checking settings on subscriber
pg_createsubscriber: checking settings on publisher

[...]
```

Run

```
$ pg_createsubscriber -d postgres -D /standby -P "host=192.168.0.3 port=5432" -v

[...]

pg_createsubscriber: stopping the subscriber
pg_createsubscriber: server was stopped
pg_createsubscriber: creating publication "pg_createsubscriber_5_2c7db45c" in database "postgres"
pg_createsubscriber: creating the replication slot "pg_createsubscriber_5_2c7db45c" in database
"postgres"
pg_createsubscriber: create replication slot "pg_createsubscriber_5_2c7db45c" on publisher
pg_createsubscriber: starting the subscriber
2024-10-20 11:28:46.368 EEST [8210] LOG:  redirecting log output to logging collector process
2024-10-20 11:28:46.368 EEST [8210] HINT:  Future log output will appear in directory "log".
pg_createsubscriber: server was started
pg_createsubscriber: waiting for the target server to reach the consistent state
pg_createsubscriber: target server reached the consistent state
pg_createsubscriber: hint: If pg_createsubscriber fails after this point, you must recreate the physical
replica before continuing.
```

Run

[...]

pg_createsubscriber: dropping publication "pg_createsubscriber_5_2c7db45c" in database "postgres"
pg_createsubscriber: creating subscription "pg_createsubscriber_5_2c7db45c" in database "postgres"
pg_createsubscriber: setting the replication progress (node name "pg_24576", LSN 0/DD0D648) in database "postgres"
pg_createsubscriber: enabling subscription "pg_createsubscriber_5_2c7db45c" in database "postgres"
pg_createsubscriber: dropping the replication slot "primary" in database "postgres"
pg_createsubscriber: stopping the subscriber
pg_createsubscriber: server was stopped
pg_createsubscriber: modifying system identifier of subscriber
pg_createsubscriber: system identifier is 7429256349443543031 on subscriber
pg_createsubscriber: running pg_resetwal on the subscriber
pg_createsubscriber: subscriber successfully changed the system identifier
pg_createsubscriber: Done!

Warning & Limitations

- pg_createsubscriber fails after promotion, you must recreate the physical replica
- connections to target server might fail during the transformation
  - starts the server with different settings (port, unix socket directory)
- If the target server was already transformed and DDL commands were executed on primary, logical replication might fail
- If the target server is a synchronous replica, transaction commits might wait for replication while pg_createsubscriber is running
- pg_createsubscriber runs pg_resetwal so if the target server has replicas you should rebuild them
- pg_createsubscriber creates subscriptions with two-phase commit disabled
- If the replication is using primary_slot_name, this replication slot name is removed

TODO

- base backup support
- option to start the server when it is done
- option to drop non-specified databases from the target server
- option to specify a set of tables to a specified database
- option to avoid invalidating physical standbys (cascading)
- option to not remove the failover replication slots

Questions?