PostgreSQL-Consulting

data egret

Troubleshooting streaming replication

Alexey Lesovsky

alexey.lesovsky@dataegret.com

01 Quick introduction

- WAL and replication internals.
- Replication setup.

02 Troubleshooting tools

- 3rd party tools.
- Builtin tools.

03 Troubleshooting cases

- Symptoms and problems.
- Detection and solutions.
- Lessons learned.

01

Quick introduction

01 Goals

Better understanding of streaming replication.

How to quickly find and fix problems.

https://goo.gl/Mm3ugt

01 Agenda

Write-Ahead Log.

Streaming replication internals.

Streaming replication setup.

Troubleshooting tools overview (3-rd party).

Troubleshooting tools overview (builtin).

Troubleshooting in practice.

Questions.

01 Write Ahead Log

Durability in ACID.

Almost all changes fixed in WAL.

pg_xlog/ (*pg_wal/*) directory in the DATADIR.

Synchronous WAL write by backends.

Asynchronous WAL write by WAL writer.

Recovery process relies on WAL.

01 Streaming replication internals

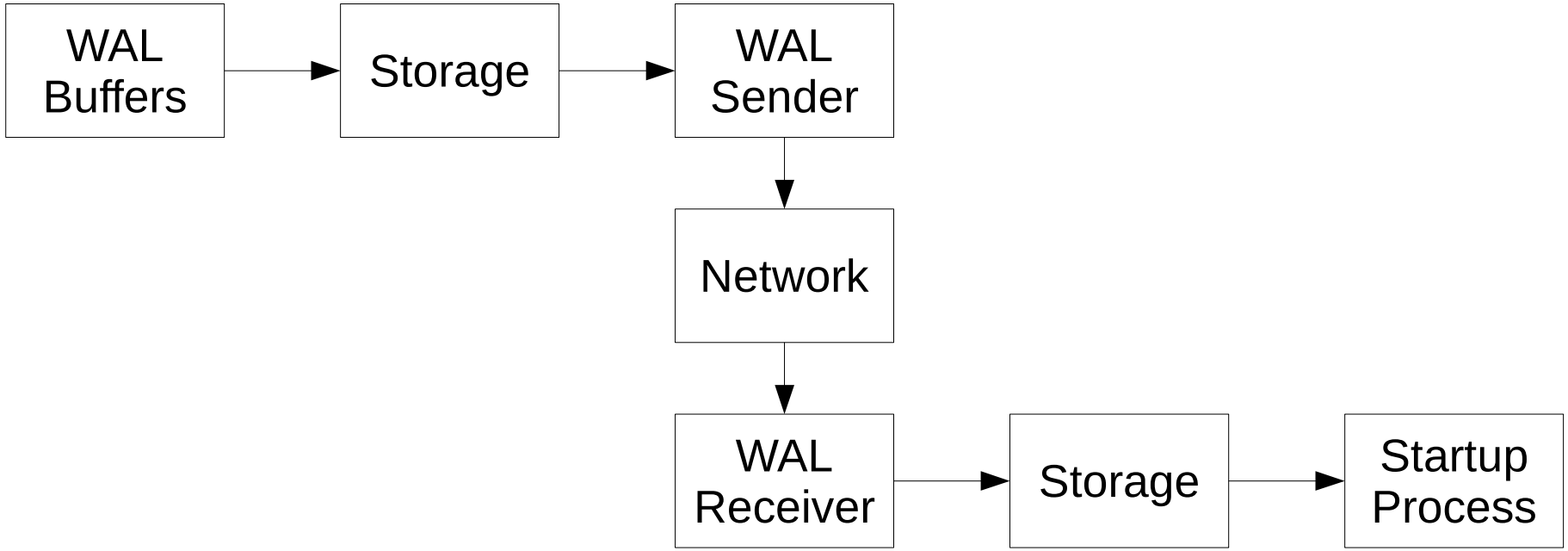
WAL Sender process.

WAL Receiver process.

Startup process (recovery).

Streaming replication *vs.* WAL archiving.

01 Streaming replication internals



01 Streaming replication setup

Master:

- *postgresql.conf*;
- Restart.

Standby:

- *pg_basebackup*;
- *postgresql.conf*;
- *recovery.conf* setup.

01 Master setup

wal_level, ***max_wal_senders***, ***max_replication_slots***.

archive_mode, *archive_command*.

wal_keep_segments.

wal_sender_timeout.

synchronous_standby_names.

01 Master setup pitfalls

wal_level, *archive_mode*, *max_wal_senders*, *max_replication_slots* – require restart.

wal_keep_segments – requires extra storage space.

wal_sender_timeout – reduce that, if network is bad.

synchronous_standby_names – master freezes if standby fails.

01 Standby setup

hot_standby.

max_standby_streaming_delay, *max_standby_archiving_delay*.

hot_standby_feedback.

wal_receiver_timeout.

01 Standby setup pitfalls

hot_standby – enables SELECT queries.

max_standby_streaming_delay – increases max possible lag.

hot_standby_feedback:

- postpones vacuum;
- potential tables/indexes bloat.

wal_receiver_timeout – reduce that, if network is bad.

01 Recovery.conf

primary_conninfo and/or ***restore_command***.

standby_mode.

trigger_file.

Recovery target:

- immediate;
- particular point/xid/timestamp;
- *recovery_min_apply_delay*.

01 Recovery.conf pitfalls

Any changes require restart.

02

Troubleshooting tools

02 3rd party troubleshooting tools

Top (procps).

Iostat (sysstat), *iotop*.

Nicstat.

pgCenter.

Perf.

02 3rd party troubleshooting tools

Top (procps) – CPU usage, load average, mem/swap usage.

Iostat (sysstat), *iotop* – storage utilization, process IO.

Nicstat – network interfaces utilization.

pgCenter – replication stats.

Perf – deep investigations.

02 Builtin troubleshooting tools

Statistics views.

Auxiliary functions.

pg_xlogdump utility.

02 Builtin troubleshooting tools

Statistics view:

- *pg_stat_replication*;
- *pg_stat_databases*, *pg_stat_databases_conflicts*;
- *pg_stat_activity*;
- *pg_stat_archiver*.

02 Builtin troubleshooting tools

Auxiliary functions:

- *pg_current_xlog_location*, *pg_last_xlog_receive_location*;
- *pg_xlog_location_diff*;
- *pg_xlog_replay_pause*, *pg_xlog_replay_resume*;
- *pg_is_xlog_replay_paused*.

02 Builtin troubleshooting tools

pg_xlogdump:

- Decodes and displays XLOG for debugging;
- Can give wrong results when the server is running.

```
pg_xlogdump -f -p /xlog_96 \
    $(psql -qAtX -c "select pg_xlogfile_name(pg_current_xlog_location())")
```

03

Troubleshooting cases

03 Troubleshooting cases

Replication lag.

pg_xlog/ bloat.

Long transactions and recovery conflicts.

Recovery process: 100% CPU usage.

03 Replication lag

Main symptom – answers differ between master and standbys.

Detection:

- *pg_stat_replication* and *pg_xlog_location_diff()*;
- *pg_last_xact_replay_timestamp()*.

03 Replication lag

```
# \d pg_stat_replication
          View "pg_catalog.pg_stat_replication"
      Column      |           Type           | Modifiers
------------------+--------------------------+-----------
 pid              | integer                  |
 usesysid         | oid                      |
 usename          | name                     |
 application_name | text                     |
 client_addr      | inet                     |
 client_hostname  | text                     |
 client_port      | integer                  |
 backend_start    | timestamp with time zone |
 backend_xmin     | xid                      |
 state            | text                     |
 sent_location    | pg_lsn                   |
 write_location   | pg_lsn                   |
 flush_location   | pg_lsn                   |
 replay_location  | pg_lsn                   |
 sync_priority    | integer                  |
 sync_state       | text                     |
```

03 Replication lag

```
# SELECT
  client_addr AS client, usename AS user, application_name AS name,
  state, sync_state AS mode,
  (pg_xlog_location_diff(pg_current_xlog_location(),sent_location) / 1024)::int as pending,
  (pg_xlog_location_diff(sent_location,write_location) / 1024)::int as write,
  (pg_xlog_location_diff(write_location,flush_location) / 1024)::int as flush,
  (pg_xlog_location_diff(flush_location,replay_location) / 1024)::int as replay,
  (pg_xlog_location_diff(pg_current_xlog_location(),replay_location))::int / 1024 as total_lag
FROM pg_stat_replication;

  client  |  user  |    name     |   state   | mode  | pending | write | flush | replay | total_lag
----------+--------+-------------+-----------+-------+---------+-------+-------+--------+-----------
 10.6.6.9 | repmgr | walreceiver | streaming | async |       0 |     0 |     0 | 410480 |    410480
 10.6.6.7 | repmgr | walreceiver | streaming | async |       0 |  2845 | 95628 | 112552 |    211025
 10.6.6.6 | repmgr | walreceiver | streaming | async |       0 |     0 |  3056 |   9496 |     12552
 10.6.6.8 | repmgr | walreceiver | streaming | async |  847582 |     0 |     0 |   3056 |    850638
```

03 Replication lag

```
# SELECT
  client_addr AS client, usename AS user, application_name AS name,
  state, sync_state AS mode,
  (pg_xlog_location_diff(pg_current_xlog_location(),sent_location) / 1024)::int as pending,
  (pg_xlog_location_diff(sent_location,write_location) / 1024)::int as write,
  (pg_xlog_location_diff(write_location,flush_location) / 1024)::int as flush,
  (pg_xlog_location_diff(flush_location,replay_location) / 1024)::int as replay,
  (pg_xlog_location_diff(pg_current_xlog_location(),replay_location))::int / 1024 as total_lag
FROM pg_stat_replication;

  client  |  user  |    name     |   state   | mode  | pending | write | flush | replay | total_lag
----------+--------+-------------+-----------+-------+---------+-------+-------+--------+-----------
 10.6.6.9 | repmgr | walreceiver | streaming | async |       0 |     0 |     0 | 410480 |    410480
 10.6.6.7 | repmgr | walreceiver | streaming | async |       0 |  2845 | 95628 | 112552 |    211025
 10.6.6.6 | repmgr | walreceiver | streaming | async |       0 |     0 |  3056 |   9496 |     12552
 10.6.6.8 | repmgr | walreceiver | streaming | async |  847582 |     0 |     0 |   3056 |    850638
```

03 Replication lag

```
# SELECT
  client_addr AS client, usename AS user, application_name AS name,
  state, sync_state AS mode,
  (pg_xlog_location_diff(pg_current_xlog_location(),sent_location) / 1024)::int as pending,
  (pg_xlog_location_diff(sent_location,write_location) / 1024)::int as write,
  (pg_xlog_location_diff(write_location,flush_location) / 1024)::int as flush,
  (pg_xlog_location_diff(flush_location,replay_location) / 1024)::int as replay,
  (pg_xlog_location_diff(pg_current_xlog_location(),replay_location))::int / 1024 as total_lag
FROM pg_stat_replication;

  client  |  user  |    name     |   state   | mode  | pending | write | flush | replay | total_lag
----------+--------+-------------+-----------+-------+---------+-------+-------+--------+-----------
 10.6.6.9 | repmgr | walreceiver | streaming | async |       0 |     0 |     0 | 410480 |    410480
 10.6.6.7 | repmgr | walreceiver | streaming | async |       0 |  2845 | 95628 | 112552 |    211025
 10.6.6.6 | repmgr | walreceiver | streaming | async |       0 |     0 |  3056 |   9496 |     12552
 10.6.6.8 | repmgr | walreceiver | streaming | async |  847582 |     0 |     0 |   3056 |    850638
```

03 Replication lag

```
# SELECT
  client_addr AS client, usename AS user, application_name AS name,
  state, sync_state AS mode,
  (pg_xlog_location_diff(pg_current_xlog_location(),sent_location) / 1024)::int as pending,
  (pg_xlog_location_diff(sent_location,write_location) / 1024)::int as write,
  (pg_xlog_location_diff(write_location,flush_location) / 1024)::int as flush,
  (pg_xlog_location_diff(flush_location,replay_location) / 1024)::int as replay,
  (pg_xlog_location_diff(pg_current_xlog_location(),replay_location))::int / 1024 as total_lag
FROM pg_stat_replication;

  client  |  user  |    name     |   state   | mode  | pending | write | flush | replay | total_lag
----------+--------+-------------+-----------+-------+---------+-------+-------+--------+-----------
 10.6.6.9 | repmgr | walreceiver | streaming | async |       0 |     0 |     0 | 410480 |    410480
 10.6.6.7 | repmgr | walreceiver | streaming | async |       0 |  2845 | 95628 | 112552 |    211025
 10.6.6.6 | repmgr | walreceiver | streaming | async |       0 |     0 |  3056 |   9496 |     12552
 10.6.6.8 | repmgr | walreceiver | streaming | async |  847582 |     0 |     0 |   3056 |    850638
```

03 Replication lag

Network problems – *nicstat*.

Storage problems – *iostat*, *iotop*.

Recovery stucks – *top*, *pg_stat_activity*.

WAL pressure:

- *pg_stat_activity*, *pg_stat_progress_vacuum*;
- *pg_xlog_location_diff()*.

03 Replication lag

Network/storage problems:

- check workload;
- upgrade hardware.

Recovery stucks – wait or cancel queries on standby.

WAL pressure:

- Reduce amount of work;
- Reduce amount of WAL:
  - *full_page_writes = off, wal_compression = on, wal_log_hints = off*;
  - expand interval between checkpoints.

03 pg_xlog/ bloat

Main symptoms:

- unexpected increase in the usage of the disk space;
- abnormal size of *pg_xlog/* directory.

03 pg_xlog/ bloat

Detection:

- *du -csh*;
- *pg_replication_slots*, *pg_stat_archiver*;
- errors in postgres logs.

03 pg_xlog/ bloat

Problems:

- Massive CRUD.
- Unused slot.
- Broken *archive_command*.

03 pg_xlog/ bloat

Solutions:

- check replication lag;
- reduce *checkpoints_segments/max_wal_size, wal_keep_segments;*
- change reserved space ratio (ext filesystems);
- add an extra space (LVM, ZFS, etc);
- drop unused slot or fix slot consumer;
- fix WAL archiving;
- checkpoint, checkpoint, chekpoint.

03 Recovery conflicts

Main symptoms – errors in postgresql or application logs.

postgres.c:errdetail_recovery_conflict():

- *User was holding shared buffer pin for too long.*
- *User was holding a relation lock for too long.*
- *User was or might have been using tablespace that must be dropped.*
- *User query might have needed to see row versions that must be removed.*
- *User transaction caused buffer deadlock with recovery.*
- *User was connected to a database that must be dropped.*

03 Recovery conflicts

Detection:

- *pg_stat_databases* + *pg_stat_databases_conflicts*;
- postgresql logs.

03 Recovery conflicts

Problems:

- queries are cancelled too often;
- long transactions on a standby – check *pg_stat_activity*;
- huge apply lag – check *pg_stat_replication*.

03 Recovery conflicts

Solutions:

- increase streaming delay (potentially causes lag);
- enable *hot_standby_feedback* (potentially causes bloat);
- rewrite queries;
- setup dedicated standby for long queries.

03 Recovery 100% CPU usage

Main symptoms:

- huge apply lag;
- 100% CPU usage by recovery process.

03 Recovery 100% CPU usage

Detection:

- *top* – CPU usage;
- *pg_stat_replication* – amount of lag.

03 Recovery 100% CPU usage

Investigation:

- *perf top/record/report* (required debug symbols);
- *pg_xlogdump*.

03 Recovery 100% CPU usage

Solutions:

- depend on investigation' results;
- change problematic workload (if found).

03 Lessons learned

Streaming replication problems are always distributed.

There are many sources of problems:

- system resources, app/queries, workload.

Always use monitoring.

Learn how to use builtin tools.

Links

PostgreSQL official documentation – The Statistics Collector
https://www.postgresql.org/docs/current/static/monitoring-stats.html

PostgreSQL Mailing Lists (*general, performance, hackers*)
https://www.postgresql.org/list/

PostgreSQL-Consulting company blog
http://blog.postgresql-consulting.com/


Thanks for watching!