Tale from Trenches: How autovacuum, streaming replication, batch query took down availability and performance

Sushant Pawar –

Database Solution Consultant

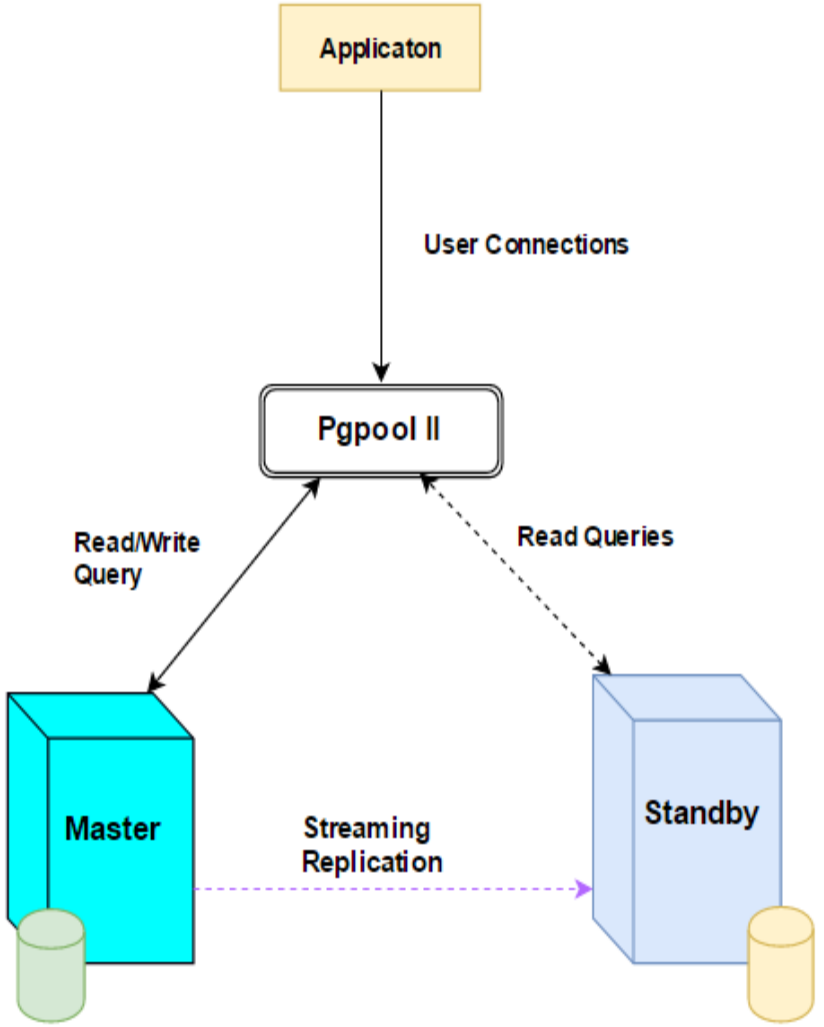
Agenda

- Client Architecture
- Issue 1 - Occurred on production server
- Issue 2 – Occurred on standby server
- Actions Taken
- Issue 2 Reoccurrence
- Recreation of issue in Pre-Prod server
- Post recreation analysis of issue in Pre-Prod server
- Resolution
- Key Learnings

Client Architecture

Configuration:

- 4 Core CPU and 32GB RAM
- PostgreSQL 9.4.3 Installation
- HA setup with pgpool



Issue 1: Occurred on the production server

- High CPU usage
- 500+ concurrent sessions on server

Immediate action taken & Analysis

Action Taken

- Controlled failover to standby server
- Hardware upgradation on old production server

Analysis:

- Some frequently run costly queries
- Batch job which fires costly queries for select & delete
- Gradual increase in batch job execution
- Batch job causing VACUUM

Issue 2: Occurred on standby server

- Errors on standby server
- ERROR: Cancelling statement due to conflict with recovery
- DETAIL: User was holding a relation lock for too long.

- Reoccurrence of conflict usually occur between 1 to 2 pm

- Revert back to old configuration

- Enable hot_standby_feedback on standby server

- Thought on enabling max_standby_streaming_delay on standby server

- Check pg_stat_database_conflicts output

Issue Analysis - pg_stat_database_conflicts

```
postgres=# select * from pg_stat_database_conflicts;
 datid |     datname      | confl_tablespace | confl_lock | confl_snapshot | confl_bufferpin | confl_deadlock
-------+------------------+------------------+------------+----------------+-----------------+----------------
     1 | template1        |                0 |          0 |              0 |               0 |              0
 13051 | template0        |                0 |          0 |              0 |               0 |              0
 13056 | postgres         |                0 |          0 |              0 |               0 |              0
 16519 | test1            |                0 |          0 |              0 |               0 |              0
 16733 | ebank            |                0 |          0 |              0 |               0 |              0
 16907 | mydb             |                0 |          0 |              0 |               0 |              0
 16917 | fdw_example      |                0 |          0 |              0 |               0 |              0
 16983 | template_postgis |                0 |          0 |              0 |               0 |              0
(8 rows)
```

confl_tablespace:

Number of queries in this database that have been cancelled due to dropped tablespaces

confl_lock:

Number of queries in this database that have been cancelled due to lock timeouts

Issue Analysis - pg_stat_database_conflicts

confl_snapshot:

Number of queries in this database that have been cancelled due to old snapshots

confl_bufferpin:

Number of queries in this database that have been cancelled due to pinned buffers

confl_deadlock:

Number of queries in this database that have been cancelled due to deadlocks

Actions Taken –

Monitored the production server

- Monitored long running queries
- Monitored High CPU queries

Identified the bottleneck queries

- Identified locking queries
- High CPU queries for very short interval
- Made select queries faster
- Tuned the queries

Issue 2 Reoccurrence

- Conflict occurred past midnight
- Surprising to have issue re-occurred after doing some tuning

Recreation of issue in Pre-Prod server

- Logic inside the batch job
- Queries executing at the time of issue.
- Set log_autovacuum_min_duration parameter
- Run batch multiple times
- Two tables getting vacuumed during every batch run
- Vacuum acquiring exclusive lock on a table

Post recreation analysis of issue in Pre-Prod server

- In production environment, object's lock was getting replicated due to autovaccum

- Query cancellation on standby server

Resolution

- Interim resolution

  - ❑ Switch off auto vacuum on a table.
  - ❑ Manual vacuum full

- Long term resolution

  - ❑ Fix application logic

Key Learnings

- Always keep hot_standby_feedback on
- Tune frequent running queries and not just long running queries.
- Vacuum can also shrink pages if necessary
- Switch off autovacuum at table level if really necessary but avoid at database level
- Conflict because of Row version and lock are two different errors.

Q & A

Thank you