Online Upgrade and Maintenance

pgDay Paris 2015

Petr Jelínek, 2ndQuadrant
(petr@2ndQuadrant.com)
Why Online Upgrade

• Downtime is bad
• Online upgrade is safer
  – Check the data are what you expected after upgrade while application is still running
  – Roll-back the upgrade easily
  – Test if application is indeed running correctly
Before 9.4

• Trigger based solutions
  – High overhead
  – Slow copy
• pg_upgrade
  – Not online
  – Data loss risk if something goes wrong
  – Does not help with maintenance
9.4 and Up

• Logical decoding
  – Very little overhead
  – Well integrated

• UDR
  – Built using logical decoding
  – Easy to install and remove
  – Can be combined with pg_upgrade
Performance - WAL

pgbench scale 50, -c 32 -j 32
Performance - TPS

Custom workload throttled to <2s replication latency

- HotStandby
- UDR
- BDR
- Londiste3
- Slony

Sustainable TPS (higher is better)
Example Flow #1

1. 9.4 → bdr_init_copy → 9.4
2. UDR catchup → 9.5
3. pg_upgrade → 9.5
Example Flow #2

bdr.bdr_subscribe()
The Switch-over

• Upgrade is not the full story
• Changing database connection in application can mean downtime too
• Multiple applications are often connected to same database
pgbouncer

- Connection pooler
- Online/transparent database connection change
Setup #1

• Change postgresql.conf:
  - `max_wal_senders = 10`
  - `max_replication_slots = 10`
  - `wal_level = 'logical'`
  - `shared_preload-libraries = 'bdr'`

• Allow access in `pg_hba.conf`
Setup #2

- pgbouncer.ini
  - mydb = host=oldhost dbname=mydb
Switch-over #1

- Old server
  - CREATE EXTENSION bdr;

- New server
  - CREATE EXTENSION bdr;
  - SELECT bdr.bdr_subscribe('upgrade', 'dbname=foo host=oldhost', 'dbname=foo host=newhost');
  - SELECT bdr.bdr_node_join_wait_for_ready();
Switch-over #2

• pgbouncer.ini
  – mydb = host=newhost dbname=mydb

• pgbouncer
  – PAUSE;
  – RELOAD;
  – RESUME;
Demo
Online Maintenance

- Kernel upgrades
- OS (library) updates
- Hardware maintenance
Online Maintenance

- Similar use-case to online upgrade
- Similar solutions (replication)
- Easier in some regards
  - Same major version
  - Can be done using built-in physical replication
- Harder in others
  - Needs switch back
  - Or reverse the replication direction
The Future is Now

• BDR
  – Bi-Directional Replication
  – Switch-over and switch-back are just matter of changing the connection
  – Currently requires patched PostgreSQL
  – Patches submitted (some committed in 9.5)
BDR Setup #1

• Change postgresql.conf:
  - `max_wal_senders = 10`
  - `max_replication_slots = 10`
  - `wal_level = 'logical'`
  - `shared_preload_libraries = 'bdr'`

• Allow access in pg_hba.conf
BDR Setup #2

- Server #1:
  - CREATE EXTENSION bdr;
  - bdr.bdr_group_create('server1', 'hostname=host1 dbname=db');

- Server #2..n:
  - CREATE EXTENSION bdr;
  - bdr.bdr_group_join('server2', 'hostname=host2 dbname=db', 'hostname=host1 dbname=db');
Thanks and Links

• UDR/BDR
  – Developed and maintained by 2ndQuadrant (contributions welcome!)
    – http://bdr-project.org
    – http://bdr-project.org/docs
    – https://github.com/2ndQuadrant/bdr

• pgbouncer
  – https://pgbouncer.github.io
  – https://github.com/pgbouncer/pgbouncer

© 2ndQuadrant 2015