Repcloud

A repacker for PostgreSQL in cloud

Federico Campoli

FOSDEM 01 February 2020

Few words about the speaker

- Twitter: @4thdoctor_scarf
- Born in 1972
- Passionate about IT since 1982
- Joined the Oracle DBA secret society in 2004
- In love with PostgreSQL since 2006
- PostgreSQL tattoo on the right shoulder

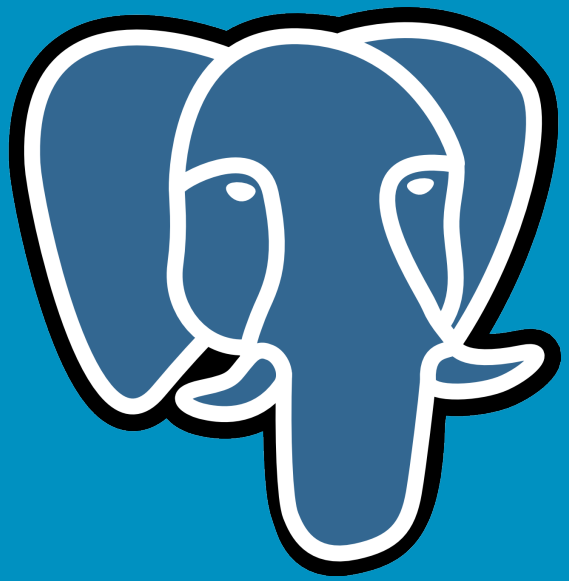
Disclaimer

- I'm not a developer
- I'm a DBA...
- Therefore I'm hated by everybody
- And I hate everybody
- So, to put things in the right perspective...

Tabs

Palpatine

PostgreSQL



- The most advanced open source database
- Gaining momentum
- Available as database as service on cloud providers
- Have its own drawbacks though

PostgreSQL MVCC

- MVCC, Multi Version Concurrency Control
- Allows concurrent read and writes with minimal lock
- PostgreSQL implementation very efficient
- Snapshots managed on the data page

PostgreSQL MVCC

- Transaction id, 4 bytes integer a.k.a. XID
- Two system fields within the row tracks its visibility
- The field **xmin** stores the transaction id which created the row
- The field **xmax** stores the transaction id which deleted the row
- **There is no field for tracking the updates!**

PostgreSQL MVCC

- In PostgreSQL there is no such thing like an update
- An update is an **insert/delete within the same transaction**
- Old row versions are left in place and removed by VACUUM
- When updated, the row may change its data page
- Indices need update when this happens
- Which may result in...

BLOAT!

Source https://imgflip.com/i/3al53c

Reducing the risk of bloat

- Database design
  - Data model should avoid tables with large rows
  - Group the most updated fields in separate tables
  - Use those fields to lookup the rest of the data
  - Remove unused indices
- Routine maintenance
  - VACUUM, low impact on the cluster, less effective on indices
  - REINXED highly effective blocking until PostgreSQL 11

Dealing with an existing bloat

- VACUUM FULL if you can afford blocking
- pg_repack if you can get the extension installed
- repcloud if all the other options are not feasible

REPacker in CLOUD

- Uses a similar strategy like pg_repack
- But without the physical file swap
- The tool can run on environments where privileges are limited
- e.g Heroku, RDS...

The repack flow

- Creates a new table like the original
- Creates a log table where to store the data changes
- A trigger on the original table logs the changes
- The data is copied from the original to the new table
- Replays the logged changes against the new table
- Then attempts to swap

Dependency hell

The hellish dependency system

- The swap is not trivial
- Because of the PostgreSQL's dependency system
- When renaming or changing the table schema all the dependent objects follow the table change
- On the new table all the objects must be built from scratch using the old table as a model

Rebuilding the objects

The hellish dependency system

- Sequences
- Views
- Materialised views
- Foreign keys

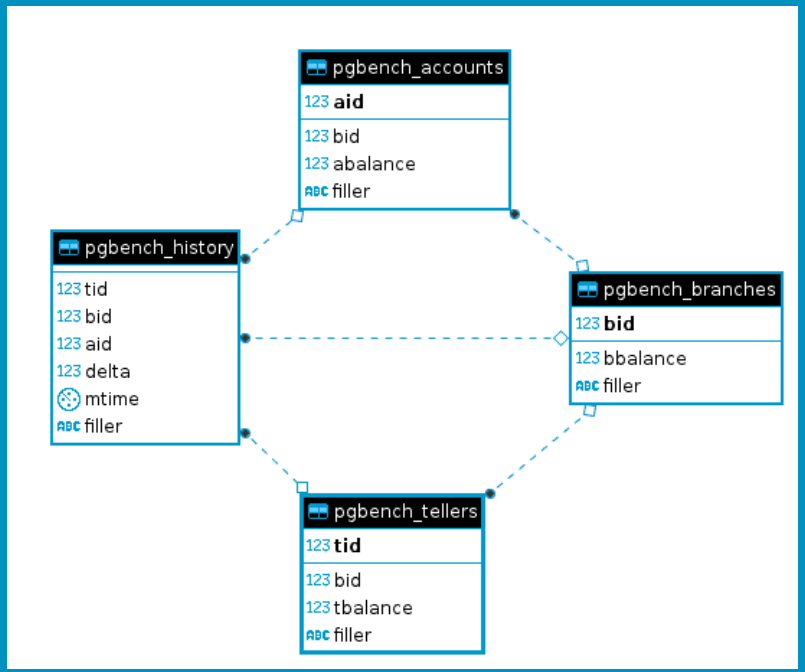
Rebuilding the objects

Any other object which is not related directly with the original table but is using those objects must be rebuilt as well!

Deadlock warning

Deadlock detected

The foreign keys handling may cause a Deadlock.
In particular if we have inter dependent foreign keys.



Deadlock warning

Deadlock detected

PostgreSQL's Deadlock resolution kills randomly one of the locking sessions.
Repcloud can resolve the Deadlock by killing or cancelling the other session before the Deadlock resolution kicks in.
However this may not be a good strategy for some systems.

Deadlock warning

Deadlock detected

Repcloud can prepare the table swap leaving everything in sync.
Then it's possible to stop the application for few seconds leaving repcloud free to perform the swap without risk of Deadlock.

Metamorphoses

Transform on the fly

Repcloud can manipulate the data or set different storage parameters on the new table.

- Fillfactor
- Cleanup the json/jsonb fields from the null keys
- Remove keys from the jsonb fields

Recognition

Repcloud will not be possible without the sponsoring of Cleo LTD.
All of my gratitude for trusting me to build this tool and to give me the permission to release it as an Open Source project.
https://www.meetcleo.com/
Kudos!

Recognition

The dependency resolution is greatly derived from the amazing work of the pgadmin team.
https://www.pgadmin.org/
Kudos!

Recognition

The replay strategy with the transaction check before starting the copy is inspired by the pg_repack code.
http://reorg.github.io/pg_repack/
Kudos!

repcloud

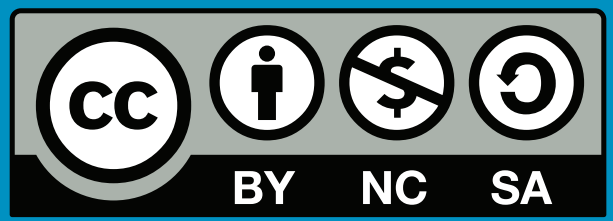
Github project: https://github.com/the4thdoctor/repcloud
Pypi: https://pypi.org/project/repcloud/

repcloud

Repcloud is released under the PostgreSQL license
Currently in alpha release, needs testing, please break it!

Contacts and license

- **Blog:** https://pgdba.org
- **Twitter:** @4thdoctor_scarf
- **Github:** https://github.com/the4thdoctor
- **Linkedin:** https://www.linkedin.com/in/federicocampoli/



RTFM

Next Talk

Please join me Sunday 02 February

- **PostgreSQL devroom**: Room H.2214
- **Start**: 16:00
- **End**: 16:50

I will tell some horror stories caused by people (including myself) that didn't RTFM

That's all folks!

Thank you for listening!

Copyright by dan232323 http://dan232323.deviantart.com/art/Pinkie-Pie-Thats-All-Folks-454693000

Repcloud

A repacker for PostgreSQL in cloud

Federico Campoli

FOSDEM 01 February 2020