Всё, что вы хотели узнать про автовакуум в PostgreSQL

Ilya Kosmodemiansky
ik@postgresql-consulting.com

PGDAY'15 RUSSIA

PostgreSQL-Consulting.com

Outline

- What is it and why is it so important?
- Aggressiveness of autovacuum
- What else important can autovacuum daemon do
- Autovacuum and replication
- How to remove bloat

PGDAY'15 RUSSIA

Two most common problems we meet in our practice

- autovacuum = off
- Autovacuum settings are default

PGDAY'15 RUSSIA

Two most common problems we meet in our practice

- autovacuum = off
- Autovacuum settings are default
- That means there is a lot we can do about improving performance of this particular database

What is autovacuum?

Modern (classical) databases must deal with two fundamental problems:

- **Concurrent operations**
  For that they can transactions, ACID transactions
- **Failures**
  For that they can recover to the last successful transaction using WAL

What is autovacuum?

Technically that means

- There is a combination of locking and MVCC algorithms that provides transactions support
- **Undo** and **Redo** information is stored somewhere to make recovery possible

What is autovacuum?

In PostgreSQL

- Redo - in WAL
- Undo - directly in datafiles
- UPDATE = INSERT + DELETE
- DELETE is just marking tuple as invisible

xmin

```
tt=# INSERT into test(id) values(5);
INSERT 0 1
tt=# select *,xmin,xmax from test;
 id | xmin | xmax
----+------+------
  5 | 1266 |    0
(5 rows)

tt=# select txid_current();
 txid_current
--------------
         1267
(1 row)
```

xmax

```
tt=# begin;
BEGIN
tt=# UPDATE test set id=5 where  id=4;
UPDATE 1

In another session:

tt=# select *,xmin,xmax from test;
 id | xmin | xmax
----+------+------
  4 | 1264 | 1270
(3 rows)
```

Some garbage collection is required

Tuples that are not visible to any running transaction should be removed

- Otherwise fragmentation increases and you run into bloat aka Big Data
- autovacuum workers do that, table by table
- Old-fashioned VACUUM is a bad choice

Beside that, autovacuum workers

- Collect statistics for the optimizer
- Perform wraparound for txid

Some garbage collection is required

Tuples that are not visible to any running transaction should be removed

- Otherwise fragmentation increases and you run into bloat aka Big Data
- autovacuum workers do that, table by table
- Old-fashioned VACUUM is a bad choice

Beside that, autovacuum workers

- Collect statistics for the optimizer
- Perform wraparound for txid

You do not want to turn autovacuum off!

PGDAY'15 RUSSIA

This sort of work **must** be finally done

- If your autovacuum process runs for hours and interferes with some DDL, to simply terminate it is not an option
- Especially for OLTP, autovacuum should be configured **aggressively enough**: so it can work with small portions of data quickly

PGDAY'15 RUSSIA

autovacuum: aggressive enough

```
postgres=# select name, setting, context  from pg_settings
where category ~ 'Autovacuum';

                name                 |  setting  |  context
-------------------------------------+-----------+------------
 autovacuum                          | on        | sighup
 autovacuum_analyze_scale_factor     | 0.05      | sighup
 autovacuum_analyze_threshold        | 50        | sighup
 autovacuum_freeze_max_age           | 200000000 | postmaster
 autovacuum_max_workers              | 10        | postmaster
 autovacuum_multixact_freeze_max_age | 400000000 | postmaster
 autovacuum_naptime                  | 60        | sighup
 autovacuum_vacuum_cost_delay        | 20        | sighup
 autovacuum_vacuum_cost_limit        | -1        | sighup
 autovacuum_vacuum_scale_factor      | 0.01      | sighup
 autovacuum_vacuum_threshold         | 50        | sighup
(11 rows)
```

Sometimes a good idea

in crontab:

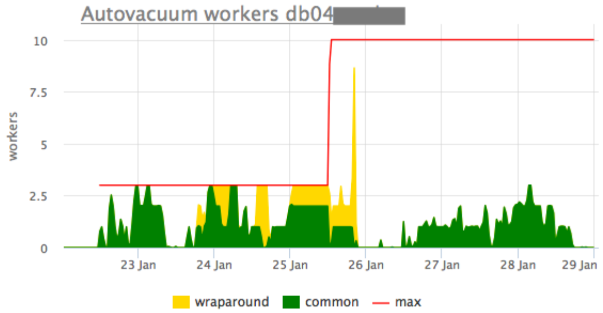
```
* * * * *  /usr/bin/pgrep -f 'postgres: autovacuum' | xargs --no-run-if-empty -I $ renice -n 20 -p $ >/dev/null 2>/dev/null
* * * * *  /usr/bin/pgrep -f 'postgres: autovacuum' | xargs --no-run-if-empty -I $ ionice -c 3 -t -p $
```

in postgresql.conf:

autovacuum_max_workers → 10-20 and autovacuum_vacuum_cost_delay → 10

PGDAY'15 RUSSIA

As a result



PGDAY'15 RUSSIA

ERROR: canceling statement due to conflict with recovery

- The tuple, cleaned up by autovacuum on master, is still in use by some query on hot standby
- hot_standby_feedback = on - The safest way, in spite of some bloat on master

Before you hurry to reconfigure your PostgreSQL

- autovacuum does not remove existing bloat
- dump/restore can be an option, but...
- http://reorg.github.io/pg_repack/
- https://github.com/PostgreSQL-Consulting/pgcompacttable

PGDAY'15 RUSSIA

Questions?

ik@postgresql-consulting.com