МУЛЬТИТРАНЗАКЦИИ И ВОЗМОЖНЫЕ ПРОБЛЕМЫ С НИМИ

Маслов Михаил
Банк ВТБ (ПАО)
Администратор БД PostgreSQL

ПЛАН

ВТБ

Типы транзакций

Идентификаторы транзакций

Мониторинг транзакций и мультитранзакций

Некоторые проблемы с мультитранзакциями

ТИПЫ ТРАНЗАКЦИЙ

Виртуальная транзакция – транзакция в которой не происходит изменения данных (select).

Транзакция – набор команд изменяющих данные, которые выполняются внутри блока begin;….commit; (rollback;) с гарантированным применением изменений или их откатом.

Вложенная транзакция (подтранзакция) – транзакция запущенная внутри другой транзакции (savepoint; блок с exception в PL/pgSQL).

Мультитранзакции – группа транзакций с разделяемым режимом блокировки строк (select … for share; select … for key share;).

ИДЕНТИФИКАТОРЫ ТРАНЗАКЦИЙ И МУЛЬТИТРАНЗАКЦИЙ

ВТБ

Виртуальная транзакция – идентификатор процесса плюс последовательное число:

3/66205, 4/4519, 4/4521

```
=# begin;
BEGIN
=*# select count(*) from pg_class;
 count
-------
   410
(1 row)

=# select relation::regclass as table, locktype, virtualxid, virtualtransaction, pid, mode from pg_locks;
               table                |  locktype  | virtualxid | virtualtransaction |  pid  |      mode
------------------------------------+------------+------------+--------------------+-------+-----------------
 pg_locks                           | relation   |            | 4/4525             | 60881 | AccessShareLock
                                    | virtualxid | 4/4525     | 4/4525             | 60881 | ExclusiveLock
 pg_class_tblspc_relfilenode_index  | relation   |            | 3/66205            | 59385 | AccessShareLock
 pg_class_relname_nsp_index         | relation   |            | 3/66205            | 59385 | AccessShareLock
 pg_class_oid_index                 | relation   |            | 3/66205            | 59385 | AccessShareLock
 pg_class                           | relation   |            | 3/66205            | 59385 | AccessShareLock
                                    | virtualxid | 3/66205    | 3/66205            | 59385 | ExclusiveLock
```

ИДЕНТИФИКАТОРЫ ТРАНЗАКЦИЙ И МУЛЬТИТРАНЗАКЦИЙ

ВТБ

Транзакции (подтранзакции) и мультранзакции используют свои отдельные 32-битные (2^32) идентификаторы.

```
=# create table test (id int);
CREATE TABLE
=# insert into test values (1);
INSERT 0 1
=# begin;
BEGIN
=*# SELECT txid_current_if_assigned();
 txid_current_if_assigned
--------------------------

(1 row)

=*# insert into test values (2);
INSERT 0 1
=*# SELECT txid_current_if_assigned();
 txid_current_if_assigned
--------------------------
                      763
(1 row)
```

```
=*# savepoint a;
SAVEPOINT
=*# insert into test values (3);
INSERT 0 1
=*# SELECT txid_current_if_assigned();
 txid_current_if_assigned
--------------------------
                      763
(1 row)

=*# commit;
COMMIT
```

ИДЕНТИФИКАТОРЫ ТРАНЗАКЦИЙ И МУЛЬТИТРАНЗАКЦИЙ

ВТБ

ctid - Физическое расположение данной версии строки в таблице.
xmin - Идентификатор транзакции, добавившей строку этой версии.
xmax - Идентификатор транзакции, удалившей строку.
backend_xid - Идентификатор верхнего уровня транзакции этого серверного процесса или любой другой.
backend_xmin - текущая граница xmin для серверного процесса.

```
=# select relation::regclass as table, locktype, transactionid, virtualxid, virtualtransaction, pid, mode
from pg_locks;
  table   |   locktype    | transactionid | virtualxid | virtualtransaction |  pid  |       mode
----------+---------------+---------------+------------+--------------------+-------+-----------------
 pg_locks | relation      |               |            | 4/4557             | 60881 | AccessShareLock
          | virtualxid    |               | 4/4557     | 4/4557             | 60881 | ExclusiveLock
 test     | relation      |               |            | 3/66233            | 59385 | RowExclusiveLock
          | virtualxid    |               | 3/66233    | 3/66233            | 59385 | ExclusiveLock
          | transactionid |           763 |            | 3/66233            | 59385 | ExclusiveLock
          | transactionid |           764 |            | 3/66233            | 59385 | ExclusiveLock
(6 rows)
```

ИДЕНТИФИКАТОРЫ ТРАНЗАКЦИЙ И МУЛЬТИТРАНЗАКЦИЙ

```
=# select xact_start, backend_xid, backend_xmin from pg_stat_activity where backend_type =
'client backend';
          xact_start           | backend_xid | backend_xmin
-------------------------------+-------------+--------------
 2023-03-22 06:38:56.602477+03 |         763 |
 2023-03-22 06:42:12.640731+03 |             |          763
```

```
=# SELECT '(0,'||lp||')' AS ctid,
       t_xmin as xmin,
       t_xmax as xmax,
       CASE WHEN (t_infomask & 256) > 0  THEN 't' END AS xmin_comm,
       CASE WHEN (t_infomask & 512) > 0  THEN 't' END AS xmin_abr,
       CASE WHEN (t_infomask & 1024) > 0  THEN 't' END AS xmax_comm,
       CASE WHEN (t_infomask & 2048) > 0  THEN 't' END AS xmax_abr,
       CASE WHEN (t_infomask & 128) > 0   THEN 't' END AS lock_only,
       CASE WHEN (t_infomask & 4096) > 0  THEN 't' END AS is_multi,
       CASE WHEN (t_infomask2 & 8192) > 0 THEN 't' END AS keys_upd,
       CASE WHEN (t_infomask2 & 16384) > 0 THEN 't' END AS hot_upd,
       CASE WHEN (t_infomask & 16) > 0 THEN 't' END AS keyshr_lock,
       CASE WHEN (t_infomask & 16+64) = 16+64 THEN 't' END AS shr_lock
FROM heap_page_items(get_raw_page('test',0))
ORDER BY lp;
 ctid  | xmin | xmax | xmin_comm | xmin_abr | xmax_comm | xmax_abr | lock_only | is_multi | keys_upd | hot_upd | keyshr_lock | shr_lock
-------+------+------+-----------+----------+-----------+----------+-----------+----------+----------+---------+-------------+---------
 (0,1) |  762 |    0 |           |          |           | t        |           |          |          |         |             |
 (0,2) |  763 |    0 |           |          |           | t        |           |          |          |         |             |
 (0,3) |  764 |    0 |           |          |           | t        |           |          |          |         |             |
(3 rows)
```

ИДЕНТИФИКАТОРЫ ТРАНЗАКЦИЙ И МУЛЬТИТРАНЗАКЦИЙ

```
=# begin;
BEGIN
postgres=*# select * from test where id = 1 for share;
 id
----
  1
(1 row)

=*# select * from test where id = 2 for key share;
 id
----
  2
(1 row)

=*# commit;
COMMIT
```

```
=# begin;
BEGIN
postgres=*# select * from test where id = 2 for key share;
 id
----
  2
```

ИДЕНТИФИКАТОРЫ ТРАНЗАКЦИЙ И МУЛЬТИТРАНЗАКЦИЙ

```
=*# select relation::regclass as table, locktype, transactionid, virtualxid, virtualtransaction, pid, mode from
pg_locks;
  table   |   locktype    | transactionid | virtualxid | virtualtransaction |  pid  |      mode
----------+---------------+---------------+------------+--------------------+-------+-----------------
 pg_locks | relation      |               |            | 4/4565             | 60881 | AccessShareLock
 test     | relation      |               |            | 4/4565             | 60881 | RowShareLock
          | virtualxid    |               | 4/4565     | 4/4565             | 60881 | ExclusiveLock
 test     | relation      |               |            | 3/66234            | 59385 | RowShareLock
          | virtualxid    |               | 3/66234    | 3/66234            | 59385 | ExclusiveLock
          | transactionid |           765 |            | 3/66234            | 59385 | ExclusiveLock
          | transactionid |           766 |            | 4/4565             | 60881 | ExclusiveLock
```

```
 ctid  | xmin | xmax | xmin_comm | xmin_abr | xmax_comm | xmax_abr | lock_only | is_multi | keys_upd | hot_upd | keyshr_lock | shr_lock
-------+------+------+-----------+----------+-----------+----------+-----------+----------+----------+---------+-------------+----------
 (0,1) |  762 |  765 | t         |          |           |          | t         |          |          |         | t           | t
 (0,2) |  763 |    2 | t         |          |           |          | t         | t        |          |         | t           |
 (0,3) |  764 |    0 | t         |          |           | t        |           |          |          |         |             |
(3 rows)
```

```
 =*# select * from pg_get_multixact_members('2');
  xid | mode
 -----+-------
  765 | keysh
  766 | keysh
 (2 rows)
```

ИДЕНТИФИКАТОРЫ ТРАНЗАКЦИЙ И МУЛЬТИТРАНЗАКЦИЙ

```
=*# select * from test where id = 1 for share;
 id
----
  1
(1 row)
```

```
 ctid  | xmin | xmax | xmin_comm | xmin_abr | xmax_comm | xmax_abr | lock_only | is_multi | keys_upd | hot_upd | keyshr_lock | shr_lock
-------+------+------+-----------+----------+-----------+----------+-----------+----------+----------+---------+-------------+----------
 (0,1) |  762 |    3 | t         |          |           |          | t         | t        |          |         | t           | t
 (0,2) |  763 |    2 | t         |          |           |          | t         | t        |          |         | t           |
 (0,3) |  764 |    0 | t         |          |           | t        |           |          |          |         |             |
(3 rows)
```

```
=*# select * from pg_get_multixact_members('3');
 xid | mode
-----+------
 765 | sh
 766 | sh
(2 rows)
```

ИДЕНТИФИКАТОРЫ ТРАНЗАКЦИЙ И МУЛЬТИТРАНЗАКЦИЙ

```
=# begin;
BEGIN
=*# select * from test where id = 1 for share;
 id
----
  1
```

```
=# begin;
BEGIN
=*#  select * from test where id = 1 for share;
 id
----
  1
```

```
ctid  | xmin | xmax | xmin_comm | xmin_abr | xmax_comm | xmax_abr | lock_only | is_multi | keys_upd | hot_upd | keyshr_lock | shr_lock
------+------+------+-----------+----------+-----------+----------+-----------+----------+----------+---------+-------------+----------
 (0,1) |  762 |    7 | t         | t        |           |          | t         | t        |          |         | t           | t
 (0,2) |  763 |    0 | t         | t        |           | t        |           |          |          |         |             |
 (0,3) |  764 |    0 | t         | t        |           | t        |           |          |          |         |             |
```

```
=*# commit;
COMMIT
```

```
=# vacuum freeze test;
```

```
 ctid  | xmin | xmax | xmin_comm | xmin_abr | xmax_comm | xmax_abr | lock_only | is_multi | keys_upd | hot_upd | keyshr_lock | shr_lock
-------+------+------+-----------+----------+-----------+----------+-----------+----------+----------+---------+-------------+----------
 (0,1) |  762 |    8 | t         | t        |           |          | t         | t        |          |         | t           | t
 (0,2) |  763 |    0 | t         | t        |           | t        |           |          |          |         |             |
 (0,3) |  764 |    0 | t         | t        |           | t        |           |          |          |         |             |
```

```
=*# select * from pg_get_multixact_members('8');
 xid | mode
-----+------
 773 | sh
```

ИДЕНТИФИКАТОРЫ ТРАНЗАКЦИЙ И МУЛЬТИТРАНЗАКЦИЙ

select … for key share для внешних ключей

```
=# CREATE TABLE parent(
   p_id integer PRIMARY KEY,
   p_val text
);
CREATE TABLE
=# CREATE TABLE child(
   c_id integer PRIMARY KEY,
   p_id integer REFERENCES parent(p_id),
   c_val text
);
CREATE TABLE
=# INSERT INTO parent (p_id, p_val)
   VALUES (1, 'a');
INSERT 0 1
=# begin;
BEGIN
=*# INSERT INTO child (c_id, p_id, c_val) VALUES (1, 1, 'a');
INSERT 0 1
```

```
=# begin;
BEGIN
=*# INSERT INTO child (c_id, p_id, c_val) VALUES (2, 1, 'b');
INSERT 0 1
```

ИДЕНТИФИКАТОРЫ ТРАНЗАКЦИЙ И МУЛЬТИТРАНЗАКЦИЙ

```
 ctid  | xmin | xmax | xmin_comm | xmin_abr | xmax_comm | xmax_abr | lock_only | is_multi | keys_upd | hot_upd | keyshr_lock | shr_lock
-------+------+------+-----------+----------+-----------+----------+-----------+----------+----------+---------+-------------+----------
 (0,1) |  785 |   12 | t         |          |           |          | t         | t        |          |         | t           |
```

```
=# select * from pg_get_multixact_members('12');
 xid | mode
-----+-------
 786 | keysh
 787 | keysh
```

МОНИТОРИНГ ТРАНЗАКЦИЙ И МУЛЬТИТРАНЗАКЦИЙ

ВТБ

```
=# SELECT datname
    , age(datfrozenxid)
    , mxid_age(datminmxid)
    , current_setting('autovacuum_freeze_max_age')
    , current_setting('autovacuum_multixact_freeze_max_age')
FROM pg_database
ORDER BY 2 DESC;
```

datname	age	mxid_age	current_setting	current_setting
postgres	58	8	200000000	400000000
template1	58	8	200000000	400000000
template0	58	8	200000000	400000000

```
=# select c.oid::regclass as table, c.relfrozenxid, age(c.relfrozenxid), current_setting('autovacuum_freeze_max_age') as
av_xid_freeze, c.relminmxid, mxid_age(c.relminmxid), current_setting('autovacuum_multixact_freeze_max_age') as
av_mxid_freeze, pg_size_pretty(pg_total_relation_size(c.oid))
   from pg_class c
   join pg_namespace n on c.relnamespace = n.oid
   where relkind in ('r','t','m')
   order by 3 asc limit 50;
```

table	relfrozenxid	age	av_xid_freeze	relminmxid	mxid_age	av_mxid_freeze	pg_size_pretty
test	773	1	200000000	7	2	400000000	40 kB
pg_toast.pg_toast_1262	722	52	200000000	1	8	400000000	8192 bytes
pg_database	719	55	200000000	1	8	400000000	80 kB
pg_type	716	58	200000000	1	8	400000000	232 kB

МОНИТОРИНГ ТРАНЗАКЦИЙ И МУЛЬТИТРАНЗАКЦИЙ

```
=# select next_xid, next_multixact_id, oldest_xid, oldest_xid_dbid, oldest_multi_xid, oldest_multi_dbid
from pg_control_checkpoint();
 next_xid | next_multixact_id | oldest_xid | oldest_xid_dbid | oldest_multi_xid | oldest_multi_dbid
----------+-------------------+------------+-----------------+------------------+------------------
 0:774    |                 9 |        716 |               1 |                1 |                1
```

```
/usr/pgsql-15/bin/pg_controldata -D $PGDATA
pg_control version number:            1300
Catalog version number:               202209061
Database system identifier:           7208618065697320740
Database cluster state:               in production
pg_control last modified:             Wed 22 Mar 2023 07:17:15 AM MSK
Latest checkpoint location:           0/1671998
Latest checkpoint's REDO location:    0/1671960
Latest checkpoint's REDO WAL file:    000000010000000000000001
Latest checkpoint's TimeLineID:       1
Latest checkpoint's PrevTimeLineID:   1
Latest checkpoint's full_page_writes: on
Latest checkpoint's NextXID:          0:774
Latest checkpoint's NextOID:          24576
Latest checkpoint's NextMultiXactId:  9
Latest checkpoint's NextMultiOffset:  15
Latest checkpoint's oldestXID:        716
Latest checkpoint's oldestXID's DB:   1
Latest checkpoint's oldestActiveXID:  774
Latest checkpoint's oldestMultiXid:   1
Latest checkpoint's oldestMulti's DB: 1
Latest checkpoint's oldestCommitTsXid:0
Latest checkpoint's newestCommitTsXid:0
Time of latest checkpoint:            Wed 22 Mar 2023 07:17:15 AM MSK
```

МОНИТОРИНГ ТРАНЗАКЦИЙ И МУЛЬТИТРАНЗАКЦИЙ

```
du -hc /pg_data/data15/pg_multixact/

8.0K    /pg_data/data15/pg_multixact/members - размер 1 файла 262144
8.0K    /pg_data/data15/pg_multixact/offsets - размер 1 файла 262144
16K     /pg_data/data15/pg_multixact/
16K     total


du -hc /pg_data/data15/pg_xact/

8.0K    /pg_data/data15/pg_xact/
8.0K    total



du -hc /pg_data/data15/pg_subtrans/

8.0K    /pg_data/data15/pg_subtrans/
8.0K    total


LWLock: MultiXactGen
LWLock: MultiXactMemberBuffer
LWLock: MultiXactMemberSLRU
LWLock: MultiXactOffsetBuffer
LWLock: MultiXactOffsetSLRU
LWLock: MultiXactTruncation
```

ПЕРЕПОЛНЕНИЕ СЧЕТЧИКА МУЛЬТИТРАНЗАКЦИЙ

```
/usr/pgsql-15/bin/initdb -D data15

/usr/pgsql-15/bin/pg_ctl start -o '-c port=8432' -o '-c max_prepared_transactions=1' -D data15 -l logfile

postgres=# CREATE TABLE test_table (id integer PRIMARY KEY, val text);
CREATE TABLE
postgres=# INSERT INTO test_table VALUES (1, 'a') RETURNING xmin;
 xmin
------
  736
(1 row)

INSERT 0 1
postgres=# DELETE FROM test_table WHERE id = 1 RETURNING xmax;
 xmax
------
  737
(1 row)

DELETE 1
postgres=# INSERT INTO test_table VALUES (2, 'b') RETURNING xmin;
 xmin
------
  738
(1 row)

INSERT 0 1
```

ПЕРЕПОЛНЕНИЕ СЧЕТЧИКА МУЛЬТИТРАНЗАКЦИЙ

```
postgres=# begin;
BEGIN
postgres=*# SELECT pg_current_xact_id();
 pg_current_xact_id
--------------------
                739
(1 row)

postgres=*# select * from test_table where id = 2 for
share;
 id | val
----+-----
  2 | b
(1 row)

postgres=*# commit;
```

```
postgres=# begin;
BEGIN
postgres=*# SELECT pg_current_xact_id();
 pg_current_xact_id
--------------------
                740
(1 row)

postgres=*# select * from test_table where id = 2 for share;
 id | val
----+-----
  2 | b
(1 row)

postgres=*# select ctid, xmin, xmax, id from test_table ;
 ctid  | xmin | xmax | id
-------+------+------+----
 (0,2) |  738 |    1 |  2
(1 row)

postgres=*# commit;
COMMIT
postgres=# BEGIN;
BEGIN
postgres=*# LOCK test_table IN SHARE UPDATE EXCLUSIVE MODE;
LOCK TABLE
postgres=*# PREPARE TRANSACTION 'prept';
PREPARE TRANSACTION
postgres=# \q
```

ПЕРЕПОЛНЕНИЕ СЧЕТЧИКА МУЛЬТИТРАНЗАКЦИЙ

```
/usr/pgsql-15/bin/pg_ctl stop -D data15

/usr/pgsql-15/bin/pg_resetwal -m 2137483648,1 -D data15
Write-ahead log reset

dd if=/dev/zero of=data15/pg_multixact/offsets/7F67 bs=8192
count=15

/usr/pgsql-15/bin/pg_ctl start -o '-c port=8432' -o '-c
max_prepared_transactions=1' -D data15 -l logfile

postgres=# begin;
BEGIN
postgres=*# SELECT pg_current_xact_id();
 pg_current_xact_id
--------------------
                742
(1 row)

postgres=*# select * from test_table where id = 2 for share;
 id | val
----+-----
  2 | b
(1 row)

postgres=*# select ctid, xmin, xmax, id from test_table ;
 ctid  | xmin |    xmax    | id
-------+------+------------+----
 (0,2) |  738 | 2137483648 |  2
(1 row)

postgres=*# commit;
```

```
postgres=# begin;
BEGIN
postgres=*# SELECT pg_current_xact_id();
 pg_current_xact_id
--------------------
                743
(1 row)

postgres=*# select * from test_table where id = 2 for share;
WARNING:  database "postgres" must be vacuumed before
10000000 more MultiXactIds are used
HINT:  Execute a database-wide VACUUM in that database.
You might also need to commit or roll back old prepared
transactions, or drop stale replication slots.
 id | val
----+-----
  2 | b
(1 row)

postgres=*# commit;
COMMIT
```

ПЕРЕПОЛНЕНИЕ СЧЕТЧИКА МУЛЬТИТРАНЗАКЦИЙ

```
/usr/pgsql-15/bin/pg_ctl stop -o -D data15 -l logfile

/usr/pgsql-15/bin/pg_resetwal -m 2146483648,1 -D data15

dd if=/dev/zero of=data15/pg_multixact/offsets/7FF0 bs=262144
count=1

postgres=# begin;
postgres=*# SELECT pg_current_xact_id();
 pg_current_xact_id
--------------------
                744
(1 row)

postgres=*# select * from test_table where id = 2 for share;
 id | val
----+-----
  2 | b
(1 row)

postgres=*# commit;
```

```
SELECT gid, prepared, owner, database, transaction AS xmin
FROM pg_prepared_xacts
ORDER BY age(transaction) DESC;
  gid  |           prepared            |  owner   | database | xmin
-------+-------------------------------+----------+----------+------
 prept | 2023-03-22 14:47:59.745343+03 | postgres | postgres |  741

rollback prepared'prept';
ROLLBACK PREPARED
```

```
postgres=# begin;
BEGIN
postgres=*# SELECT pg_current_xact_id();
 pg_current_xact_id
--------------------
                745
(1 row)

postgres=*# select * from test_table where id = 2 for
share;
ERROR:  database is not accepting commands that generate
new MultiXactIds to avoid wraparound data loss in database
with OID 0
HINT:  Execute a database-wide VACUUM in that database.
You might also need to commit or roll back old prepared
transactions, or drop stale replication slots.
postgres=!# commit;
ROLLBACK
postgres=# vacuum verbose;
INFO:  aggressively vacuuming "postgres.public.test_table“
```

ПЕРЕПОЛНЕНИЕ СЧЕТЧИКА МУЛЬТИТРАНЗАКЦИЙ

ВТБ

```
select ctid, xmin, xmax, id from test_table ;
 ctid  | xmin | xmax | id
-------+------+------+----
 (0,2) |  738 |  744 |  2

 next_xid | next_multixact_id | oldest_xid | oldest_xid_dbid | oldest_multi_xid | oldest_multi_dbid
----------+-------------------+------------+-----------------+------------------+-------------------
 0:746    |        2146483648 |        716 |               1 |                1 |                 0

         table          | relfrozenxid | age | av_xid_freeze | relminmxid | mxid_age | av_mxid_freeze | pg_size_pretty
------------------------+--------------+-----+---------------+------------+----------+----------------+----------------
 pg_type                |          735 |  11 | 200000000     | 2146483648 |        0 | 400000000      | 232 kB
 pg_index               |          735 |  11 | 200000000     | 2146483648 |        0 | 400000000      | 96 kB
 pg_depend              |          735 |  11 | 200000000     | 2146483648 |        0 | 400000000      | 288 kB
 pg_attribute           |          735 |  11 | 200000000     | 2137483648 |  9000000 | 400000000      | 688 kB
 pg_class               |          735 |  11 | 200000000     | 2146483648 |        0 | 400000000      | 232 kB
 pg_constraint          |          735 |  11 | 200000000     | 2146483648 |        0 | 400000000      | 144 kB
 test_table             |          738 |   8 | 200000000     | 2146483648 |        0 | 400000000      | 64 kB
```

ВОПРОСЫ

???

ССЫЛКИ

https://postgrespro.ru/docs/postgresql/15/
https://github.com/postgres/
https://www.cybertec-postgresql.com/en/whats-in-an-xmax/
https://www.cybertec-postgresql.com/en/transaction-id-wraparound-a-walk-on-the-wild-side/
https://www.cybertec-postgresql.com/en/reasons-why-vacuum-wont-remove-dead-rows/
https://habr.com/ru/company/postgrespro/blog/463819/
https://habr.com/ru/company/postgrespro/blog/445820/

Рогов Е.В. PostgreSQL 15 изнутри