

Novinky v PostgreSQL 12

Prague PostgreSQL Developer Day, 6. února 2020

Tomáš Vondra <tomas.vondra@2ndquadrant.com>

Odstraněné věci

- datové typy (nahrazeno SQL-standard typy)
 - abstime
 - reltime
 - tinternal
- podpora pro `--disable-strong-random`
 - vyžadován silný zdroj náhodných dat



2ndQuadrant
PostgreSQL

DBA / Admin

pg_stat_ssl

- nové sloupce
 - client_serial
 - issuer_dn
- dat skrytá pro neprivilegované uživatele

SSL konfigurace

kontrola min/max SSL verze

- `ssl_min_protocol_version=TLSv1`
- `ssl_max_protocol_version=""`

všechny podporované úrovně TLS

- TLSv1, TLSv1.1, TLSv1.2, TLSv1.3

GSSAPI šifrování

- šifrování bez SSL
- bez potřeby certifikátů apod.
- předpokládá že GSSAPI už funguje
- `pg_stat_gssapi`

VACUUM

SKIP_LOCKED

- přeskočí relace které nejde okamžitě zamknout

DISABLE_PAGE_SKIPPING

- obchází visibility map
- určeno pro debuggování

```
vacuumdb --min-xid-age X --min-mxid-age Y
```

COPY FROM WHERE

```
COPY mytable (a,b,c)
FROM '/tmp/myfile.csv'
WITH CSV
WHERE a > 5
```



CSV výstupní formát v psql

```
postgres=# \pset format csv
```

Output format is csv.

```
postgres=# SELECT * FROM mytable;
```

```
a,b,c
```

```
7,8,9
```

pg_stat_statements

resetování statistik pro jednotlivé dotazy

```
SELECT pg_stat_statements_reset(queryid => -6363133595);
```

Progress monitoring

`pg_stat_progress_create_index`

- CREATE INDEX
- REINDEX

`pg_stat_progress_cluster`

- CLUSTER
- VACUUM FULL

REINDEX CONCURRENTLY

- reindexování bez zamezení zápisů
- trochu jako DROP + CREATE CONCURRENTLY
- `reindexdb --concurrently`

Checksums

- offline zapnutí / vypnutí
- progress report for zapínání / kontrolu
- `pg_checksums --progress`

Pluggable access method

- API pro "storage engines"
- existuje jediná implementace (zatím)





2ndQuadrant
PostgreSQL

SQL a vývojáři

WITH OIDS

- Odstraněno!
- označeno jako "deprecated" od 2005
- oid je nyní normální sloupec
- žádná další magie

GENERATED sloupce

- sloupce s výsledkem výpočtu
- aktuálně podporována pouze STORED varianta
- bez triggerů

```
CREATE TABLE foo (  
    a int NOTNULL,  
    b int GENERATED ALWAYS AS (a*2) STORED  
);
```

ENUM

- přidání enum hodnoty v transakci

```
postgres=# BEGIN;  
postgres=# ALTER TYPE etype ADD VALUE 'foo';  
postgres=# ROLLBACK;
```

- nové omezení

```
postgres=# BEGIN;  
postgres=# ALTER TYPE etype ADD VALUE 'foo';  
postgres=# SELECT 'foo'::etype;  
ERROR:  unsafe use of new value "foo" of enum type etype
```

JSONPATH

- SQL standard pro dotazování JSON dokumentů
- nový "query language"
- stále stejné indexování
- nové funkce a operátory
 - `jsonb_path_exists()` @?
 - `jsonb_path_matches()` @@
 - `jsonb_path_query()`
 - ...

JSONPATH

```
SELECT jsonb_path_exists ('{"a": 1}', '$.a');
```

```
SELECT '{"a": 1}'::jsonb @? '$.a';
```

```
SELECT jsonb_path_match ('{"a": 1}', '$.a == 1');
```

```
SELECT '{"a": 1}'::jsonb @@ '$.a == 1';
```

CTE

- ne nutně "optimization barrier" jako dříve
- nové klíčové slovo MATERIALIZE

```
postgres=# WITH t AS (SELECT * FROM foo),  
postgres-#      t2 AS (SELECT * FROM foo)  
postgres-# SELECT * FROM t UNION ALL SELECT * FROM t2;
```

CTE / MATERIALIZED

```
postgres=# EXPLAIN
postgres-# WITH t AS (SELECT * FROM foo),
postgres-#      t2 AS MATERIALIZED (SELECT * FROM foo)
postgres-# SELECT * FROM t UNION ALL SELECT * FROM t2;
```

QUERY PLAN

```
-----
Append  (cost=35.50..147.50 rows=5100 width=4)
  CTE t2
    -> Seq Scan on foo foo_1  (cost=0.00..35.50 rows=...)
    -> Seq Scan on foo  (cost=0.00..35.50 rows=2550 width=4)
    -> CTE Scan on t2  (cost=0.00..51.00 rows=2550 width=4)
(5 rows)
```

CREATE STATISTICS

- nový typ statistik MCV

```
CREATE TABLE t AS
SELECT mod(i,100) AS a, mod(i,100) AS b
FROM generate_series(1,1000000) s(i);

CREATE STATISTICS s (mcv) ON a, b FROM t;

ANALYZE t;

EXPLAIN ANALYZE SELECT * FROM t WHERE (a > 0) AND (b < 2);
```



2ndQuadrant
PostgreSQL

Zálohování & replikace

max_wal_senders

- nadále není zahrnuto v max_connections
- oddělený parametr, dedikovaný wal sender procesům

recovery.conf

- integrováno do postgresql.conf
- v hlavním konfiguračním souboru / include souboru
 - např. postgresql.auto.conf
- rekonfigurace pouze reloadem
 - recovery_min_apply_delay, archive_cleanup_command, ...
- recovery.signal
 - nový "trigger" soubor (protože recovery.conf už neexistuje)
- standby.signal
 - nový "trigger" soubor pro standby mode

recovery.conf

- rekonfigurace pouze reloadem
 - `recovery_min_apply_delay`, `archive_cleanup_command`, ...
- `recovery_target_timeline`
 - nový default "latest"
- `pg_promote()`
 - funkce pro "promote" standby

Exklusivní backupy

ještě více označeny jako "deprecated"





2ndQuadrant
PostgreSQL

Výkon

Indexy

SP-GiST

- nyní podporují KNN vyhledávání

GiST, GIN a SP-GiST

- méně WAL generováno během vytváření indexů
- rychlejší a lepší!

GiST covering indexy (IOS)

- INCLUDE klauzule

TOAST

- částečná dekomprese TOAST hodnot
- pokud stačí pouze část hodnoty (např. PostGIS)

Partitioning

- flexibilnější limity partitions (generalized expressions)
- zamykání odloženo až do exekuce
 - může vést k daleko rychlejším skenům
 - pokud se pracuje s mnoha partitions
- ATTACH nevyžaduje access exclusive lock
- Multi-inserty pro COPY
 - rychlejší COPY do partitioned tabulek
- Cizí klíče ukazující na partitioned tabulku
 - pokud partitioned tabulka má PRIMARY KEY

SERIALIZABLE

- nyní funguje s paralelními dotazy

JIT kompilace

- nyní zapnuto by default



2ndQuadrant
PostgreSQL

... a spousta dalšího



2ndQuadrant
PostgreSQL

<https://www.postgresql.org/docs/12/release-12.html>