Novinky v PotgreSQL 16

Tomáš Vondra, EDB

tomas.vondra@enterprisedb.com

OpenAlt 2023

https://blog.pgaddict.com / tv@fuzzy.cz

https://www.postgresql.org/docs/release/16.0/

EDB

- DBA a administrace
- SQL a vývoj
- backup a replikace
- výkon

EDB

Nekompatibility

- Windows - minimální verze 10
- odstraněné porty: HPUX, Itanium
- odstranění promote_trigger_file
  - používejte pg_ctl_promote (ze shellu)
  - nebo pg_promote() (v SQL)
- odstranění vacuum_defer_cleanup_age

EDB

DBA / administrace

Novinky / psql

- možnost použít "extended" protokol

```
postgres=# SELECT count(*) FROM pg_class WHERE relname = $1;
postgres=# \bind 'jmeno_tabulky'
postgres=# \g
```

- skriptování v psql
- testování chování s extended protokolem

EDB

Novinky / psql

- výsledek / exit status shell skriptů
- SHELL_ERROR
  - true = error, false = OK
  - \!, \g, \o, \w, a \copy meta-příkazy, a (`) expanzi
- SHELL_EXIT_CODE
  - exit status posledního shell příkazu
  - 0–127 program exit, 128–255 signal, -1 chyba při spuštění/ukončení

EDB

Novinky

`SYSTEM_USER`

- autentizační metoda : přihlášený uživatel
- alternativa k `SESSION_USER`, `CURRENT_USER`

`pg_hba / pg_ident`

- matchování uživatelů / databází pomocí regulárních výrazů
- include souborů (`include`, `include_if_exists`, `include_dir`)
- zobrazeno v `pg_hba_file_rules` a `pg_ident_file_mappings`

EDB

libpq

- `require_auth`
  - povolené / zakázané autentizační metody (z klienta)
  - `require_auth=scram-sha-256`
  - `require_auth=!password`
- `sslcertmode = [allow, disable, require]`
  - může/musí si klient vyžádat certifikát od klienta?
  - `sslcertmode=require`
- sslrootcert=system
  - použití systémového CA store, vynucuje `verify-full`
    (triviální získat "systémový" certifikát, slabší verifikace je k ničemu)

EDB

Libpq load balancing

- speficikace více potenciálních hostů

```
postgresql://host1:123,host2:456/somedb
host=host1,host2 port=123,456
```

- dosud připojení jeden po druhém
  - `target_session_attrs (any, read-write, read-only, ...)`
  - implicitní preference pro servery na začátku seznamu
- `load_balance_hosts=random`
  - výběr serveru je náhodný / rovnoměrný

EDB

Kerberos credential delegation

- povoluje delegaci informací z klienta na server
- server se pak může připojovat jinam jako uživatel
  - postgres_fdw / dblink / ...
- povolit na serveru

  `gss_accept_delegation`
- povolit na klientu (libpq)

  `gssdelegation=1`

EDB

VACUUM

- instrumentace

```
frozen: 47 pages from table (35.34% of total) had 10000 tuples frozen
```

- aktualizace "cost" parameterů během autovacuum
  - původně se načítalo pouze mezi tabulkami
- page level freezing
  - provádí "freeze" datových stránek daleko dříve
  - efektivnější než freeze jednotlivých řádek
  - also triggers aer FPI by heap pruning

EDB

VACUUM / ANALYZE / vacuumdb

BUFFER_USAGE_LIMIT

- ring buffer size
- kontrola využití DB cache (shared buffers)
- má vliv pro velké tabulky apod.
- `vacuumdb --buffer-usage-limit`
- `vacuum_buffer_usage_limit` (`ANALYZE`)

EDB

Statistiky

- čas posledního sekvenčního / index skenu relace
  - užitečné pro sledování zda se tabulka používá
- pg_stat_io
  - detailnější I/O statistiky (čtení, zápisy, růst souboru, evikce, fsync, ...)
  - globální pohled, ale kontext (typ procesu/objektu, ...)
  - rozšíření toho co už jsme měli v pg_stat_bgwriter

SQL / developer

EDB

ANY_VALUE

- agegační funkce definovaná v SQL standardu
- "libovolná hodnota ve skupině"

```
SELECT a, ANY_VALUE(b) FROM tabulka GROUP BY a;
```


EDB

COPY DEFAULT

možnost specifikovat výchozích hodnot v COPY

EDB

Nastavení STORAGE

- specifikace jak je uložený daný sloupec
  - přímo v tabulce / externě
  - komprese / bez komprese
- dříve vyžadovalo `CREATE TABLE + ALTER TABLE`

```
CREATE TABLE t (
    sloupec TEXT STORAGE MAIN
);
```


EDB

INTEGER / NUMERIC

- možnost zadávat integery v jiných bázích
  - hexadecimal, octal a binární
  - `0x42F, 0o273, 0b100101`
- podržítka v integer/numeric literálech
  - `1_000_000_000`

EDB

JSON / SQL standard

JSON konstruktory

```
JSON_ARRAY

JSON_ARRAYAGG

JSON_OBJECT

JSON_OBJECTAGG
```

JSON predikáty

```
IS JSON

IS JSON ARRAY

IS JSON OBJECT

IS JSON SCALAR
```


EDB

backup a replikace

EDB

pg_dump komprese

- nová syntaxe + nové kompresní algoritmy

  `-Z gzip:9`

- LZ4 podpora

  `-Z lz4:9`

- ZSTD podpora

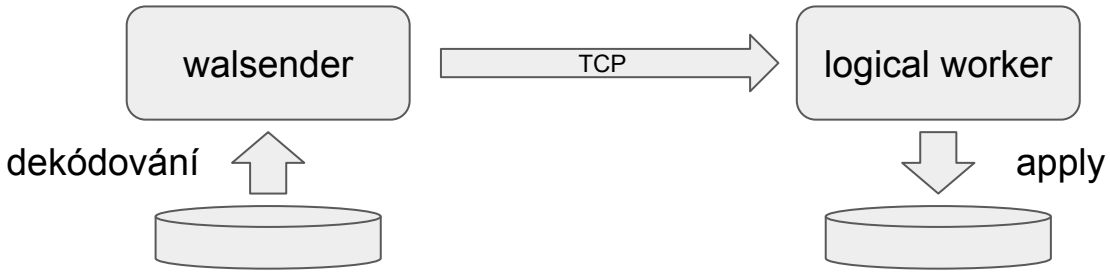
  `-Z zstd:9`

EDB

Logická replikace

parallel apply

- paralelní "worker" procesy na subscriberu
- použije více procesů pro jednu "subscripci"
- povoluje se pro jednotlivé subscripce

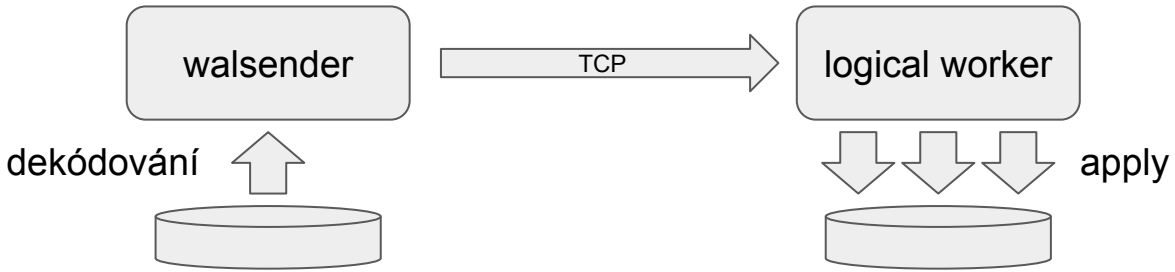


EDB

Logická replikace

parallel apply

- paralelní "worker" procesy na subscriberu
- použije více procesů pro jednu "subscripci"
- povoluje se pro jednotlivé subscripce



EDB

Logická replikace

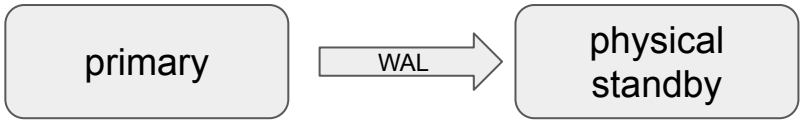
bezpečnost

- předdefinovaná role: `pg_create_subscription`
- default: změny se provádí jako "table owner"
- volitelně: `run_as_owner=true` jako "subscription owner"

EDB

Logická replikace ze standby

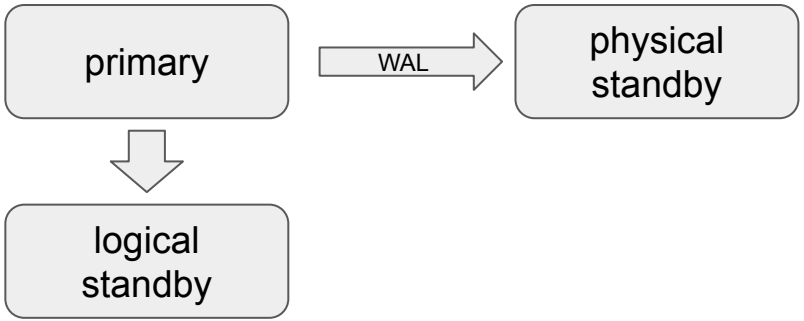
- z fyzické standby
- logické dekódování na standby
- publikaci musíte vytvořit na "primary"
- ale logickou repliku připojíte ke standby



EDB

Logická replikace ze standby

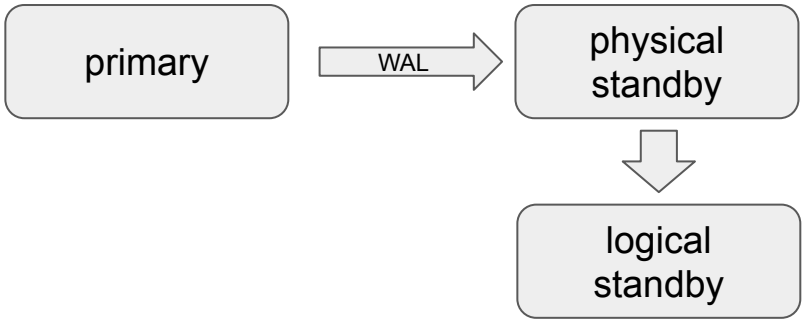
- z fyzické standby
- logické dekódování na standby
- publikaci musíte vytvořit na "primary"
- ale logickou repliku připojíte ke standby



EDB

Logická replikace ze standby

- z fyzické standby
- logické dekódování na standby
- publikaci musíte vytvořit na "primary"
- ale logickou repliku připojíte ke standby



EDB

výkon

Spousta malých vylepšení

je vidět jenom nepřímo

- prostě běží rychleji
- např. optimalizace memory managementu

explicitní podpora SIMD

- Intel SSE2
- ARM NEON

(opět) rychlejší "relation extension"

Paralelizace

paralelní agregační funkce

- string_agg()
- array_agg()

paralelní hash FULL JOIN

- dříve byl paralelizovaný pouze LEFT JOIN
- nyní funguje i FULL a RIGHT JOIN

EDB

DISTINCT

- incremental sort pro DISTINCT
- LIMIT pokud jsou "distinct keys" redundantní

```
SELECT DISTINCT col,col2 FROM tab WHERE col = 1 AND col2 = 10;
```

- chytřejší při využívání třídění pro agregaci
  - např. může používat index
- odstraňuje redundantní grouping
  - např. výrazy které víme že jsou identické

EDB

Window funkce

- některé funkce nerozlišují mezi
  - ROWS UNBOUNDED PRECEDING AND CURRENT ROW
  - RANGE UNBOUNDED PRECEDING AND CURRENT ROW
- ROWS je rychlejší než RANGE
- automaticky "interně" přepne
- `row_number(), rank(), dense_rank()`
  `percent_rank(), cume_dist(), ntile()`
- další optimalizace ...

EDB

Hromada dalších ...

hromada malých fixů

výkonnostních vylepšení

infrastrukturních změn

... atd. atd.

https://www.postgresql.org/docs/release/16.0/

EDB

pgconf.eu 2023

- prosinec 12-15
- Praha, Clarion Congress Hotel
- https://pgconf.eu
- ~50 přednášek (+workshopy + keynotes ...)

EDB

Q&A