



# Novinky v PostgreSQL 10 (a 11)

Tomáš Vondra  
<[tomas.vondra@2ndquadrant.com](mailto:tomas.vondra@2ndquadrant.com)>

Prague PostgreSQL Developer Day 2018

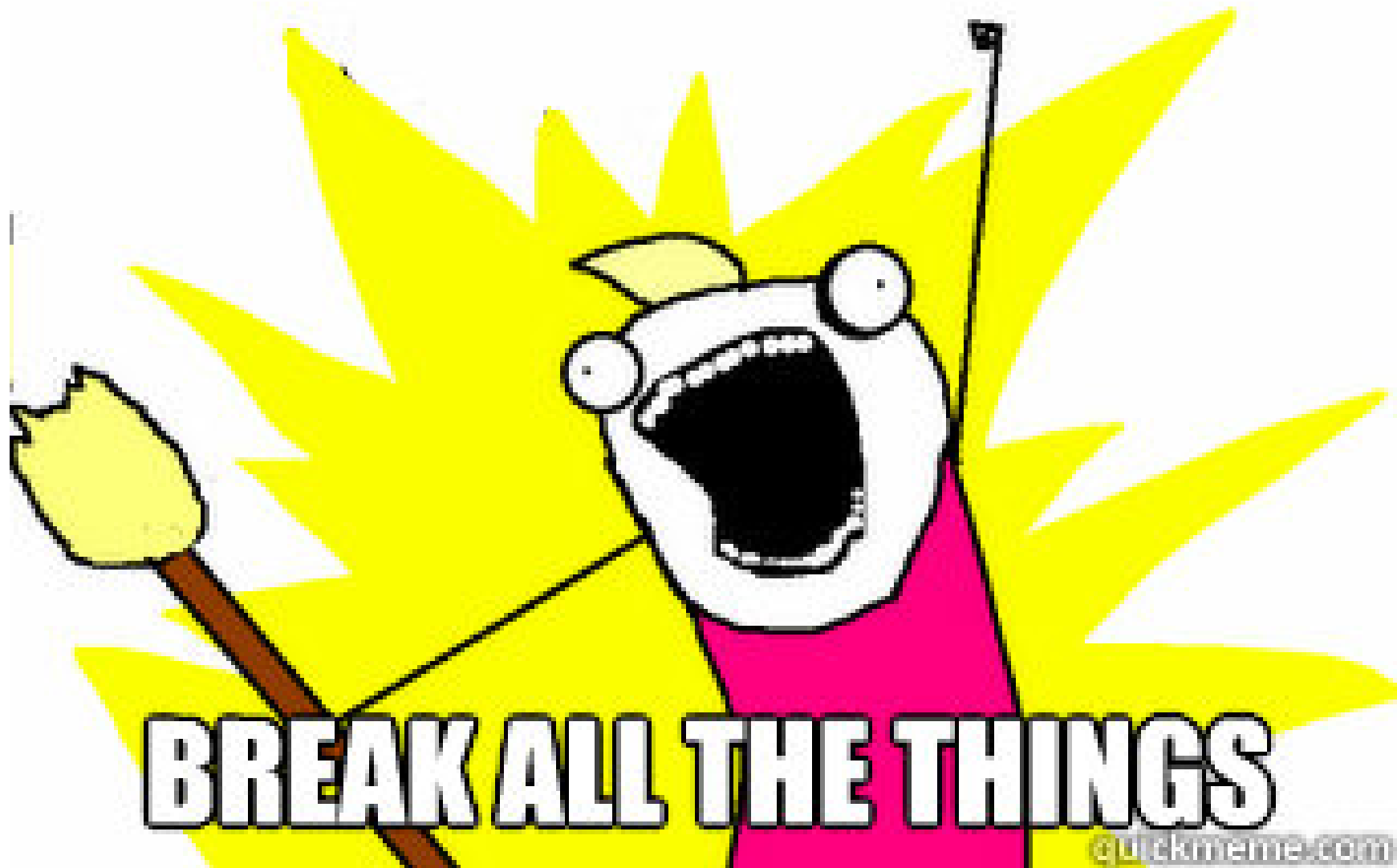


### **E.3.1. Overview**

Major enhancements in PostgreSQL 10 include:

- Logical replication using publish/subscribe
- Declarative table partitioning
- Improved query parallelism
- Significant general performance improvements
- Stronger password authentication based on SCRAM-SHA-256
- Improved monitoring and control

The above items are explained in more detail in the sections below.



**BREAK ALL THE THINGS**



Změna systému verzování



...  
9.0  
  9.0.1  
  9.0.2  
9.1  
  
...  
9.5  
9.6  
  9.6.1



...  
9.0  
  9.0.1  
  9.0.2  
9.1  
...  
9.5  
9.6  
  9.6.1  
**10**  
  **10.1**



odstranění protokolu 1.0  
(žádní klienti pro < 6.3)



odstranění “floating point timestamps”





## přejmenování

```
pg_xlog => pg_wal  
pg_clog => pg_xact
```



## přejmenování

pg\_xlog => pg\_wal  
pg\_clog => pg\_xact

pg\_switch\_xlog() => pg\_switch\_wal()  
pg\_xlogfile\_name() => pg\_walfile\_name()  
pg\_current\_xlog\*() => pg\_current\_wal\*  
pg\_last\_xlog\*() => pg\_last\_wal\*  
pg\_xlog\_location\_diff() => pg\_wal\_location\_diff()

pg\_receivexlog => pg\_receivewal  
pg\_resetxlog => pg\_resetwal  
pg\_xlogdump => pg\_waldump

pg\_basebackup --xlog-method => --wal-method  
pg\_basebackup --xlogdir => --waldir



# DBA a administrace



# SCRAM

- Salted Challenge Response Authentication
- Standardizovaný způsob autentizace
- Bezpečnější než md5
- Přejděte až se podpora dostane do klientů



# libpq

- Lze specifikovat více hostů

```
host=pg1,pg2,pg3 user=bob password=xxx
```

- Lze si vyžádat hosta umožňujícího zápisy

```
host=pg1,pg2,pg3 target_session_attrs=read-write
```

- Specifikace .pgpass souboru

```
host=pg1 passfile=/nejaky/adresar/mypgpass
```



# Monitoring



```
log_line_prefix = '%m [%p]'
```



# pg\_stat\_activity

- nyní jsou vidět všechny typy backendů
  - background worker
  - walsender procesy
  - ...

| datid | pid   | backend_type        | application_name             |
|-------|-------|---------------------|------------------------------|
| 12295 | 18301 | client backend      | psql                         |
|       | 17295 | autovacuum launcher |                              |
|       | 17297 | background worker   | logical replication launcher |
|       | 17293 | background writer   |                              |
|       | 17292 | checkpointer        |                              |
|       | 18615 | walsender           | pg_receivewal                |
|       | 17294 | walwriter           |                              |

(7 rows)





# Nové wait eventy

- Latches
  - Extensions
  - Client/socket
  - Timeout
  - ...
- I/O events
  - jednotlivě identifikované (datafile, WAL, ...)
  - Reads
  - Writes
  - Flushes
  - ...



# Role pro monitoring

- Umožňují používání běžných uživatelů pro monitoring.
  - `pg_read_all_settings`
  - `pg_read_all_stats`
  - `pg_stat_scan_tables`
  - `pg_monitor`



# Funkce pro monitoring

- `pg_current_logfile()`
  - aktuální soubor do kterého se loguje
- `pg_ls_logdir()`
- `pg_ls_waldir()`
  - výpis obsahu log/WAL adresáře
  - pro ne-superuživatele (přes `pg_monitor` roli)
- `pg_hba_file_rules()`
  - výpis obsahu `pg_hba.conf` přes SQL



# Vývojáři a SQL novinky



# regexp\_match

```
regexp_match(string text,  
             pattern text [, flags text])
```

- V podstatě `regexp_matches` ale vrací jen první výsledek.
- Jednodušší na místech která očekávají jeden výsledek.

```
SELECT id, regexp_match(body, '<title>(.*?)</title>')  
FROM t;
```



# IDENTITY sloupce

- něco jako (BIG)SERIAL ale odpovídající SQL Standardu
  - víceméně (standard umožňuje jediný sloupec per tabulka)
- hodnoty se stále generují ze sekvence
- řeší některé “nedomyšlenosti” (BIG)SERIAL sloupců
  - samostatná přístupová práva na sekvenci
  - CREATE TABLE / LIKE kopíruje default ale ten odkazuje na tu původní stejnou sekvenci :-)
  - neexistuje ALTER TABLE ... ALTER ... SET/DROP SERIAL;
  - ... DROP DEFAULT neodstraní sekvenci



# IDENTITY sloupce

```
CREATE TABLE t (  
    id INTEGER GENERATED ALWAYS AS IDENTITY,  
    ...,  
    ...,  
    ...,  
);
```



# Indexy

- Hash indexy
  - (konečně) crash-safe a replikované
  - řešilo se jestli odstranit úplně nebo přidat WAL
- BRIN indexy
  - agresivnější sumarizace (nová page range)
  - sumarizace na vyžádání (`brin_summarize_page`)





# Paralelní dotazy

- další “driver” scany
  - Parallel Index Scan
  - Parallel Bitmap Index Scan
- Parallel Merge Joins
- Gather Merge
- paralelizace dotazů z procedurálních jazyků (PL/pgSQL)
- `max_parallel_workers` ( $\leq$  `max_worker_processes`)



# Deklarativní partitioning

```
-- parent
CREATE TABLE messages (
    log_date TIMESTAMP;
    log_message
) PARTITION BY RANGE (log_date);

-- child
CREATE TABLE messages_2017_01
PARTITION OF messages
FOR VALUES FROM ('2017-01-01 00:00:00')
                TO ('2017-02-01 00:00:00')
```



# Deklarativní partitioning

- méně flexibilní než trigger-based partitioning
  - v triggeru si můžete napsat cokoliv
- automatické routování řádek
  - nemusíte psát žádné triggery
  - žádné neošetřené krajní situace
- umožňuje různé pokročilé optimalizace dotazů
  - databáze “rozumí” pravidlům a může toho využít



# RLS

- výrazné vylepšení plánování dotazů
  - původně často velmi neefektivní plány
  - to je nyní víceméně vyřešené (velmi dobré plány)
- standardní chování RLS je “permisivní”
  - pokud alespoň jedna z politik platí, přístup povolen
- nově možné i “restriktivní” chování
  - politika musí platit bez ohledu na ostatní



# Lepší fulltext na JSON(B)

- fulltext zacházel s JSON(B) víceméně jako s textem
  - např. fráze se hledaly v sousedních položkách
- vylepšeny dvě funkce aby toto respektovaly
  - `to_tsvector()`
  - `ts_headline()`



# Zálohy a replikace



# výchozí konfigurace

- Výchozí konfigurace nyní umožňuje backupy / replikaci.
  - `wal_level = replica`
  - `max_wal_senders = 10`
  - `max_replication_slots = 10`
  - `hot_standby = on`
- `pg_hba.conf` - lokální replikační spojení (`pg_basebackup`)
- `pg_basebackup` nyní používá “`-X stream`” by default



# replikační lag

- `pg_stat_replication` nyní obsahuje časové metriky
  - `write_lag` (flush - write)
  - `flush_lag` (flush - flush)
  - `apply_lag` (flush - apply)





# logická replikace

```
-- master
CREATE PUBLICATION p
  FOR TABLE t
  WITH ( publish = 'insert, delete' );

-- replika
CREATE SUBSCRIPTION s
  CONNECTION 'host=master dbname=db1'
  PUBLICATION p
  WITH ( create_slot = false );
```



Otázky?



# PostgreSQL 11



# PG 11 / committed

- Partitioning
  - Hash partitioning
  - partition-wise join for declarative partitioned tables
- Parallelism
  - Parallel tuplesort (for parallel B-Tree index creation)
  - Parallel Hash
  - Parallel Append
  - local partitioned indexes



# PG 11 / maybe

- SERIALIZABLE with parallel query
- Incremental sort
- Partition-wise aggregation/grouping
- ALTER TABLE ADD COLUMN fast default
- Logical decoding of 2PC, TRUNCATE, streaming
- UPDATE of a partition key
- jsonpath, SQL/JSON
- shared ispell dictionaries
- MERGE



<https://commitfest.postgresql.org/17/>



Otázky?