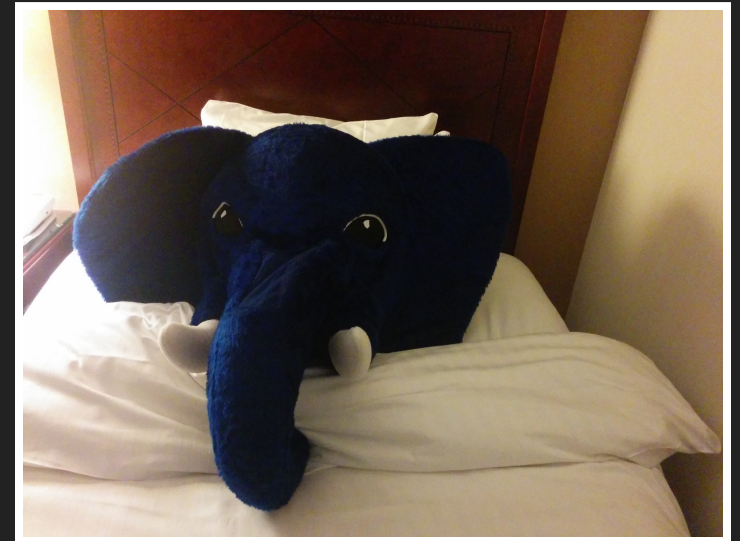


A look at the Elephants Trunk

PostgreSQL 10

PGDay UK
London, UK

Magnus Hagander
magnus@hagander.net



Magnus Hagander

- Redpill Linpro
 - Principal database consultant
- PostgreSQL
 - Core Team member
 - Committer
 - PostgreSQL Europe

PostgreSQL 10

A new era of versioning

Versioning

- ...
- 8.4
- 9.0
- ...
- 9.4
- 9.5
- 9.6

Versioning

- 9.6
 - 9.6.2
- 10
 - 10.1
- 11
- 12

Development schedule

- August 2016 - branch 9.6
- September 2016 - CF1
- November 2016 - CF2
- January 2017 - CF3
- March 2017 - CF4
- May 2017 - Beta1!
 - *Please test!*
- *(August 2017 - Beta2)*

New Features

- DBA and administration
- Monitoring
- Developer and SQL features
- Backup and replication
- Performance

First things first

BREAK



ALL THE THINGS

memegenerator.net

The small things

- Drop support for protocol 1.0
 - (No more clients < 6.3)
- Drop support for floating point timestamps

The bigger things

Directory `pg_xlog` is now `pg_wal`

Directory `pg_clog` is now `pg_xact`

Even bigger...

`pg_switch_xlog()` is now `pg_switch_wal()`
`pg_xlogfile_name()` is now `pg_walfile_name()`
`pg_current_xlog*()` is now `pg_current_wal*`
`pg_last_xlog*()` is now `pg_last_wal*`
`pg_xlog_location_diff()` is now
`pg_wal_location_diff()`

One more...

pg_receivexlog is now pg_receivewal

pg_resetxlog is now pg_resetwal

pg_xlogdump is now pg_waldump

Last one!

pg_basebackup --xlog-method is now --wal-method

pg_basebackup --xlogdir is now --waldir

OK, some good news

- Logging about multixact wraparound protection less chatty
 - Don't be scared...

New Features

- DBA and administration
- Monitoring
- Developer and SQL features
- Backup and replication
- Performance

SCRAM authentication

- Salted Challenge Response Authentication
- Standardized way to do auth
- More secure than md5!
- Switch when your clients support it

libpq enhancements

- Multiple hosts can be specified

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

- Writable host can be requested

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

- .pgpass overridden in connection string

```
host=pg1 passfile=/some/where/mypgpass
```

New Features

- DBA and administration
- **Monitoring**
- Developer and SQL features
- Backup and replication
- Performance

pg_stat_activity

- walsender processes now visible

```
-[ RECORD 2 ]-----+-----  
datid          |  
datname        |  
...  
application_name | pg_receivewal  
...  
backend_start   | 2017-03-19 16:09:59.842833+01  
...  
wait_event      | WalSenderMain  
state           | active  
...  
backend_type    | walsender
```

pg_stat_activity

- Background worker processes now visible!

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

New wait events

- Latches
 - Extensions
 - Client/socket
 - Timeout
 - ...
- I/O events
 - Reads
 - Writes
 - Individually identified

Monitoring roles

Avoid superuser!

- pg_read_all_settings
- pg_read_all_stats
- pg_stat_scan_tables
- pg_monitor

New Features

- DBA and administration
- Monitoring
- Developer and SQL features
- Backup and replication
- Performance

regexp_match

- `regexp_match()`
 - Like *regexp_matches* but not SRF
 - No subquery needed

IDENTITY columns

- Same functionality as *SERIAL*
 - Minus the permissions gotcha!
- SQL standard

```
CREATE TABLE itest (  
  a int GENERATED BY DEFAULT AS IDENTITY,  
  b int GENERATED ALWAYS AS IDENTITY  
)
```

Data ingestion

- *file_fdw* can now use program

```
CREATE FOREIGN TABLE
test(a int, b text)
SERVER csv
OPTIONS (program 'gunzip -c /tmp/data.csv.gz');
```

XMLTABLE

- (almost) per SQL standard
- Convert XML document to resultset
- Mapping XPath etc
- Much faster than individual queries

JSON(b) FTS

- JSON-aware full text search
- Working `ts_headline()`

```
postgres=# SELECT to_tsvector('{\"foo\": \"bar\", \"baz\": 3}');  
          '3':4 'bar':2 'baz':3 'foo':1
```

```
postgres=# SELECT to_tsvector('{\"foo\": \"bar\", \"baz\": 3}'::jsonb);  
          'bar':1
```

```
postgres=# SELECT ts_headline('{\"foo\": \"bar\", \"baz\": 3}', 'foo');  
{\"<b>foo</b>\": \"bar\", \"baz\": 3}
```

```
postgres=# SELECT ts_headline('{\"foo\": \"bar\", \"baz\": 3}'::jsonb, '  
          {\"baz\": 3, \"foo\": \"bar\"}
```

ICU collations

- More choice for collations
 - Not just OS ones
- Stable across versions
 - Except ICU major versions
 - But those are *detected*

ICU collations

```
SELECT * FROM t ORDER BY a COLLATE "sv-SE-x-icu";
```

```
valle  
vera  
walle  
wera
```


ICU collations

```
SELECT * FROM t ORDER BY a COLLATE "sv-SE-u-co-standard-x-icu";
```

```
valle  
walle  
vera  
wera
```

New Features

- DBA and administration
- Monitoring
- Developer and SQL features
- Backup and replication
- Performance

New defaults

- New postgresql.conf defaults:
 - wal_level = replica
 - max_wal_senders = 10
 - max_replication_slots = 10
- New pg_hba.conf defaults
 - Replication connections by default

Replication slots

- Support for temporary replication slots
- Automatically dropped at end of session
- Prevents fall-behind with less risk

pg_basebackup

- WAL streaming supported in tar mode (*-Ft*)
- Better excludes
- New defaults
 - WAL streaming (*-X stream*) now default
 - Uses temporary replication slots by default

Quorum based sync replication

- Support *ANY* and *FIRST* mode
- Previously only *FIRST*

```
synchronous_standby_names=  
FIRST 2 (pg1, pg2, pg3, pg4)
```

```
synchronous_standby_names=  
ANY 2 (pg1, pg2, pg3, pg4)
```

Logical replication

- Based on WAL
- And logical decoding
- Replicate individual tables
 - Or sets of tables

Logical replication

```
CREATE TABLE testtable (a int PRIMARY KEY, b text);
```

```
CREATE PUBLICATION testpub FOR TABLE testtable;
```


Logical replication

```
CREATE TABLE testtable (a int PRIMARY KEY, b text);
```

```
CREATE SUBSCRIPTION testsub  
  CONNECTION 'host=/tmp port=5500 dbname=postgres user=mha'  
  PUBLICATION testpub;
```

Limits

- No schema replication
- No sequence replication
- Not suitable for fail-over

New Features

- DBA and administration
- Monitoring
- Developer and SQL features
- Backup and replication
- Performance

Hash indexes

- Now WAL logged
 - So actually useful
- Many performance enhancements
 - Better caching
 - Supports page-level vacuum
 - ...
- Sometimes better than btree

Partitioning

- Based on existing inheritance
 - Same as old "manual partitioning"
- Easier to work with
 - Automatic tuple routing
- More limitations -> more optimizations
 - Many not there yet

Partitioning

- Range partitioning
 - (Single column only)
- List partitioning

Partitioning

```
CREATE TABLE testlog (t timestamptz DEFAULT now(), txt text)
PARTITION BY RANGE(t);
```

```
CREATE TABLE testlog_2017
  PARTITION OF testlog (t)
  FOR VALUES FROM ('2017-01-01') TO ('2018-01-01');
```

```
INSERT INTO testlog (txt) VALUES ('test');
```

Partitioning

```
CREATE TABLE testcat (category text, txt text)  
PARTITION BY LIST(category);
```

```
CREATE TABLE testcat_cat13  
  PARTITION OF testcat (category)  
  FOR VALUES IN ('cat1', 'cat2', 'cat3');
```

```
INSERT INTO testcat VALUES ('cat1', 'Test1');
```


Partitioning

- Still many limitations
 - No row-movement
 - No cross-partition indexes
 - No cross-partition keys
 - No partition-wise processing
 - No tuple routing for foreign partitions

More parallelism

- 9.6 introduced parallelism
 - Sequential scans
 - Aggregates
 - Hash and loop joins

Usability

- New parameter *max_parallel_workers*
- query string now in workers
 - Shows in `pg_stat_activity`

```
pid                | 28040
...
wait_event_type    | Timeout
wait_event         | PgSleep
state              | active
query              | select x, pg_sleep(2000) from tt;
backend_type       | background worker
```

Index scans

- Regular index scans (btree)
- Index Only scans (btree)
- Bitmap Heap Scan
 - Index still scanned serially

Joins

- Merge joins

Multi column statistics

- Collect statistics across columns
 - Previously each column individually
- Combinations must be explicitly selected

```
CREATE STATISTICS test_stats ON b,c FROM test
```

- Collects *dependency* and *n_distinct*

That's a lot!

There's always more

- Lots of smaller fixes
- Performance improvements
- etc, etc
- Can't mention them all!

Please help!

Please help!

- Download and test!
 - apt packages available
 - rpm/yum packages available
 - Both beta and dev snapshots!

Thank you!

Magnus Hagander

magnus@hagander.net

@magnushagander

<http://www.hagander.net/talks/>

This material is licensed

