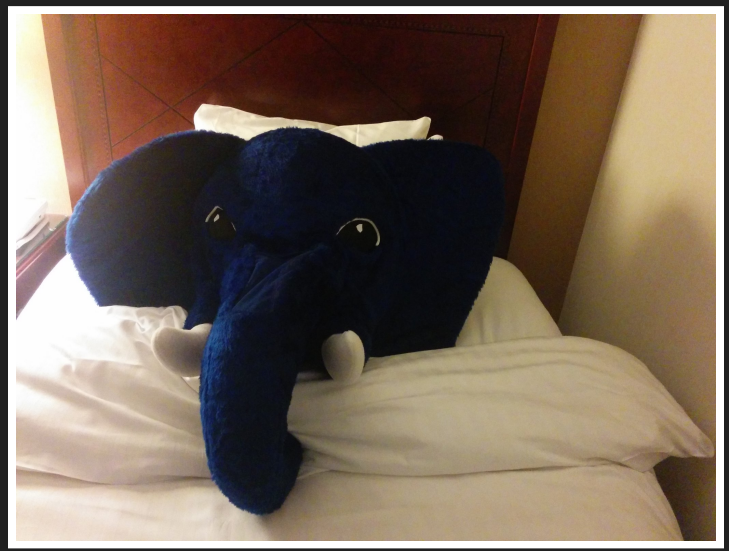
A look at the Elephants Trunk

PostgreSQL 12

Postgres Open 2019
Orlando, FL

Magnus Hagander
magnus@hagander.net



Magnus Hagander

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

PostgreSQL 12

PostgreSQL 12

NOT done yet!

- (*almost*)

Development schedule

- July 2018 - branch 11
- July 2018 - CF1
- September 2018 - CF2
- November 2018 - CF3
- January 2019 - CF4
- March 2019 - CF5
- September 2019 - *Beta 4*

Breaking stuff!

Datatype removal

- abstime
- reltime
- tinterval

Random numbers

- Removed support for --disable-strong-random

(more later)

New features

- DBA and administration
- SQL and developer
- Backup and replication
- Performance

psql

```
postgres=# \h vacuum
Command:     VACUUM
Description: garbage-collect and optionally analyze a database
Syntax:
...
...


URL: https://www.postgresql.org/docs/devel/sql-vacuum.html
```

Figures in documentation

- Graphs and stuff!

New features

- DBA and administration
- SQL and developer
- Backup and replication
- Performance

pg_stat_ssl

- Added columns:
  - client_serial
  - issuer_dn
- Mask data for unprivileged users

SSL configuration

- Control min/max SSL version
  - ssl_min_protocol_version=TLSv1
  - ssl_max_protocol_version=''
- Any supported TLS level
  - TLSv1, TLSv1.1, TLSv1.2, TLSv1.3

GSSAPI encryption

- Encryption without SSL
- No need for certificates etc
- Assuming GSSAPI already in place
- *pg_stat_gssapi*

VACUUM

- SKIP_LOCKED
  - Skip any relation not immediately lockable
- DISABLE_PAGE_SKIPPING
  - Bypasses visibility map
  - For debugging!

VACUUM

- *vacuumdb*
  - --min-xid-age
  - --min-mxid-age

COPY

- COPY FROM WHERE

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

psql

- CSV output format

```
postgres=# \pset format csv
Output format is csv.
postgres=# SELECT * FROM mytable;
a,b,c
7,8,9
```

pg_stat_statements

- Reset individual query statistics

```
SELECT pg_stat_statements_reset(queryid => -6363133595459221451);
```

Progress monitoring

- CREATE INDEX
- REINDEX
- CLUSTER

REINDEX CONCURRENTLY

Checksums

- Offline enable/disable
- Progress report for checking

Pluggable access method

- API for storage engines
- Won't have >1 engine
  - Yet

New features

- DBA and administration
- SQL and developer
- Backup and replication
- Performance

WITH OIDS

- Removed!
  - (but you weren't using it, right?)
  - (deprecated since 2005)
- oid is now a regular column
  - No more magic!

Generated columns

- Columns with results of computation
  - Only STORED
  - Without triggers

```
CREATE TABLE foo (
   a int NOT NULL,
   b int GENERATED ALWAYS AS (a*2) STORED
);
```

ENUMs

- Addition of enum inside transactions

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

ENUMs

- New restriction

```
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 to query JSON
- New "query language"
- Still same indexing!

JSONPATH

New functions and operators

- 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';
```

CTEs

- Optimization barrier
- Or is it?

CTEs

- New materialization keyword

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

CTEs

```
QUERY PLAN
-------------------------------------------------------------------------
 Append  (cost=2.02..2.06 rows=2 width=17)
   CTE t
     ->  Seq Scan on foo  (cost=0.00..1.01 rows=1 width=17)
   CTE t2
     ->  Seq Scan on foo foo_1  (cost=0.00..1.01 rows=1 width=17)
   ->  CTE Scan on t  (cost=0.00..0.02 rows=1 width=17)
   ->  CTE Scan on t2  (cost=0.00..0.02 rows=1 width=17)
(7 rows)
```

CTEs

- New materialization keyword

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

CTEs

```
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=2550 width=
   ->  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)
```

New features

- DBA and administration
- SQL and developer
- Backup and replication
- Performance

max_wal_senders

- No longer part of *max_connections*
- Separate setting
- Now dedicated to wal senders

recovery.conf

- Integrated in postgresql.conf
- In main config, or include file
  - E.g. postgresql.auto.conf

recovery.signal

- New "trigger" file
- Since recovery.conf doesn't exist
- *Update your scripts!*

standby.signal

- New "trigger" file
- For standby mode

recovery.conf

- Reconfigure with reload only
  - recovery_min_apply_delay
  - archive_cleanup_command
  - ...

recovery_target_timeline

- New default: **latest**

pg_promote()

- SQL function to promote standby

Exclusive base backups

- Even more deprecated!

New features

- DBA and administration
- SQL and developer
- Backup and replication
- Performance

SP-GiST

- Now supports KNN searches

GiST, GIN and SP-GiST WAL

- Less WAL generated during index build
- Faster and better!

Detoast compressed datum

- Partial decompression of TOAST
- Don't decompress whole Datum when only part is needed
- Useful for eg postgis

Partitioning

- More flexible partition bounds
  - Generalized expressions

Partitioning

- Locking delayed until scan
- Can lead to much faster scans
  - When many partitions are involved
- ATTACH without access exclusive lock

Partitioning

- Multi-inserts for COPY
- (simply: faster COPY into partitioned tables)

Partitioning

- Foreign keys referencing partitioned tables
  - When partitioned table has PRIMARY KEY

SERIALIZABLE

- For parallel query

JIT compilation

- Now enabled by default

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
  - Windows installers available

Thank you!

Magnus Hagander
magnus@hagander.net
@magnushagander
https://www.hagander.net/talks/