A look at the Elephants Trunk

PostgreSQL 13

Percona Live
Online

Magnus Hagander
magnus@hagander.net

Magnus Hagander

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

PostgreSQL 13

Development schedule

- July 2019 - branch 12
- July 2019 - CF1
- September 2019 - CF2
- November 201 - CF3
- January 2020 - CF4
- March 2020 - CF5
- September 2020 - Release

New features

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

DROP DATABASE

- Drop database
- And automatically disconnect users!

DROP DATABASE

- Drop database
- And automatically disconnect users!

```
DROP DATABASE production
```

DROP DATABASE

- Drop database
- And automatically disconnect users!

```
DROP DATABASE production
```

- Hangs...

DROP DATABASE

- Drop database
- And automatically disconnect users!

```
DROP DATABASE production
```

- Hangs...

```
DROP DATABASE production WITH (FORCE)
```

Breaking changes

- Not as bad as some..

Wait events renamed

- Many wait events renamed
- More consistent now
- May cause issues with monitoring

pg_stat_statements

- Timing columns renamed
  - total_time -> total_exec_time
  - min_time -> min_exec_time
  - max_time -> max_exec_time
  - mean_time -> mean_exec_time
  - stddev_time -> stddev_exec_time

New features

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

ALTER STATISTICS

- Statistics target on extended statistics
- Separate from underlying columns

ALTER STATISTICS

- Statistics target on extended statistics
- Separate from underlying columns

```
ALTER STATISTICS foo
   SET STATISTICS 1000;
```

ANALYZE

- Progress report for analyze
- How far along is the analyze
- How much has been generated

```
SELECT * FROM pg_stat_progress_analyze
```

Parallel VACUUM

- (partially so)
- Parallel processing of indexes
- Main vacuum is still serial

```
VACUUM PARALLEL mytable
```

Autovacuum

Trigger by INSERT

- Triggers VACUUM for insert only workloads
  - Not just for anti-wraparound
- More frequent, cheaper, runs
- autovacuum_vacuum_insert_scale_factor
- autovacuum_vacuum_insert_threshold

Slow query sampling

- Log sample of slow queries
- Instead of all of them
- Prevent overload of system due to logging
- instead of *log_min_duration_statement*

Slow query sampling

```
log_min_duration_sample = 100ms
log_statement_sample_rate = 0.1
```

- *log_min_duration_statement* works independently

SSL

SSL versions

- New default: TLSv1.2
  - Can still be lowered
- SSL version in libpq connection string
  - *sslminprotocolversion*
  - *sslmaxprotocolversion*

postgres_fdw

- Specify *sslkey* and *sslcert*
  - In of user mapping
- Specify *password_required=false*
  - Superuser only
  - Allow use of e.g. peer or gss

pg_stat_slru

- Statistics about SLRU structures
- pg_xact, pg_subtrans, pg_multixact etc

pg_stat_activity

- New column *leader_pid*
- Shows leader in parallel query
- NULL for non-parallel query

Trusted extensions

- Extensions installed as non-superuser
- Also replaces PL templates
- Several built-in ones migrated
  - btree_gist
  - intarray
  - pgcrypto
  - ...

New features

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

UUID

- Random generation now in core
- Covers most use-cases
- No need for extension

UUID

- Random generation now in core
- Covers most use-cases
- No need for extension

```
postgres=# SELECT gen_random_uuid();
           gen_random_uuid
--------------------------------------
 f34f4732-1ce9-46e1-a3e1-4da94d6e07db
```

JSONPATH

- Now supports datetime

JSONPATH

- Now supports datetime

```
postgres=# SELECT jsonb_path_query(
  '{"t": "2020-03-05 14:15:10"}'::jsonb,
  '$.t .datetime() < "2020-03-05 15:00:22" .datetime()'::jsonp
 jsonb_path_query
------------------
 true
```

FETCH FIRST WITH TIES

FETCH FIRST WITH TIES

```
postgres=# SELECT * FROM n ORDER BY i FETCH FIRST 2 ROWS ONLY;
 i
---
 1
 2
(2 rows)

postgres=# SELECT * FROM n ORDER BY i FETCH FIRST 2 ROWS WITH
 i
---
 1
 2
 2
(3 rows)
```

New features

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

pg_stat_progress_basebackup

- Shows process while basebackup is running
- Requires progress report enabled to show details
  - On by default with pg_basebackup

Backup manifests

- Included in base backups
  - By default
- List of files, checksums
- New tool:

```
pg_verifybackup
```

Logical replication of partitioned tables

- Replicates individual partitions
  - Each partition replicated separately
  - Published individually or via root

```
publish_via_partition_root=true
```

- Receive into partitioned table

Limit slot disk usage

- max_wal_slot_keep_size
- Avoids running out of disk on primary
- Breaks replication/archiving if hit

Online reconfiguration

- *primary_conninfo* and *primary_slot_name*
- Change in file or with *ALTER SYSTEM*
- Only a *reload* needed

New features

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

Incremental sort

- Optimize multi-level sort
- When "lower level" in query is partially sorted
- Split in "buckets" and only sort individually

Disk based hash aggregate

- Hash aggregates used in more cases
  - Even when work_mem is not enough
- Avoid OOMing on bad estimates
- GROUP BY, DISTINCT and grouping sets

WAL usage tracking

```
postgres=# EXPLAIN (ANALYZE, WAL) INSERT INTO n VALUES (11);
                                                QUERY PLAN
-----------------------------------------------------------------------------------------
 Insert on n  (cost=0.00..0.01 rows=1 width=4) (actual time=0.
   WAL: records=1 bytes=59
   ->  Result  (cost=0.00..0.01 rows=1 width=4) (actual time=0
 Planning Time: 0.026 ms
 Execution Time: 0.094 ms
(5 rows)
```

pg_stat_statements

- WAL usage tracking
- Planner statistics
- Renames execution statistics columns

pg_stat_statements

```
postgres=# SELECT * FROM pg_stat_statements WHERE ...;

-[ RECORD 1 ]-------+--------------------------
query               | insert into n values ($1)
plans               | 1
total_plan_time     | 0.033068
min_plan_time       | 0.033068
max_plan_time       | 0.033068
mean_plan_time      | 0.033068
stddev_plan_time    | 0
....
wal_records         | 1
wal_fpi             | 1
wal_bytes           | 618
```

Deduplication in btree

- Indexes with many duplicates
  - Full set of columns
- Reduced storage
- Improved performance

There's always more

There's always more

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

Thank you!

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