New features of PostgreSQL 9.5

News from a hacker

Michael Paquier

Moscow, PgConf Russia 2015

2015/02/07

About the lecturer

- Michael Paquier
  - Working on Postgres for VMware
  - Community hacker and blogger
  - Based in Tokyo
- Contacts
  - Twitter: @michaelpq
  - Blog: http://michael.otacoo.com
  - email: michael@otacoo.com

Resources

- Planet Postgres: planet.postgresql.org
- Blogs
  - Depesz: www.depesz.com
  - My stuff: michael.otacoo.com
  - Development documentation: http://www.postgresql.org/docs/devel/static/

Current status

- Postgres 9.5 development
  - Last Commit Fest to begin on 15th Feb
- Feature freeze soon
- First Beta for April
- GA for September (?)

Categories

- **SQL-related features**
- Administration and DBA
- Infrastructure and replication
- To-be-committed features?

BRIN indexes

- Index entry: range of data by scanning a number of blocs
- Entry match: check that entry fits in a range
- Mix of:
  - index scan: look in which group of blocks value is by scanning ranges
  - Seq scan: Scan the set of blocks for matches
- Operators
  - minmax: 1D-operators
  - Possible to get 2D-stuff like box (WIP)
- Data warehouse: geographical data, ordered data

UPDATE tab SET (col1, col2)

- Nice with functions!
- Simplify UPDATE … SET … FROM ()

```
CREATE TABLE table1 (a int, b int, c int);
CREATE TABLE table2 (a int, b int, c int);
UPDATE table1 AS t1
SET b = t2.b, c = t2.c
   FROM (SELECT a, b, c FROM table2) AS
t2 WHERE t1.a = t2.a;
```

```
UPDATE table1
SET (b, c) = (SELECT b, c FROM table2
                  WHERE table2.a = table1.a);
```

SKIP LOCKED for row-level locks

- New layer for FOR [SHARE | UPDATE | …]
  - Default, wait for lock
  - NOWAIT, error instead of waiting
  - SKIP LOCKED, skip locked tuples
- View of data not consistent
- Reduction of lock contention
- For queue-like tables

ALTER TABLE SET LOGGED / UNLOGGED

- Unlogged table (9.1~)
  - Permanent table storage
  - Not WAL'ed, not persistent on crash
- Change persistence of table
  - LOGGED / UNLOGGED switch
  - Complete table rewrite
  - Unlogged -> logged faster than logged -> unlogged

IMPORT FOREIGN SCHEMA (1)

- New API hook for Foreign Data wrappers
  - Returns list of CREATE FOREIGN TABLE strings

```
List *
ImportForeignSchema(ImportForeignSchemaStmt *stmt,
                    Oid serverOid);
```

- Query

```
=# IMPORT FOREIGN SCHEMA public
    FROM SERVER postgres_server INTO public;
 IMPORT FOREIGN SCHEMA
```

IMPORT FOREIGN SCHEMA (2)

- postgres_fdw
  - import_collate
  - import_default
  - import_not_null
- Use in-core example for your own FDW

width_bucket()

- Find to which bucket operand would be assigned in a list
  - Actual number to place
  - Boundaries of bucket
  - Number of buckets
- Available with array defining the lower bounds of buckets

```
width_bucket(1.5, 0, 10, 20) => 4
width_bucket(1.5, array[0, 1, 2, 3]::numeric[]) => 2
```

generate_series(numeric)

- Now for numeric
- Already here for:
  - Timestamps
  - Integers

```
=# SELECT generate_series(-4.9, 5.5, 2.2);
 generate_series
-----------------------
          -4.9
          -2.7
          -0.5
           1.7
           3.9
(5 rows)
```

Categories

- SQL-related features
- **Administration and DBA**
- Infrastructure and replication
- To-be-committed features?

Row-level security (1)

- Horizontal control of tuple access
- ALTER TABLE ENABLE ROW LEVEL SECURITY
  - Control switch
  - Default is off
- CREATE POLICY
  - USING for check on existing rows
  - WITH CHECK for new rows
  - Multiple policies apply with OR, not AND

Row-level security – Example (2)

- Data set

```
-- Create data
CREATE ROLE admin SUPERUSER;
CREATE ROLE alice LOGIN;
CREATE TABLE employee (name text, salary int);
ALTER TABLE employee ENABLE ROW LEVEL SECURITY;
INSERT INTO employee VALUES ('CEO', 10000000);
INSERT INTO employee VALUES ('alice', 1000);
-- Allow SELECT access to a normal employee
GRANT SELECT ON employee TO alice;
```

Row-level security – Example (3)

- And no data for Alice

```
-- Admin view
CREATE POLICY admin_view ON employee TO admin
     USING (true) WITH CHECK (true);
-- Only allow a vanilla user to view its own stuff
CREATE POLICY own_view ON employee FOR SELECT
    USING (current_user = name);

-- And Alice…
$ psql -At -c "SELECT * FROM employee;" -U alice postgres
alice|1000
```

REINDEX SCHEMA

- Reindex all relations on a single schema
- Including toast relations
- Good for maintenance of multi-schema environments
- Notes:
  - One transaction per relation
  - Exclusive lock
  - REINDEX SCHEMA pg_catalog <=> REINDEX SYSTEM

Event triggers – table_rewrite

- New event: table_rewrite
- Kicked for some flavors of ALTER TABLE
- Define rewrite policies
  - Schema, user control
  - Scheduling
  - Table size
  - Etc.

Parallel VACUUM with vacuumdb

- Defined by jobs -j
- Support for custom list of tables
- Support for ANALYZE-related options
- Largest tables first
- Risk of deadlocks with FULL on catalogs
  - Low number of application tables
  - High number of jobs

ALTER SYSTEM RESET

- Reset value of one parameter in postgresql.conf.auto
- RESET ALL for all parameters

```
=# ALTER SYSTEM SET shared_buffers = '100GB’;
ALTER SYSTEM
=# ALTER SYSTEM SET work_mem = '1TB’;
ALTER SYSTEM
=# ALTER SYSTEM RESET work_mem;
ALTER SYSTEM
=# \! cat $PGDATA/postgresql.auto.conf
# Do not edit this file manually!
# It will be overwritten by ALTER SYSTEM command.
shared_buffers = '100GB'
```

Categories

- SQL-related features
- Administration and DBA
- **Infrastructure and replication**
- To-be-committed features?

Abbreviated keys

- Improve sort of text Datum
  - Use first bytes of strxfrm
  - Fallback to old method if 8-first bytes identical
- Performance
  - CREATE INDEX time 3x faster
  - Slower for 8-first bytes identical
  - Good for raw dumps
- Available on Windows as well

New WAL format

- Improve detection of relation blocks touched by a WAL record
- Stats:
  - 93 files changed
  - 3945 insertions(+)
  - 4366 deletions(-)
- New API for redo and WAL record construction
- xlog.c split a bit more
- Useful for tools => pg_rewind

Actions at the end of recovery

- New parameter recovery_target_action
- In recovery.conf
- Values when recovery target is reached
  - pause, same as pause_at_recovery_target
  - promote, perform promotion to next timeline
  - Shutdown, stop the node
- Work for target XID, timestamp, name

Improvements of Windows build

- Addition of file versioning
  - 4-digit number
  - File information
  - All libs and bins covered
  - Useful for packagers and upgrade processes
- --extra-version support in MSVC
- Shared Libraries installed in bin/ and lib/ (MSVC pending)

Logging of replication commands

- Control with GUC log_replication_commands
- Default = off
  - Log messages as DEBUG1
  - Same as prior releases
- When on, written as LOG

```
LOG: received replication command: IDENTIFY_SYSTEM
LOG: received replication command: START_REPLICATION
0/3000000 TIMELINE 1
```

cluster_name

- Set custom string in process title
- Identify instances on same host

```
35839  ??  Ss    0:00.04 postgres: pg_5432: checkpointer process
35873  ??  Ss    0:00.02 postgres: pg_5433: checkpointer process
35909  ??  Ss    0:00.02 postgres: pg_5434: checkpointer process
35956  ??  Ss    0:00.03 postgres: pg_5435: checkpointer process
36001  ??  Ss    0:00.02 postgres: pg_5436: checkpointer process
36044  ??  Ss    0:00.02 postgres: pg_5437: checkpointer process
36095  ??  Ss    0:00.03 postgres: pg_5438: checkpointer process
36223  ??  Ss    0:00.00 postgres: pg_5439: checkpointer process
36353  ??  Ss    0:00.00 postgres: pg_5440: checkpointer process
```

Exported snapshots with pg_dump

- Get database at state of slot creation

```
$ psql "replication=database dbname=mpaquier" [...]
=# CREATE_REPLICATION_SLOT foo3 LOGICAL test_decoding;
slot_name | consistent_point | snapshot_name | output_plugin
---------------+----------------------+----------------------+------------------
          foo |        0/16ED738 |       000003E9-1 | test_decoding
(1 row)
[… hold on replication connection …]
pg_dump --snapshot 000003E9-1
```

- Base image to consume changes

And more…

- Features for replication
  - Niche use-cases: Slony, BDR, UDR…
  - Commit timestamp
  - Textual representation of objects
- pageinspect for GIN and BRIN indexes
- Scalability work
- etc.

Categories

- SQL-related features
- Administration and DBA
- Infrastructure and replication
- **To-be-committed features?**

UPSERT

- If fully-baked, number #1 for 9.5
- Grammar:
  - Agreement (!)
  - ON CONFLICT [UPDATE | IGNORE]
- Still development work going on…
- Commit forecast: cloudy

WAL compression

- Reduce WAL I/O with some CPU cost
- Move of pglz in src/common
- Patches advanced
- Agreement on spec
  - Session switch wal_compression
- Commit forecast: sunny

ALTER TABLE log_min_vacuum_duration

- log_min_vacuum_duration at relation level
- Filter autovacuum report spams in logs
- Anyone caring about that?
- Commit forecast: rainy-cloudy

File-level incremental backup

- Get file-level differential backups from LSN position
- Targets niche use-case applications, like:
  - Set of relations with few updates
  - Set of relations with a lot of updates
- pg_rman does it, at block level.
- Commit forecast: rainy

New types for abbreviated keys

- For Datum sorts
- And numeric sorts
- Infrastructure already in place
- Forecast for 9.5: sunny

CRC improvements

- Slice-by-8 algorithm (?)
- Cool CPU improvements for WAL generation
- Commit forecast: sunny

Redesign of checkpoint_segments

- GUC checkpoint_wal_size
  - Set directly disk space needed by WAL
  - Hard limit, visibly
  - Intuitive
  - No more formula needed
- Commit forecast: sunny

pgaudit

- Audit capabilities:
  - At user level
  - At table level
- Spec still unsure
  - AUDIT in-core clause
  - Simple contrib module (?)
- Feature has been asked for years
- Commit forecast: rainy-cloudy

Questions?