2ndQuadrant
Professional PostgreSQL

Logical Replication in PostgreSQL

Petr Jelínek
PostgreSQL Contributor & Consultant

History

Postgres Replication History

- PostgreSQL 7.x (~2000)
  - Replication should not be part of core Postgres
- PostgreSQL 8.0 (2005)
  - Point-In-Time Recovery
- PostgreSQL 9.0 (2010)
  - Streaming Replication (physical)
- PostgreSQL 9.4 (2014)
  - Logical Decoding (changeset extraction)

Logical Replication History

- Trigger based solutions
  - Slony (~2004)
  - Londiste (~2007)
- Result of the no replication in core philosophy
- Use table(s) as queue
  - Amplify writes on the upstream
- Written in scripting languages
- But work for the most part

BDR Project

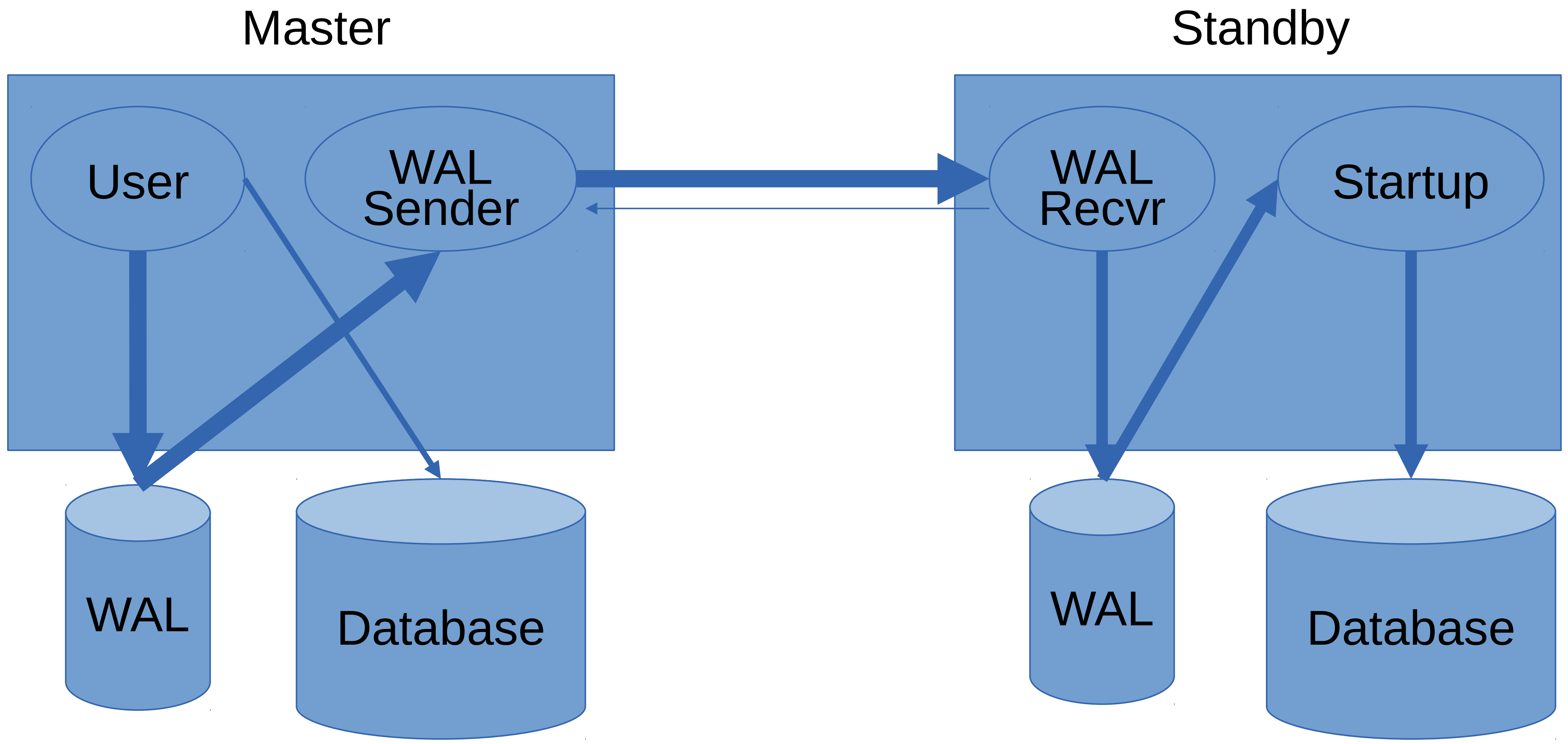
- Native logical replication
- Multi-master
- Firm design 2011
- Prototype 2012
- Resulted in many patches to PostgreSQL 9.3, 9.4, 9.5 and now 9.6
  - For example Logical Decoding
- Not **yet** fully integrated into PostgreSQL

Current state

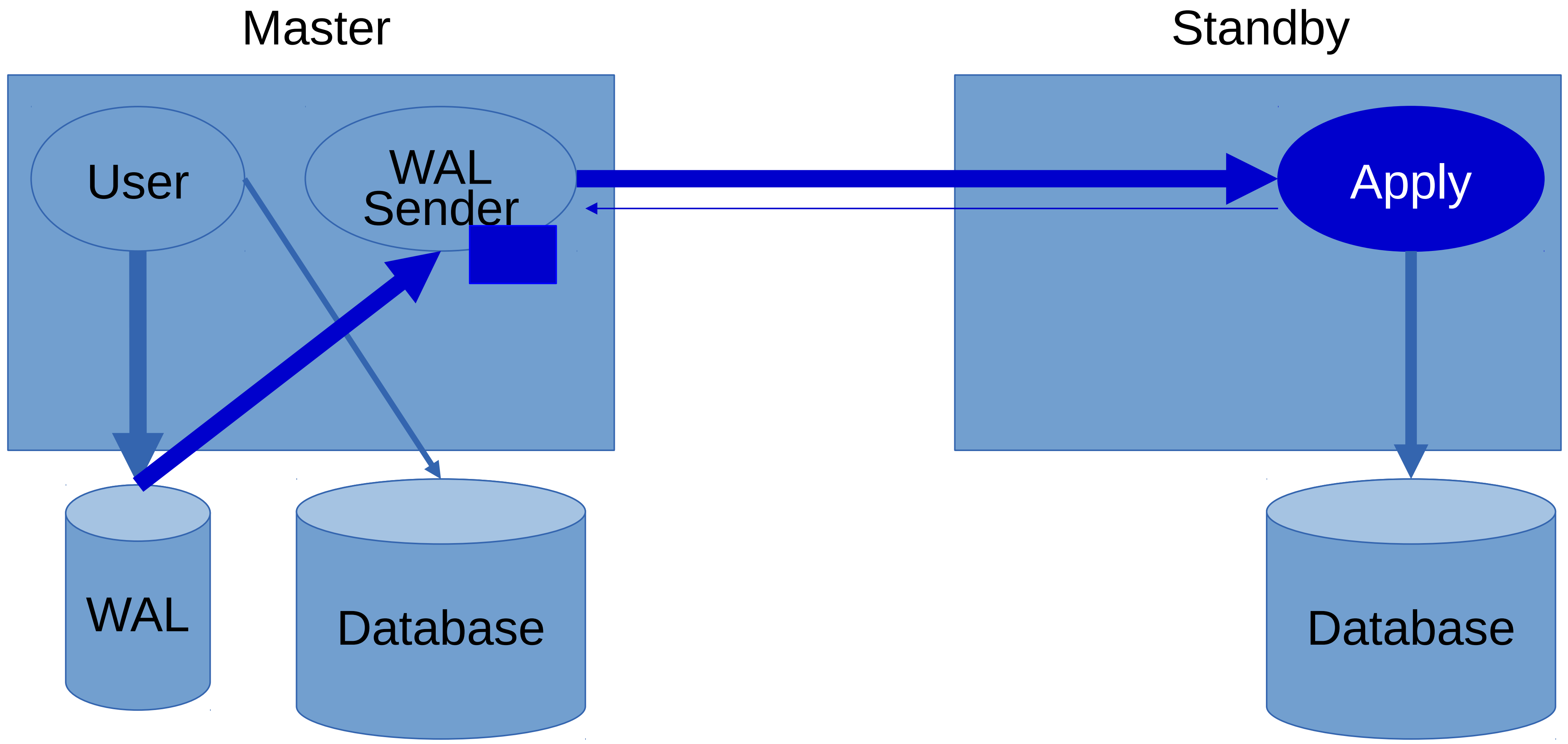
- BDR
  - Modified PostgreSQL 9.4 + Extension
  - Multi-master
  - Transparent DDL
- pglogical
  - Extension for 9.4+
  - One way replication
  - Replacement for trigger-based solutions
  - Integrating into PostgreSQL core

Streaming Replication

Physical Streaming Replication

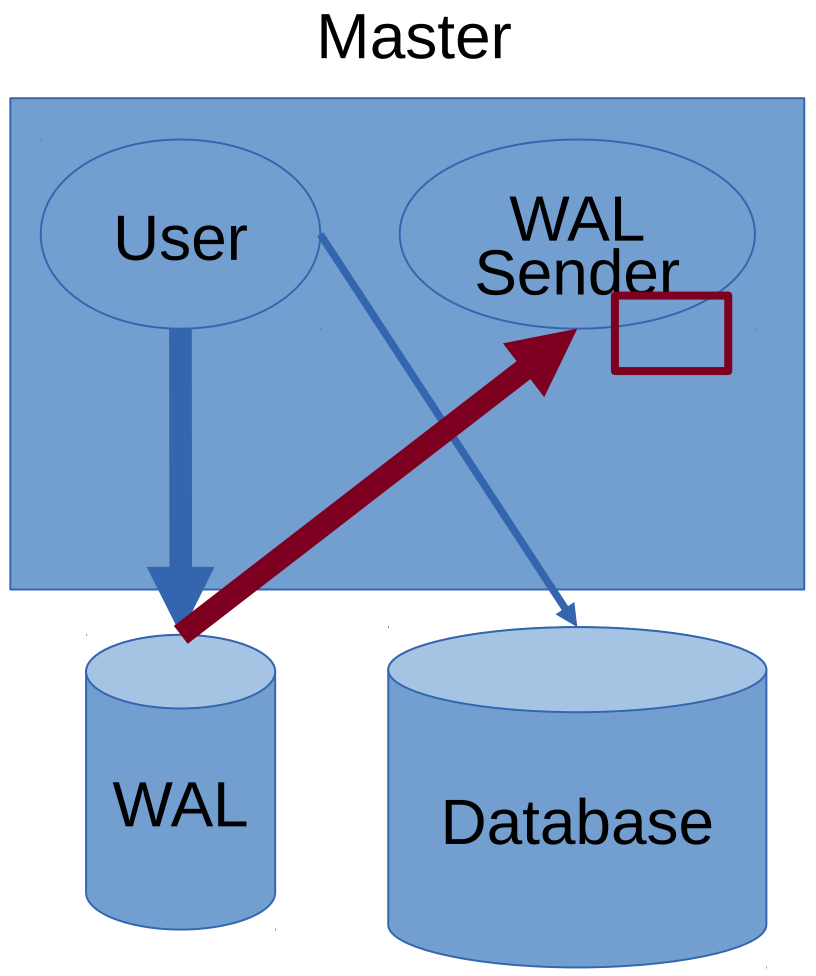


Logical Streaming Replication



Logical Decoding

Logical Decoding



Logical Decoding

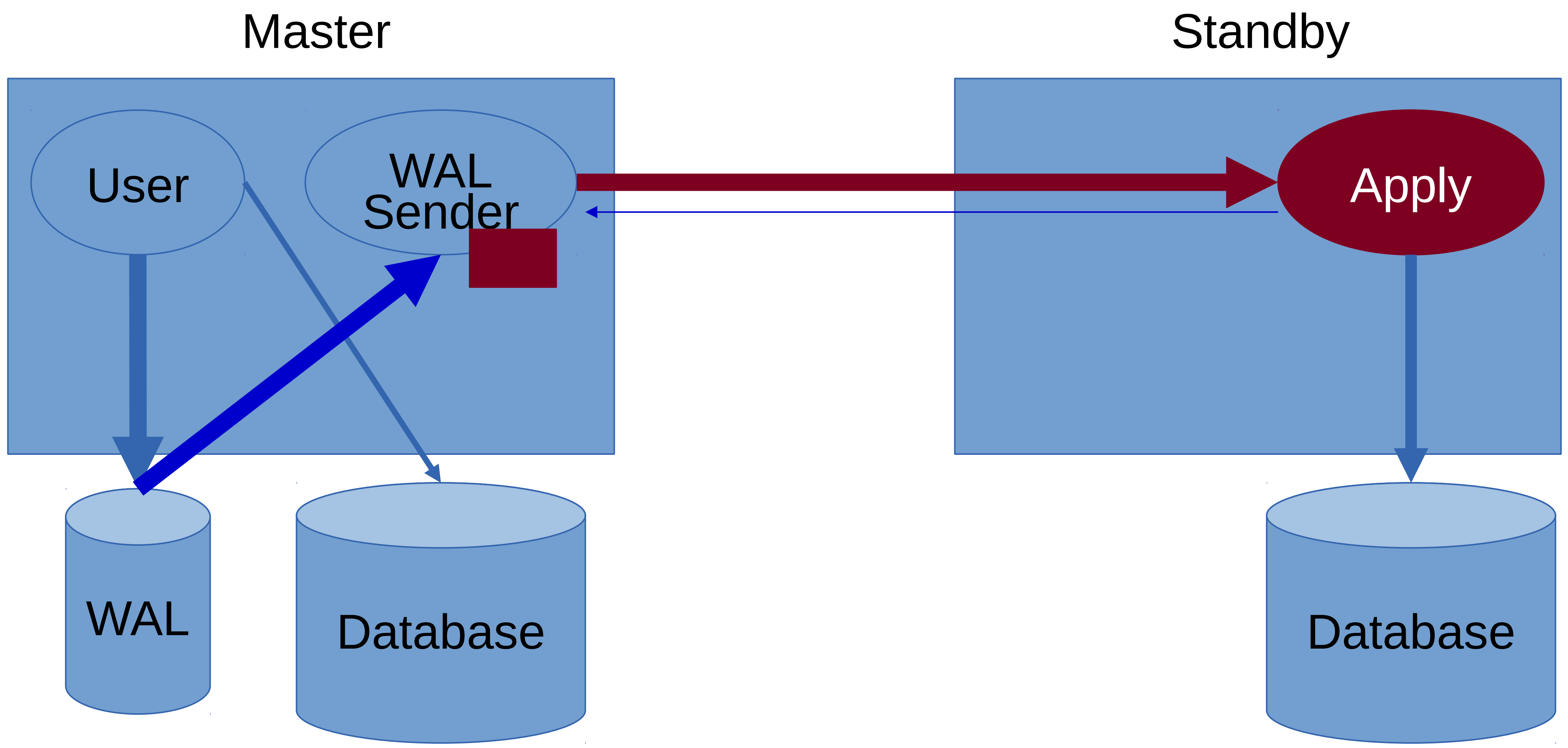
- Extracts information from Write-Ahead Log into logical changes (INSERT/UPDATE/DELETE)
- Per row and commit ordered
- C API for output plugin
- No DDL
- SQL Interface
- Streaming Interface

Existing output plugins

- https://github.com/confluentinc/bottledwater-pg
  - Decodes change into AVRO
  - Sends data to Kafka
- https://github.com/xstevens/decoderbufs
  - Decodes to protocol buffers
- pglogical_output
  - Plugin that pglogical uses
  - “Native” protocol
  - JSON protocol

pglogical

pglogical


Master
Standby
User
WAL Sender
Apply
WAL
Database
Database

Logical Replication

- Target node is writeable
  - Allows temp tables
  - Allows different indexes
  - Allows different security
  - Allows (some) data transformation
- Selective Replication
  - Can replicate subset of database
- Cross-version

pglogical

- Replicates transactions in commit order
- Selective Replication
- Online Upgrade
- Data Transport
  - Data integration
  - Streaming changes to analytical database
  - Master configuration data management
  - …
- Optionally semi-synchronous

Replication Sets

- Replication is defined in terms of groups (sets) of tables, rather than individual tables
  - Need to be defined on each upstream node
- Table is not replicated until added to a set
- Tables may be defined in more than one set, but changes for the table will only be sent **once** to each subscriber
- Predefined sets, “default”, “default_insert_only”, “ddl_sql”

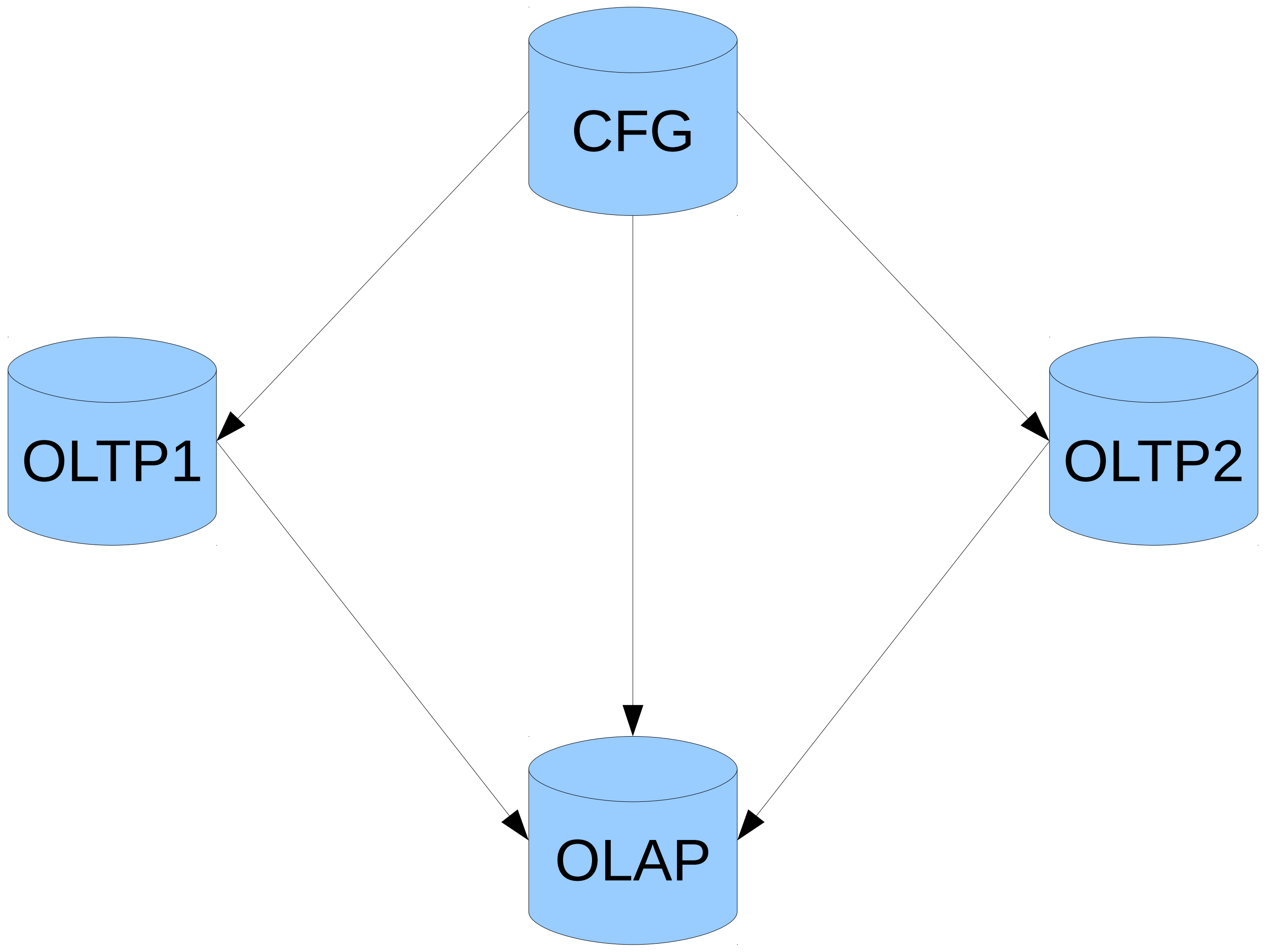
Replication Actions

- By default replication sets replicate all actions
  - INSERT, UPDATE, DELETE, TRUNCATE
- It's possible to filter actions for given replication set
- Useful for data aggregation, data warehousing etc.

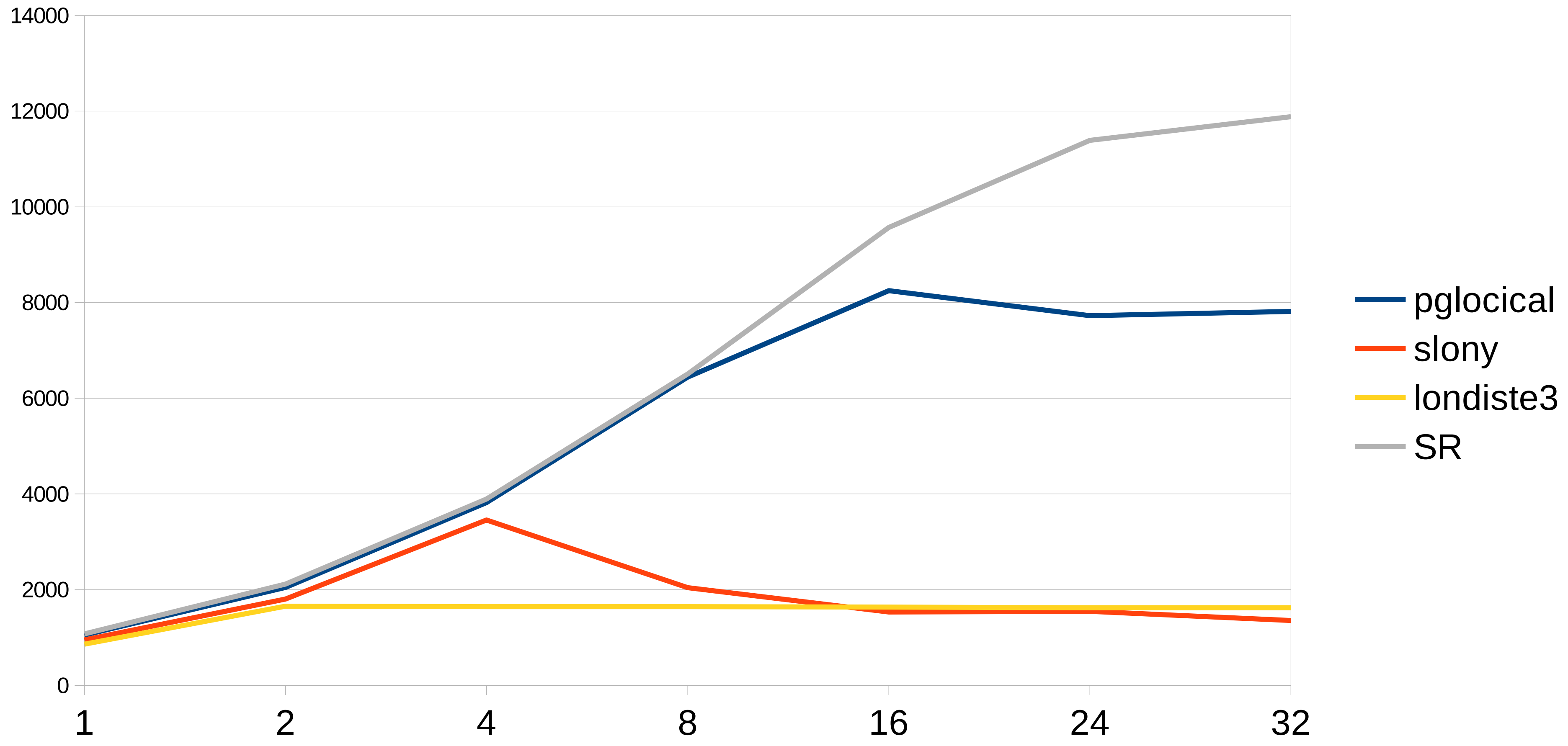
Installation

- Extension
  - All configuration in database
- Provider + Subscriber
  - CREATE EXTENSION pglogical;
  - create_node(name, dsn)
- Provider
  - replication_set_add_table(set_name, table_name)
- Subscriber
  - create_subscription(name, provider_dsn, ...)

Example setup



Performance



DDL Replication

- Initial schema either fully synchronized or not at all
- The DDL commands are not automatically replicated yet
- pglogical.**replicate_ddl_command**(
  command [, replication_sets])

Other caveats

- Sequences not replicated yet
  - Fixed in upcoming 1.1
- Big transactions
- Does not play well with physical replication yet
  - Failover
  - Fixed in PostgreSQL 9.6

Future plans

- Make it available in PostgreSQL out of the box
  - 9.7?
- Data transformation hooks
- Filtering by expression
- Push replication
- Integrate BDR features
  - Multi-master
  - Transparent DDL replication

Questions?

- petr@2ndquadrant.com
- http://2ndquadrant.com/pglogical/