PostgreSQL Replication in 2017

PGDay.RU
St Petersburg, Russia

Magnus Hagander
magnus@hagander.net

Magnus Hagander

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

Replication

"PostgreSQL doesn't have replication"

"PostgreSQL doesn't have replication"

"So we have to use MySQL"

Replication

- Wasn't true back then
- Even less true now!
- Now there are too many choices?
  - But you have to pick one
  - And can be hard to use

Replication

- Can be done at different layers
- From hardware
  - (ish)
- To application

Replication layers

- ↓ Application
- ↓ App-in-database
- ↓ Database logical
- ↓ Database physical
- ↓ Operating system
- ↓ Hardware

Start from the bottom

SAN replication

- Hardware takes care of replication
- Block level
- Transparent to OS
  - And to PostgreSQL
- Common enterprise solution
  - Especially with VMs

SAN replication

- From single rack
- To multi-site
- Synchronous
- Guaranteed to never fail
    - Riiiiight...

Replication layers

↓ Application
↓ App-in-database
↓ Database logical
↓ Database physical
↓ Operating system
↓ Hardware

DRBD

- Similar in style to SAN
- Implementation in OS driver
- Performance?

Replication layers

- ↓ Application
- ↓ App-in-database
- ↓ Database logical
- ↓ Database physical
- ↓ Operating system
- ↓ Hardware

Database physical

- WAL based replication
- File based from 8.3
- Streaming since 9.0
- Synchronous since 9.1
  - Transaction level mixing
- (etc)

```
wal_level = 'replica'
```

Synchronous mode

- off
- local
- on
- remote_apply

Streaming replication

- Primary choice today
- Easy to set up
- Hard to get wrong
- Efficient
- Built-in

Streaming replication

```
$ pg_basebackup -D /var/lib/pgsql \
  -h master -U replica \
  -X stream -R -P \
  -S replica1
$ sudo service postgresql-9.6 start
```

Streaming replication

- Architecture/compile flag dependent
- Whole cluster only
- Standby completely read-only
- Master → standby only
- Excellent for availability

Streaming replication

- No built-in cluster management
  - Manual *or* automatic
  - Provides infrastructure
- No fail-back
  - (no easy one)
- Easy to get started, harder to maintain

Cluster management

Patroni

- Designed for automatic management
- Including automatic failover
- Uses etcd, zookeeper, or consul
- Integrates with haproxy

Cluster management

repmgr

- Fewer pre-requisites
- Easier for manual management
    - Comes with *repmgrd* that does automatic
- Does not handle connection management
    - Use e.g. rebouncer
    - Or haproxy

Replication layers

↓ Application

↓ App-in-database

↓ Database logical

↓ Database physical

↓ Operating system

↓ Hardware

Database logical

- Logical decoding since 9.4
- Logical replication since 10
    - Built-in, that is
- Piggy-backs on WAL
- Very low overhead

```
wal_level = 'logical'
```

Logical replication

- Reconstructs changes by row
- Replicates row content
  - *not* SQL statements
- Fully transactional

Logical replication

- Table-level partial replication
- Table-level bi-directional replication

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;
```

Logical replication

- Data replication only
  - No schema
  - No sequences
- Suitable for data distribution
- But not for HA
- Lacks failover slots!

pglogical

- External version of logical replication
- Merged piece by piece
- More capabilities!
- Not as deeply integrated

pglogical

- Sequence replication
- Row based filtering
- Column based filtering
- Merging and conflict resolution
- …

pglogical

- Supports PostgreSQL 9.4
- Zero (or close to zero) downtime upgrades!

Replication layers

- ↓ Application
- ↓ App-in-database
- ↓ Database logical
- ↓ Database physical
- ↓ Operating system
- ↓ Hardware

App-in-database

- Trigger based systems
    - Slony
    - Bucardo
    - Londiste
    - ...

Trigger based

- For a long time the only choice
- Now *mostly* superseded
- Much higher overhead than logical
- Complex scenarios

Replication layers

↓ Application

↓ App-in-database

↓ Database logical

↓ Database physical

↓ Operating system

↓ Hardware

Application

- Replication done entirely in application
- Very difficult for transactional
- Useful in limited cases

Summary

High Availability

- Use streaming replication
- Mix of sync and async
- Consider patroni or repmgr

Data distribution

- Logical replication in 10
- pglogical in 9.4+
    - Or in 10 if built-in is not enough
- Upgrade away from your Slony...

Need both?

- Use both!

Thank you!

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