100k Appliances with PostgreSQL

Christoph Mönch-Tegeder

Sophos Network Security Group

pgconf.EU 2013
Background

- Firewall Appliances, „UTM“
- 100,000 appliances, operated by customers and partners
- sized from „cigar box“ to 2 HE 16 cores, VMWare, ...
- „We sell firewalls, not databases.“
- optional: HA/cluster, up to 10 nodes
- historical data model (we used SQLite a few years ago)
PostgreSQL

- **was**: PostgreSQL 8.4 and Slony
  - misses features, performance
  - schema changes are hard under Slony
- **now**: PostgreSQL 9.2 and streaming replication
  - convert old databases
  - replication has to be integrated with cluster technology
Upgrade PostgreSQL

- at boot time, before anything uses the databases
- detect old $DATADIR/PG_VERSION
- remove Slony
- run `pg_upgrade`, delete old $DATADIR on success
- keep logfile!
- when things go wrong: sort it out via support
Replication

- 1 primary node, up to 9 secondaries
- load distribution for firewall services
- secondaries can be added on the fly
- new nodes are auto-configured and auto-updated
- database replication has to work with our clustering code
- mode changes:
  - secondary becomes primary (promote)
  - primary becomes secondary (downgrade)
  - secondary gets new primary (topology change)
Replication (2)

- uses rsync for initial clone
- dedicated network for replication, use rsync daemon
- no parallel cloning: pg_advisory_lock()
- start streaming replication
- monitoring restarts whole process if necessary
- status communication via NOTIFY
- small: < 3 kLOC perl
Schema Changes

- Slony: EXECUTE SCRIPT, problematic with dynamic cluster
- alternative: break replication, change schema, restart slony
- now: take full copy of cluster, and DDL is replicated, too
- in any case: sync schema changes with application changes
- forces reboot, HA will prevent network outage
How to Modify the Schema

- generating the update DDL
  ```
  sed '1 i BEGIN;
  /^CREATE TABLE foo \/,\/;/ p;
  \$ a COMMIT;\n  /^COMMIT/!d;' < our_schema_defs.sql
  | psql -f - database
  ```

- when you need ALTER TABLE
  ```
  ( echo 'BEGIN;'
  echo 'ALTER TABLE foo DROP COLUMN bar;'
  sed ' /^CREATE TABLE baz \/,\/;/ p; d;' < our_schema_defs.sql
  echo 'COMMIT;' ) \
  | psql -f - database
  ```
Converting Data

- takes too long for boot sequence
- use cron, starts job as often as necessary
- compare objects (`pg_class`, `pg_get_indexdef()`)
- apply changes (safe to interrupt)
- cronjob removes itself when done
Thanks!