

Bucardo

Postgres replication system

- Multi-master**
- Master-slave**
- Asynchronous**

bucardo.org

Greg Sabino Mullane
End Point Corporation

- **Bucardo History**
- Backcountry.com multi-master need
- Existing system (Bucardo 1) to push changes
- Very asynchronous
- Slaves with different tables
- Other places now using it (bytea)

- **Bucardo Requirements**
- Perl
- Moose
- Pl/Perl, Pl/pgsql (swap and pushdelta only)
- Postgres 8 (8.3 is best)
 - HOT, replica hook, performance

- **Bucardo Shortcomings**

- Not widely used
- No smooth DDL handling
- Not for failover
- Bloat: pg_listener, q, bucardo_delta, bucardo_track
- One developer
- No locking of slaves

- **Bucardo Strengths**
- Perl
- Drop in, no changes needed at all
- Interface
- Hooks!
- Multi-master
- Async to the extreme
- No locking of slaves

- **Bucardo setup**
- Add the schema somewhere
- Add databases to db table
- Add tables to goat table
- Herds, dbggroup
- Create a sync
- Automatic validation, creation.

- **Bucardo Flow**
- Triggers
 - row-level: bucardo_delta
 - Statement-level: NOTIFY
- bucardo_delta: ID, txn_time. Insert only!
- bucardo_track: who's done what?
- MCP, CTL, KID
- NOTIFY vs. timeout vs. kick.

- **Bucardo Flow**
- Join bucardo_delta with actual table. DISTINCT.
- Both sides.
- Rules, triggers, hooks, upsert, conflicts, exceptions
- Update bucardo_track.

- **Bucardo Flow**
- NOTIFY (timeout, kick, trigger)
- MCP -> CTL
- CTL -> q table. Limits per targetdb, sourcedb
- KID -> q table. Keepalive

- **Bucardo Syncs**
- Fullcopy. Timeout/kick. truncate/COPY/done
- Pushdelta. Trigger. One way master-slave.
- Swap. Trigger. Two-way. Conflict resolution.

- **Bucardo Anti-trigger**
- ALTER TABLE DISABLE TRIGGER ALL;
- UPDATE pg_class..
- SET session_replication_role = 'replica';
 - Thanks Jan!

- **Bucardo Hooks**
- Customcode
- Conflict handlers
- Exception - try again
- Pre and post trigger drop
- Row information - href
- Live changes
- Returned value - bitmapped (both!)
- Affect the replication event
- Example: cache invalidation

- **Bucardo Interface**

- `./bucardo_ctl`
- `./bucardo_ctl stop "Adding new table - Greg"`
- `./bucardo_ctl start "Adding new table - Greg"`
- (logged + email)
- `./bucardo_ctl status foobar`
- `./bucardo_ctl add table sales sync=orders`
- `./bucardo_ctl kick foobar 0`

- **Bucardo Interface**

- `./bucardo_ctl kick foobar 0`

- `[0s] ...`

- `[1s] alpha[1s] ...`

- `[2s] alpha[1s] gamma [2s] ...`

- `[4s] alpha [1s] gamma [2s] beta [4s] DONE!`

- **Bucardo Stuff**
- Customselect (fullcopy)
 - `SELECT max(col1), col2 FROM foo GROUP BY 2;`
- Email alerts
- Logging++
- Serialization, deadlocks.
- Cleanup: vacuum, purge_delta

- **DBIx::Safe**
- Like Safe.pm, but for database handles.
- Define what it can and cannot do.
- No commit, no rollback.
- Deep view 'read-only'
- Regex-based control.

- **Bucardo Development**
- Jan idea: supercopy
- Greg idea: supertriggers
 - Automatic replication of new tables
- git.postgresql.org
- Win32 (fork, setsid)
- DDL and failover
- Master-master-master...
- Interface!
- Release early, release often

- **Bucardo Questions?**
- Mailing list at bucardo.org: regular and announce
- greg@endpoint.com