Evolving towards Postgres

David Steele
Crunchy Data

PostgresOpen
September 13, 2019

Agenda

About the Speaker

- Principal Architect at Crunchy Data, the Trusted Open Source Enterprise PostgreSQL Leader.
- Actively developing with PostgreSQL since 1999.
- Primary author of pgBackRest and co-author of pgAudit.
- PostgreSQL Contributor.

What is pgBackRest?

pgBackRest aims to be a simple, reliable backup and restore system that can seamlessly scale up to the largest databases and workloads.

What is the PostgreSQL Ecosystem?

The PostgreSQL ecosystem describes the software products and services that use or work with PostgreSQL.

Some examples:

- pgAdmin - Graphical administration
- PostGIS - GIS functions
- check_postgres - Monitoring plugin for Nagios
- pg_partman - Automated partitioning
- PgBouncer - Connection pooler
- pgBackRest - Backup & restore

And many, many more.

Why aren't they built into PostgreSQL?

There are many possible approaches to a single problem, so the general idea is to have multiple solutions, each tailored to a specific approach.

This is predicted by Conway's Law, i.e. the community is free to pursue their own ideas, therefore there will be mutliple solutions.

Over time the best ideas can be integrated into PostgreSQL core, though this process is slow and uncertain.

Ignoring PostgreSQL

It is possible to develop software in the PostgreSQL ecosystem without ever interacting with the PostgreSQL community.

The pgBackRest project did this for several years.

For many projects this model may work for a long time or indefinitely.

Contributing to PostgreSQL

So why did we start contributing to PostgreSQL?

Reviewing Useful Features

```
- Implement backup API functions for non-exclusive backups
Magnus Hagander, PostgreSQL 9.6
```

This patch allowed any external tool to run a non-exclusive backup, which had previously only been possible for the internal `pg_basebackup` tool.

A non-exclusive backup does not write `backup_label` into `$PGDATA` and does not interfere with other backups in progress.

Getting Feedback on pgBackRest Features

```
- Exclude additional directories in pg_basebackup
- Exclude pg_internal.init from BASE_BACKUP
- Exclude unlogged tables from base backups
- Exclude temp tables from base backups
David Steele, PostgreSQL 10-11
```

These exclusions were all researched and developed for pgBackRest originally. By contributing them to `pg_basebackup` we were able to get a lot of review of the changes.

Adding a New Feature

```
- Allow group access on PGDATA
David Steele, PostgreSQL 11
```

Allows `$PGDATA` to have group read permissions so backup can be performed by an unprivileged user. This was not workable without changes to core.

This commit depended on:

```
- Refactor dir/file permissions
David Steele and Adam Brightwell, PostgreSQL 11

- Refactor new file permission handling
David Steele, PostgreSQL 11
```

Choosing a Language

pgBackRest was originally written in Perl but we are migrating to C. Why?

- Speed (mostly startup speed)
- Simplified deployment
- Ability to use code and data structures from PostgreSQL
  - Page Checkums
  - pg_control structures
  - WAL structures
  - Portability

Challenge - Removing Exclusive Backup

There are a number of reasons why using exclusive backup is not a good idea, but there is a lot of resistance to removing it. A better alternative (non-exclusive mode) has been available since 9.6.

Opinions on the patch fell primarily into three (nearly equal) groups:

- No – it would break too many existing scripts.
- Yes – but not now.
- Yes – we should do this now because a better method has been available for some time.

Challenge - Improving Backup Documentation

In many cases the PostgreSQL backup/archiving documentation is over-simplistic or simply gives bad advice, i.e. using `cp` for archiving or using exclusive mode backup.

Updating this documentation with better instructions and discouraging users from using the deprecated exclusive mode backup has been a multi-year process.

```
- Improve low-level backup documentation.
David Steele, PostgreSQL 11

- Warn more strongly about the dangers of exclusive backup mode.
David Steele and Robert Haas, PostgreSQL 12
```

Conclusion

Is it worth it?

Yes. Working with the community can be challenging but there are many benefits to both projects.

Questions?

website: `http://www.pgbackrest.org`

email: david@pgbackrest.org
email: david@crunchydata.com

releases: `https://github.com/pgbackrest/pgbackrest/releases`

slides: `https://github.com/dwsteele/conference/releases`