pgBackRest Overview

David Steele
Crunchy Data

Crunchy Storm
November 28, 2018

CRUNCHY Enterprise PostgreSQL

About David

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

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 pgBackRest?

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

Performance

pgBackRest has a strong emphasis on performance:

- Parallel/asynchronous operation for all core commands
- Backup from Standby
- Advanced and flexible configuration for tuning specific commands
- Delta restore mode using checksums

Performance

pgBackRest has a strong emphasis on performance:

- Parallel/asynchronous operation for all core commands
- Backup from Standby
- Advanced and flexible configuration for tuning specific commands
- Delta restore mode using checksums

Performance

pgBackRest has a strong emphasis on performance:

- Parallel/asynchronous operation for all core commands
- Backup from Standby
- Advanced and flexible configuration for tuning specific commands
- Delta restore mode using checksums

Performance

pgBackRest has a strong emphasis on performance:

- Parallel/asynchronous operation for all core commands
- Backup from Standby
- Advanced and flexible configuration for tuning specific commands
- Delta restore mode using checksums

Reliability

pgBackRest strives to avoid corruption and misconfiguration:

- All files/manifests are checksummed
- Checksums are tested on every operation
- Version and system ID are used to avoid crossing repositories
- Archive command that de-dupes and does not overwrite existing WAL

Code Coverage

pgBackRest has unparalleled code coverage:

- 99.6% line/branch/condition coverage in the C code (unit tests)
- 93.1% line/branch/condition coverage in the Perl code (unit/integration tests)
- Each uncovered line/branch/condition is documented with the reason it is not covered
- New C patches must have 100% coverage (including documented exceptions) to be accepted

The Future

pgBackRest is currently being migrated to C and code coverage is being increased to 100% (including documented exceptions).

Some future features under consideration:

- Protocol layer over SSL (no SSH required)
- Page-level incremental
- Windows support

Questions?

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

email: david@crunchydata.com

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