PostgreSQL Replikation

Bernd Helmle, bernd.helmle@credativ.de

11. November 2011

credativ

Warum Replikation?

- Hochverfügbarkeit
- Lastverteilung
- Testsysteme
- Reporting

Am Anfang war...

- Viele konkurrierende Replikationssysteme
- Viele Kompromisse
- Externe Projekte, Eigene Releasezyklen
- Erfordern Interaktion mit dem Backend

Alte Projekte

- Rserv / eRserver
- dbmirror
- Sequoia (CJDBC)

Aktuelle Projekte

Slony-I

- Eventbasiert
- Logtrigger/Denytrigger
- “In Place” Subscription
- Asynchronous Single Master

`http://www.slony.info`

Aktuelle Projekte

Londiste

- Queuebasiert (PgQ)
- Provider/Subscriber Konzept
- Triggerbasiert
- Asynchronous Single Master

`http://pgfoundry.org/projects/skytools/`

Aktuelle Projekte

Bucardo

- Asynchronous Multi/Single Master
- Triggerbasiert
- Bedient sich mittels NOTIFY/LISTEN
- Benutzerdefinierte Konfliktbehandlung

`http://bucardo.org/wiki/Bucardo/Documentation/Overview`

Logshipping

- PostgreSQL 8.1
- Flexible Archivierung

```
archive_command = 'cp -i %p /pg-archive/%f </bin/false'
```

- Online Basissicherungen

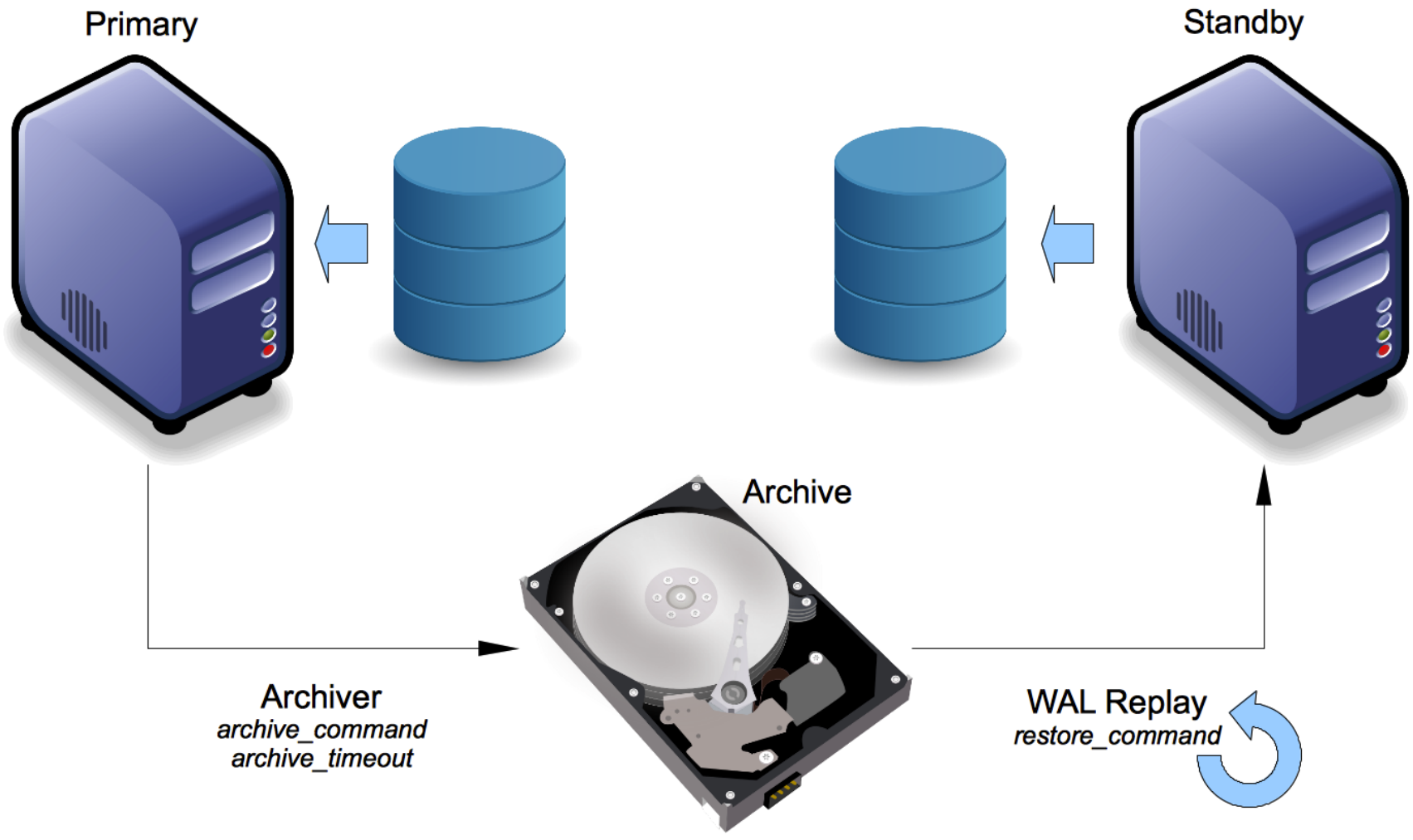
```
=# SELECT pg_start_backup('Basisicherung');
...
=# SELECT pg_stop_backup();
```

- `recovery.conf`

```
restore_command = 'cp -i /pg-archive/%f %p </bin/false'
```

- Inkrementelle Backup
- PITR (Point In Time Recovery)
- Warm Standby

Warm Standby



Strategieänderung

"[...] But it is time to include a simple, reliable basic replication feature in the core system."

Tom Lane 2008,
`http://archives.postgresql.org/pgsql-hackers/2008-05/msg00913.php`

Hot Standby

`hot_standby = on`

- Einer der innovativsten Patches
- Lesende Abfragen auf Standby
- Kein DDL, DML, Two-phase commit (2PC)
- `SHOW transaction_read_only`

Probleme

- VACUUM
- Konfliktbehaftete Abfragen
- Abwägen Replication Delay/Aktualität

```
max_standby_archive_delay = 30s
max_standby_streaming_delay = 30s
vacuum_defer_cleanup_age = 1024
```

Logshipping - Diskussionspunkte

- Latenz
- `archive_timeout`
- Methodik (rsync, cp, NFS, ...)
- Komprimieren von XLOG Dateien
- Zusätzliche Tools: SkyTools/walmgr, OmniPITR, ...

Streaming Replication

Idee:

- “Real Time Log Shipping”
- Basierend auf Transaktionslog
- Flexibles Protokoll
- Skalierbar
- Einfache Konfiguration

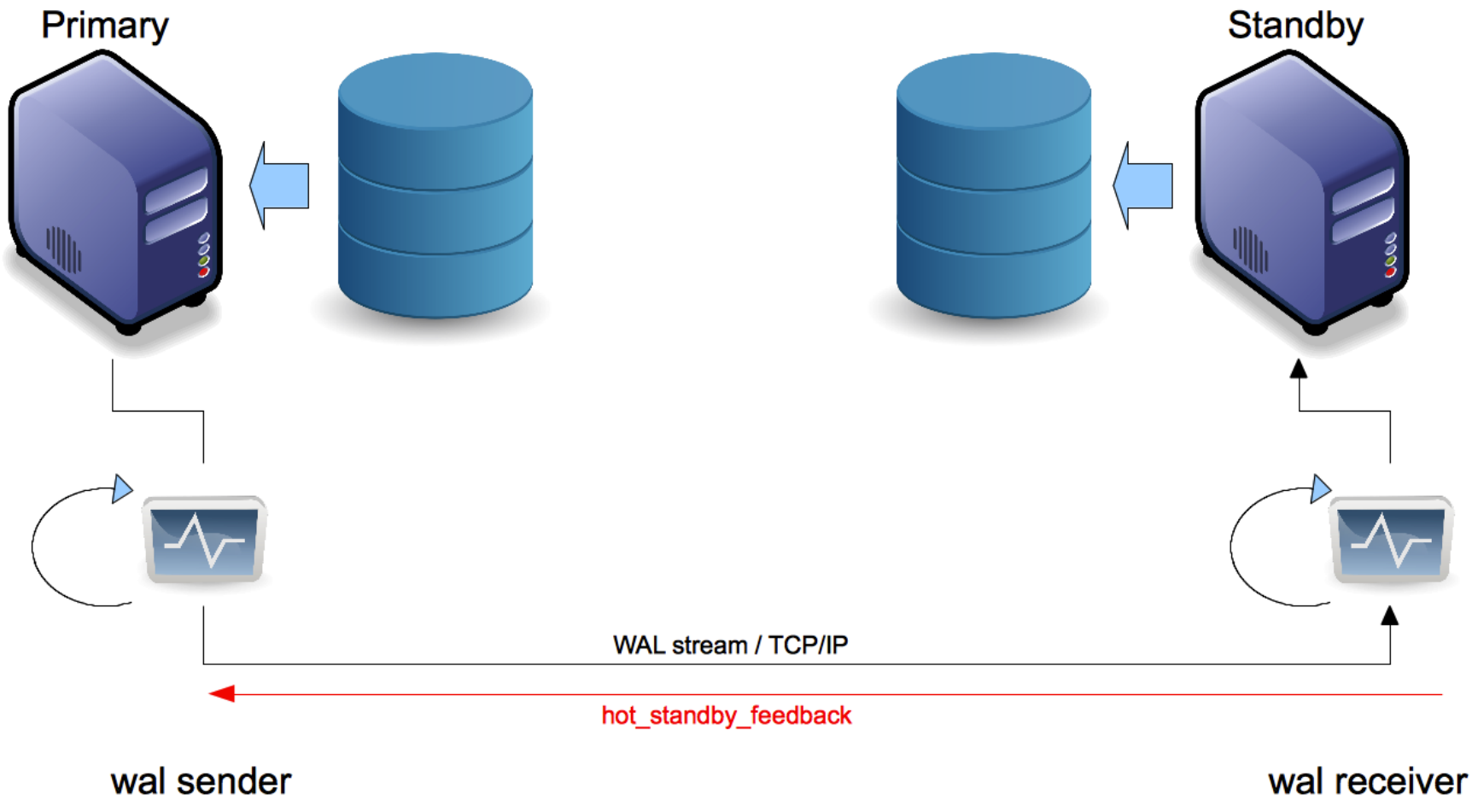
```
# Master
max_wal_sender = 1

# Standby recovery.conf
primary_conninfo='dbname=db user=postgres'
standby_mode = 'on'
```

Streaming Replication

- Separate Datenbankprozesse übernehmen Replikation
- Primary: `wal sender process`
- Streaming Standby: `wal receiver process`
- Flexibles, einfaches Kommunikations-Protokoll
- libpq (libpqwalreceiver)
- Erweiterte Funktionen in 9.1

Streaming Replication



Synchrone Replikation

- Ab PostgreSQL 9.1
- WAL Receiver bestätigt geschriebenen XLOG COMMIT record
- Standby: `application_name`
- Primary: `synchronous_standby_names`
- Ein synchroner Standby, mehrere potentielle

```
synchronous_commit = on
synchronous_standby_names='standby1, standby2, standby3'
```

Synchrone Replikation

- Transaktionskontrolle
- Verlagern in Applikationslogik
- Performance

```
bhe=# BEGIN;
BEGIN
bhe=# SET LOCAL synchronous_commit TO on;
SET
bhe=# -- WICHTIGE TRANSAKTION
...
bhe=# COMMIT;
COMMIT
```

Synchrone Replikation

Sorgfältige Planung erforderlich:

- Latenz!
- Timeouts: `replication_timeout`, TCP/IP Keepalive Einstellungen
- `hot_standby_feedback`
- `wal_receiver_status_interval`
- Hochverfügbarkeit

Erweitertes Monitoring - Status

```
    bhe=# SELECT * FROM pg_stat_replication ;
-[ RECORD 1 ]----+------------------------------
procpid          | 6165
usesysid         | 16385
usename          | replication
application_name | standby1
client_addr      | 192.168.0.58
client_hostname  |
client_port      | 36458
backend_start    | 2011-11-10 16:11:16.876693+00
state            | streaming
sent_location    | 0/232C7E88
write_location   | 0/232C7E88
flush_location   | 0/232C7E88
replay_location  | 0/232C7E88
sync_priority    | 1
sync_state       | sync
```

Erweitertes Monitoring - Konflikte

```
bhe=# SELECT * FROM pg_stat_database_conflicts WHERE datname = 'bhe';
-[ RECORD 1 ]----+------
datid            | 16384
datname          | db
confl_tablespace | 0
confl_lock       | 0
confl_snapshot   | 0
confl_bufferpin  | 0
confl_deadlock   | 0
```

Streaming Basebackups

- Erweiterung im Streaming Protokoll
- Erlaubt Streaming Online Backups
- Einschliesslich XLOG (`--xlog`)
- `tar` Format oder `plain` (`--format=plain|tar`)
- Kompression für `tar` Format

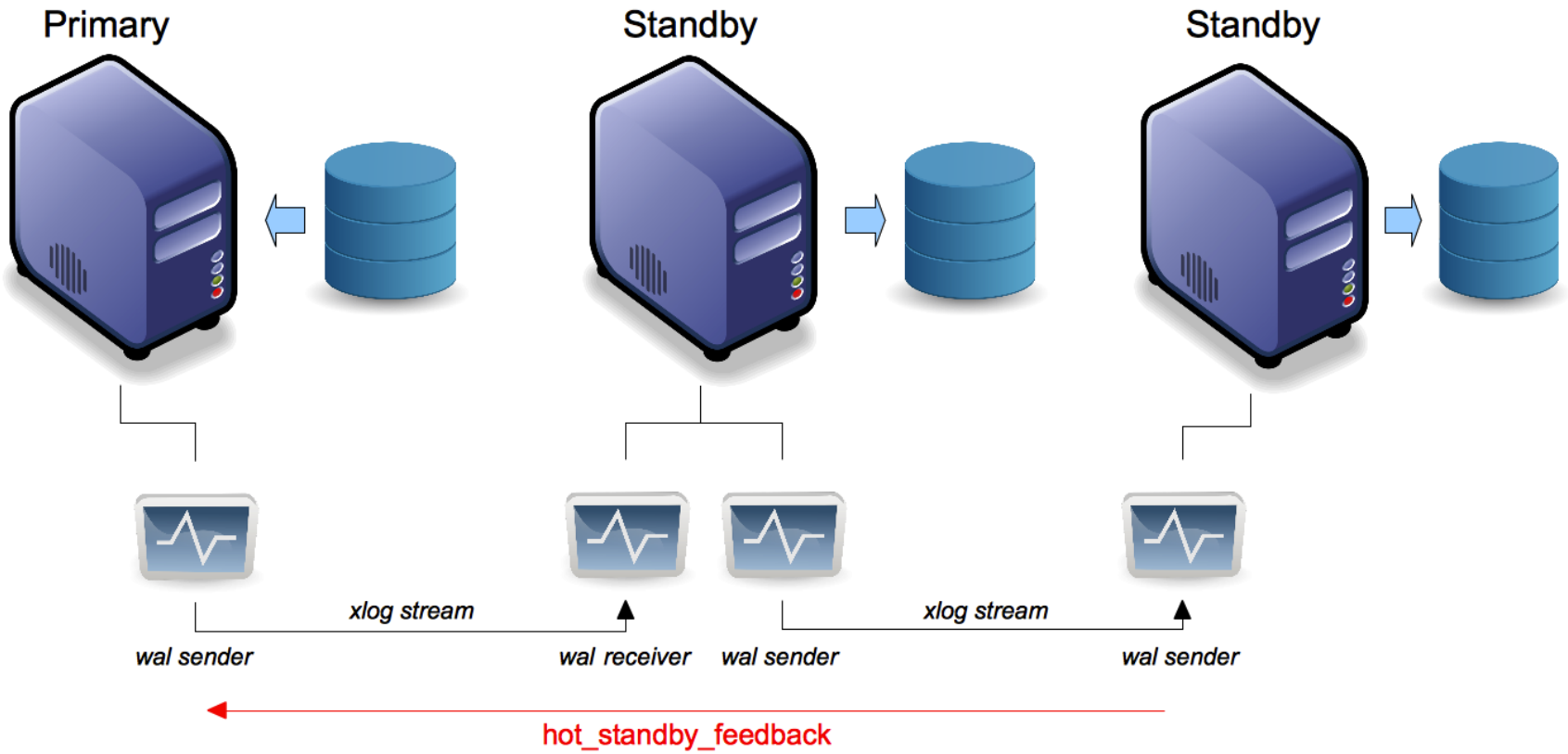
```
# pg_basebackup -x -P -c spread -l 'Backup' -P -h 192.168.0.60 \
    -U backup -D /pg_archive/data
```

Kontrollierte Replikation

Standby kann Replikation kontrolliert anhalten und fortsetzen

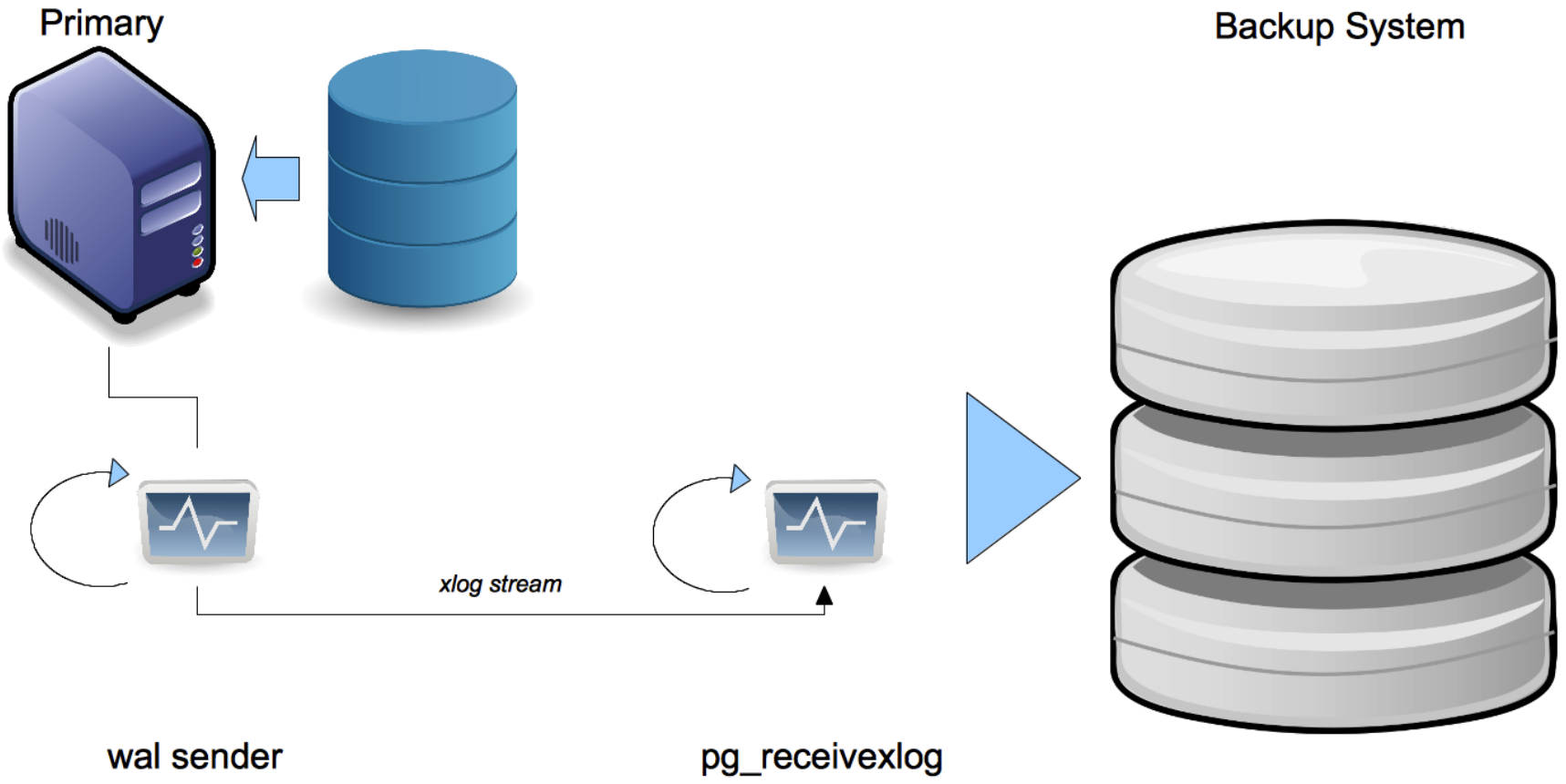
- `pg_xlog_replay_pause()`
- `pg_xlog_replay_resume()`
- `pg_is_xlog_replay_paused()`

Ausblick 9.2: Kaskadierende Replikation



Ausblick 9.2: pg_receivexlog

```
# pg_receivexlog -h db_primary -U backup -D /archive/db/xlog
```



Dankeschön

PQfinish(lecture);

Feedback:
`http://www.postgresql.eu/events/feedback/pgconfde2011`

credativ sucht Verstärkung!

info@credativ.de
http://www.credativ.de