Advances in High Availability for PostgreSQL in the Enterprise

John Dalton
Senior Director, Product Management

August 25, 2021

EDB™

What you will learn today

1. Top-tier enterprise applications can run with confidence using Always On Postgres clusters
2. The latest Postgres-BDR^{TM} release delivers faster throughput, larger clusters, and better observability
3. Applications requiring Oracle SQL compatibility and 99.999% availability can run on Postgres

Agenda

- Enterprise needs for HA Postgres
- Faster throughput
- Larger clusters for data distribution
- Better observability
- HA for Oracle SQL compatible applications
- A look ahead

What is “Always On?”

Delivering mission-critical applications/services 24x7

Finance

Payments

Bank account access

Telecom

Video conferencing

Texting/alerting

Transportation

Travel

Rideshare

GPS

Do you need to be “Always On”?

Things to consider

What’s the reputation cost of downtime to your business?

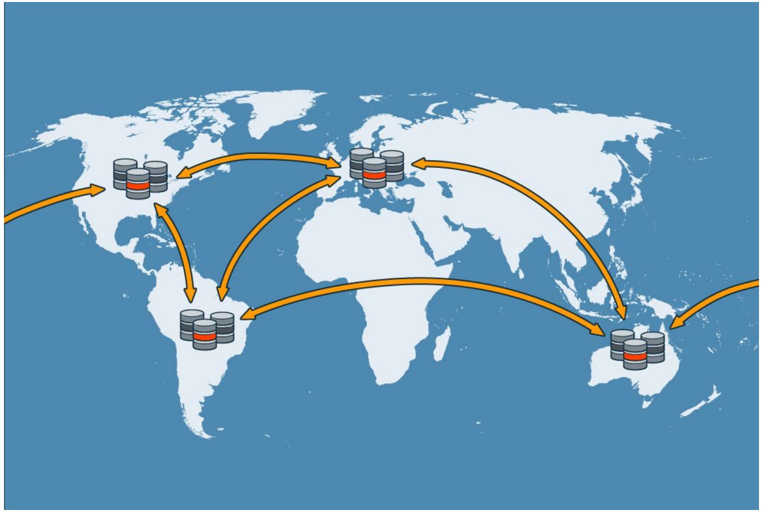
Are there times when it’s okay for an application to be inaccessible?

Is access to data tied directly to revenue?

Are your customers globally distributed?

Postgres-BDR is more than bi-directional replication

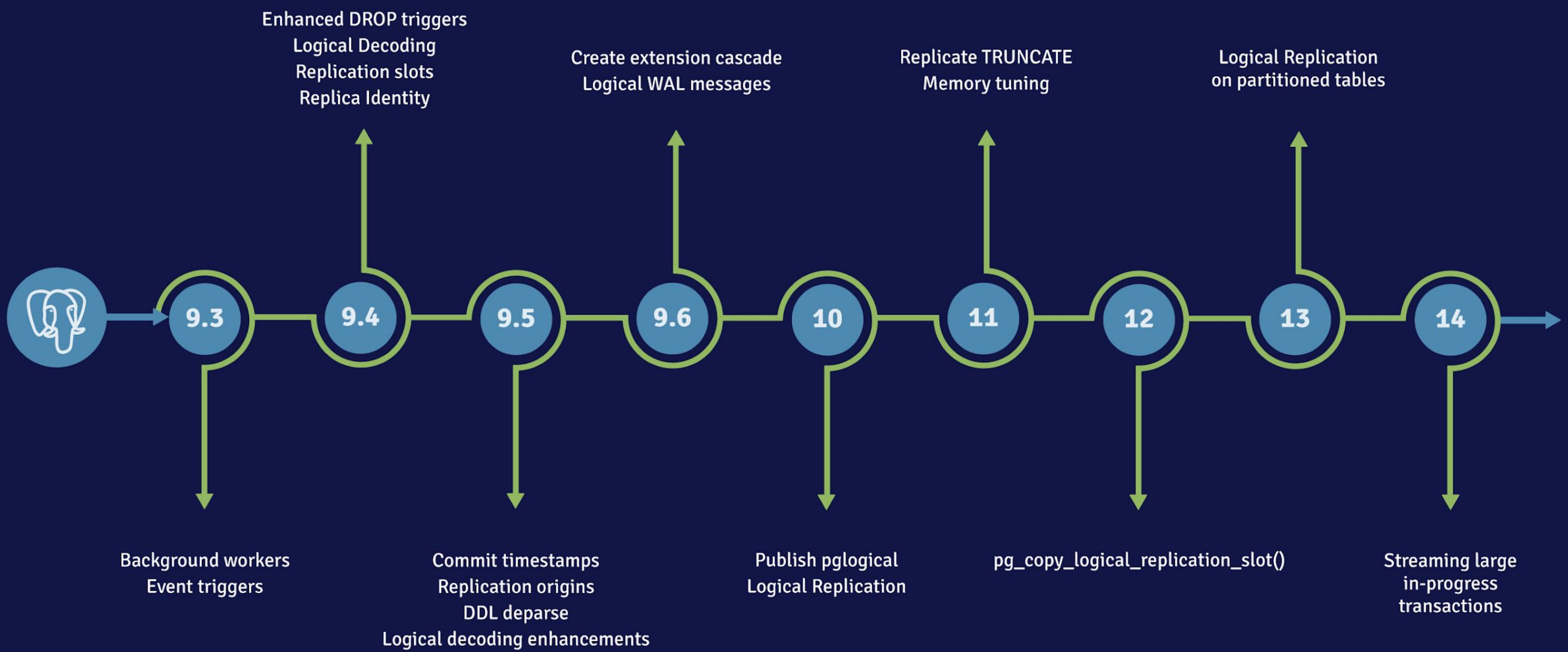
Multi-master replication enabling highly available and geographically distributed Postgres clusters



- Logical replication of data and schema enabled via standard PostgreSQL extension
- Data consistency options that span from immediate to eventual consistency
- Robust tooling to manage conflicts, monitor performance, and validate consistency
- Deploy natively to cloud, virtual, or bare metal environments

Postgres **BDR**

BDR project contributions to PostgreSQL



9.3: Background workers
Event triggers

9.4: Enhanced DROP triggers
Logical Decoding
Replication slots
Replica Identity

9.5: Commit timestamps
Replication origins
DDL deparse
Logical decoding enhancements

9.6: Create extension cascade
Logical WAL messages

10: Publish pglogical
Logical Replication

11: Replicate TRUNCATE
Memory tuning

12: pg_copy_logical_replication_slot()

13: Logical Replication
on partitioned tables

14: Streaming large
in-progress
transactions

That’s great, but enterprises are asking for more

And maintaining data consistency is paramount

Faster Throughput

I need to push more transactions (TPS) through the cluster, as fast as Postgres can run

Larger Clusters

My geographically distributed application needs a read scalable cluster without increased overhead

Better Observability

How do I ensure the right data is exposed to quickly triage issues and facilitate recovery actions

More Migrations

I want to move my applications that need 99.999% availability from Oracle to Postgres

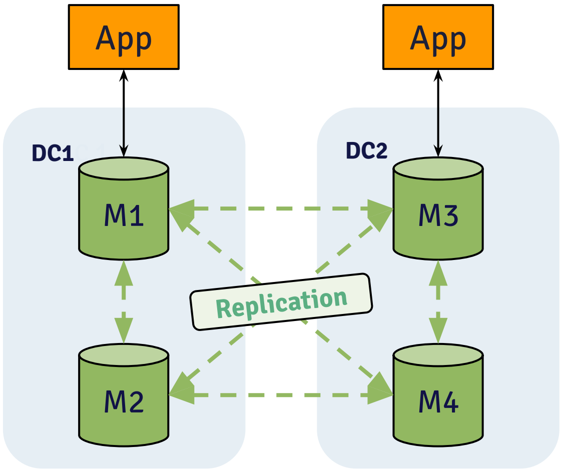
Faster Throughput

Overview

Introducing parallel data flow architecture to BDR

Goals of pipeline parallelism

- Smoother latency
- Less CPU overhead on BDR upstream nodes
- Better overall throughput for parallelizable workloads



Single decoding worker [upstream]

Single decoding worker improves performance on upstream node by doing logical decoding of WAL once instead of for each downstream node

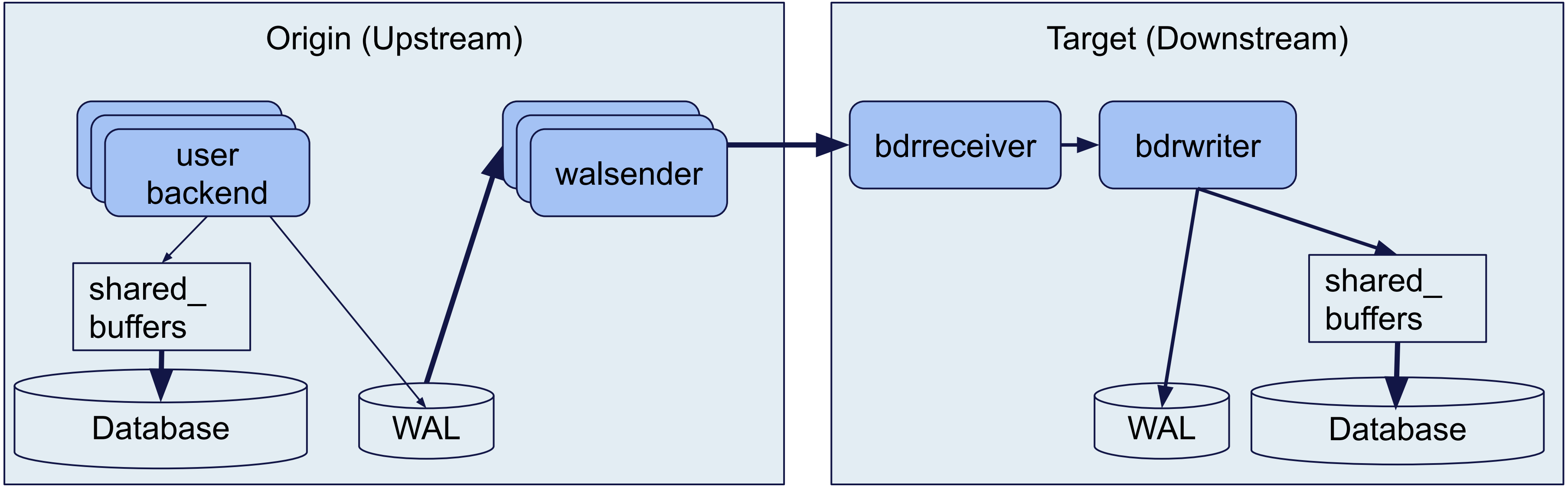
Currently not enabled by default, expected to quickly mature

Parallel apply [downstream]

Parallel apply allows multiple writer processes to apply transactions on downstream node with throughput up to 5X faster

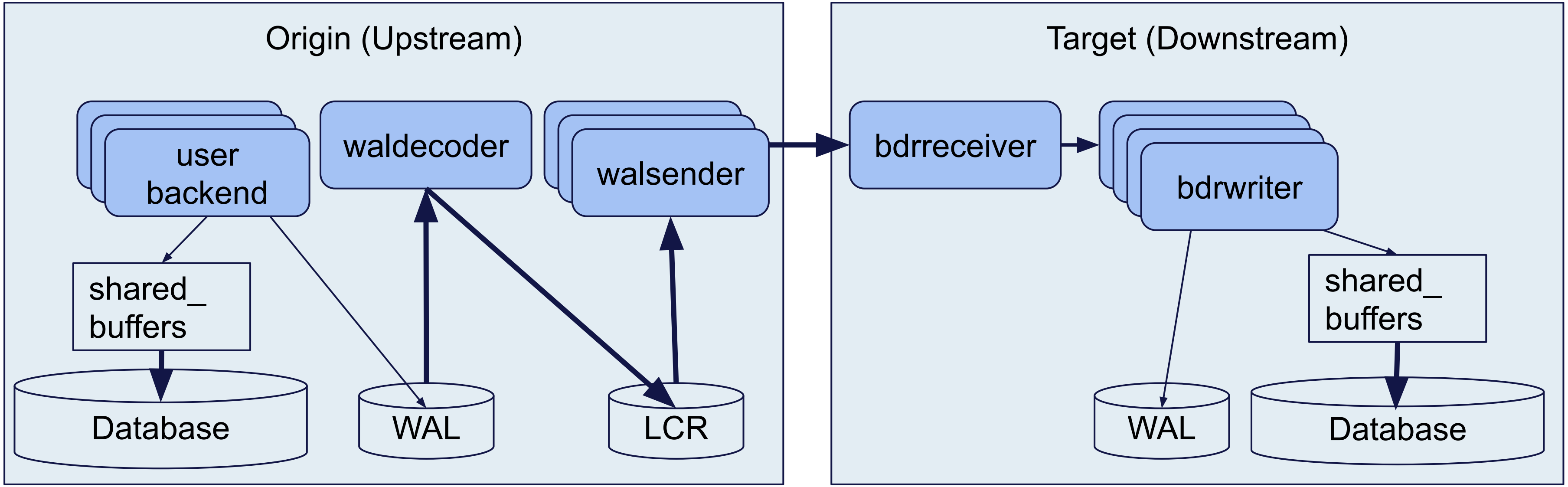
Benefits mixed write workloads, especially for larger DBs and/or i/o heavy write workloads

Logical streaming replication in 3.6



Parallel logical streaming replication in 3.7

Only available with BDR



Larger Clusters

Introducing data distribution

Support for a new use case enabling larger clusters

High read scalability is required

For geographically distributed applications requiring consistent, shared reference data

For example Telecom call routing, this can mean **hundreds of nodes** in a cluster across different regions

Built on new enabling capabilities

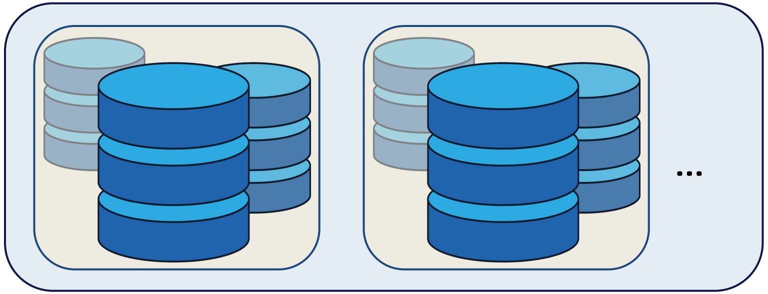
New features supporting this use case are sub-groups and subscribe-only nodes

Also leveraged is single decoding worker to minimize overhead on origin node

Sub-groups

Foundational feature for many planned enhancements

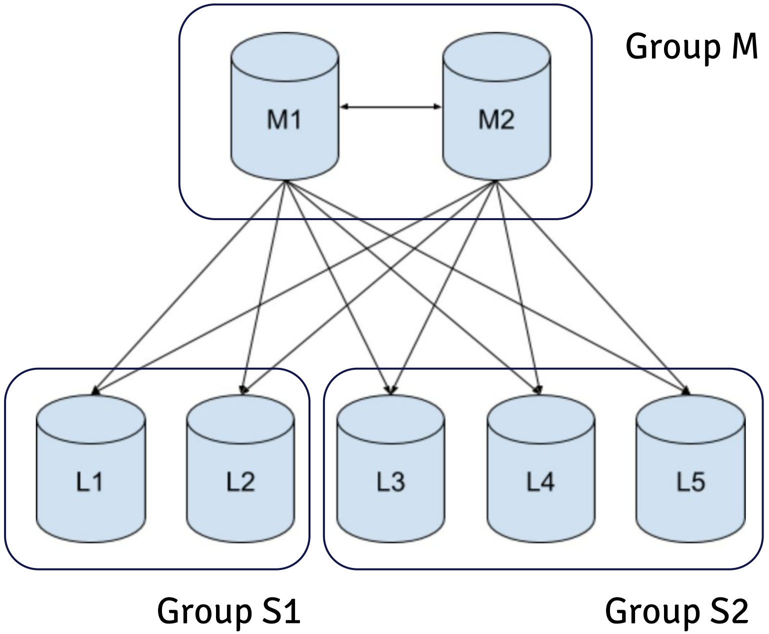
- In BDR3.6, all nodes were part of one Group
  - Mesh architecture minimizes latency between nodes
- In BDR3.7, a cluster can have nested SubGroups, with each node as part of one Group
  - AlwaysOn becomes 2 Group cluster (2x2)
  - Each Group can have >=2 Nodes
  - No limit on number of Groups
  - Each Group comes with own default repset to simplify management of filtering
- **This is a building block for more complex architectures**



Data distribution tree

Enables sharing reference data for applications requiring high read scalability

- Tree architecture minimizes **overheads in large networks**
- Data Distribution Networks
- Subscriber-Only Nodes only **receive** data from Main Group(s)
- Subscriber-Only Nodes in separate groups (sub-groups)
- Allows up to **1000** node clusters



Better Observability

Monitoring BDR

Postgres Enterprise Manager (PEM) now includes visualizations for BDR

PEM dashboards

The latest release of Postgres Enterprise Manager introduces 3 BDR monitoring dashboards for displaying information about replication activity for a BDR cluster. These are:

- Admin
- Group Monitoring
- Node Monitoring

BDR enhancements

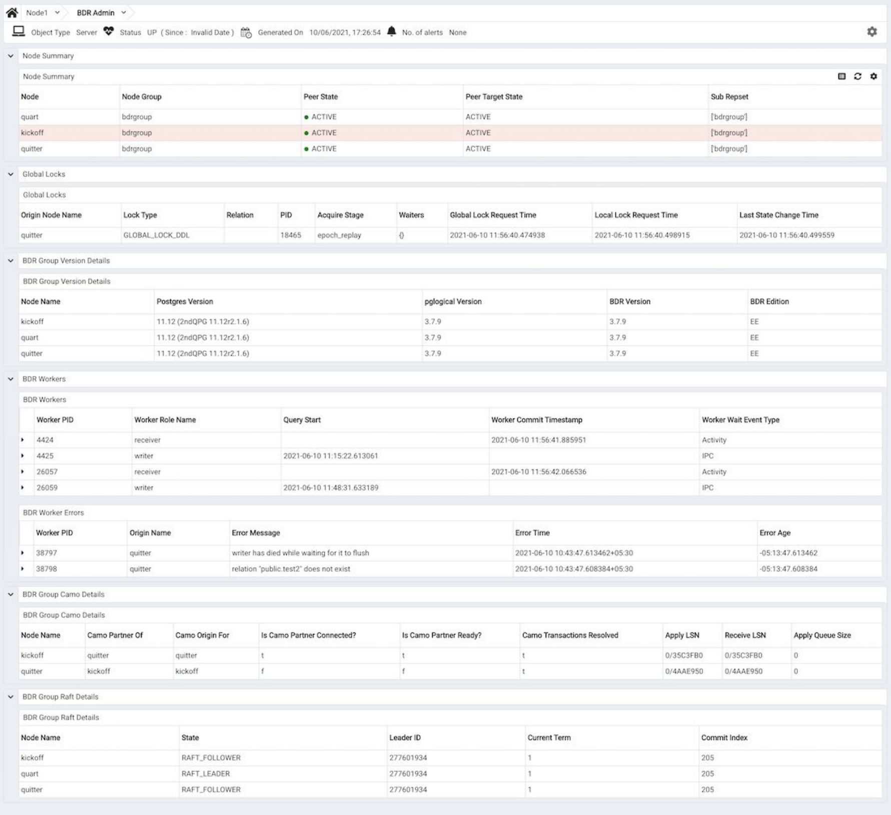
Ongoing work to improve operational insights of deployed BDR clusters. With latest BDR release a number of new views are introduced with focus on:

- Group level monitoring
- In-progress monitoring on the downstream apply side

BDR admin dashboard

This view provides cluster (or group) level information on:

- Node status across the cluster
- Status on global locks (DDL)
- Software version details by node
- Apply side worker thread status
- Worker thread errors
- CAMO status for cluster
- Raft consensus status for cluster



Object Type Server Status UP (Since : Invalid Date) Generated On 10/06/2021, 17:26:54 No. of alerts None

Node Summary

Node Summary

Node	Node Group	Peer State	Peer Target State	Sub Repset
quart	bdrgroup	● ACTIVE	ACTIVE	['bdrgroup']
kickoff	bdrgroup	● ACTIVE	ACTIVE	['bdrgroup']
quitter	bdrgroup	● ACTIVE	ACTIVE	['bdrgroup']

Global Locks

Global Locks

Origin Node Name	Lock Type	Relation	PID	Acquire Stage	Waiters	Global Lock Request Time	Local Lock Request Time	Last State Change Time
quitter	GLOBAL_LOCK_DDL		18465	epoch_replay	{}	2021-06-10 11:56:40.474938	2021-06-10 11:56:40.498915	2021-06-10 11:56:40.499559

BDR Group Version Details

BDR Group Version Details

Node Name	Postgres Version	pglogical Version	BDR Version	BDR Edition
kickoff	11.12 (2ndQPG 11.12r2.1.6)	3.7.9	3.7.9	EE
quart	11.12 (2ndQPG 11.12r2.1.6)	3.7.9	3.7.9	EE
quitter	11.12 (2ndQPG 11.12r2.1.6)	3.7.9	3.7.9	EE

BDR Workers

BDR Workers

	Worker PID	Worker Role Name	Query Start	Worker Commit Timestamp	Worker Wait Event Type
▸	4424	receiver		2021-06-10 11:56:41.885951	Activity
▸	4425	writer	2021-06-10 11:15:22.613061		IPC
▸	26057	receiver		2021-06-10 11:56:42.066536	Activity
▸	26059	writer	2021-06-10 11:48:31.633189		IPC

BDR Worker Errors

	Worker PID	Origin Name	Error Message	Error Time	Error Age
▸	38797	quitter	writer has died while waiting for it to flush	2021-06-10 10:43:47.613462+05:30	-05:13:47.613462
▸	38798	quitter	relation "public.test2" does not exist	2021-06-10 10:43:47.608384+05:30	-05:13:47.608384

BDR Group Camo Details

BDR Group Camo Details

Node Name	Camo Partner Of	Camo Origin For	Is Camo Partner Connected?	Is Camo Partner Ready?	Camo Transactions Resolved	Apply LSN	Receive LSN	Apply Queue Size
kickoff	quitter	quitter	t	t	t	0/35C3FB0	0/35C3FB0	0
quitter	kickoff	kickoff	f	f	t	0/4AAE950	0/4AAE950	0

BDR Group Raft Details

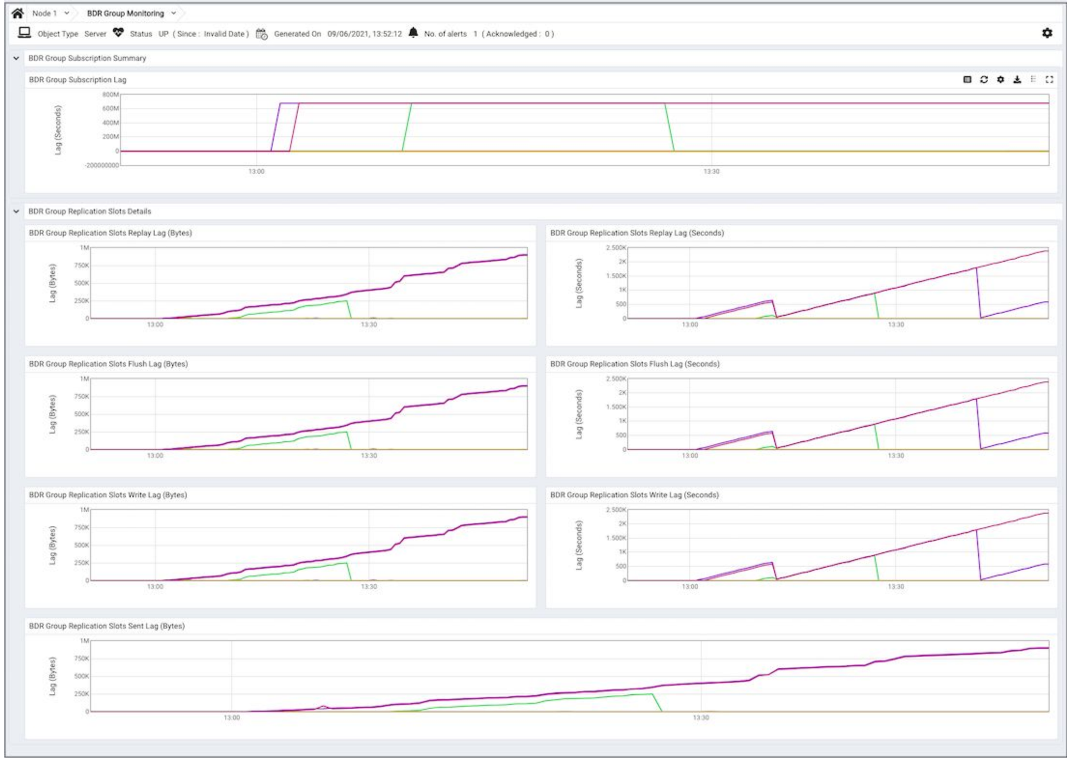
BDR Group Raft Details

Node Name	State	Leader ID	Current Term	Commit Index
kickoff	RAFT_FOLLOWER	277601934	1	205
quart	RAFT_LEADER	277601934	1	205
quitter	RAFT_FOLLOWER	277601934	1	205

BDR group monitoring dashboard

Insight to overall cluster (or group) operational health of replication is available in this view. Regular advancement of group level metrics means all nodes are actively consuming changes. Information available includes:

- Subscription lag
- Replication slots replay, flush, and write lag in bytes and seconds
- Replication slots sent lag in bytes



Node 1 › BDR Group Monitoring

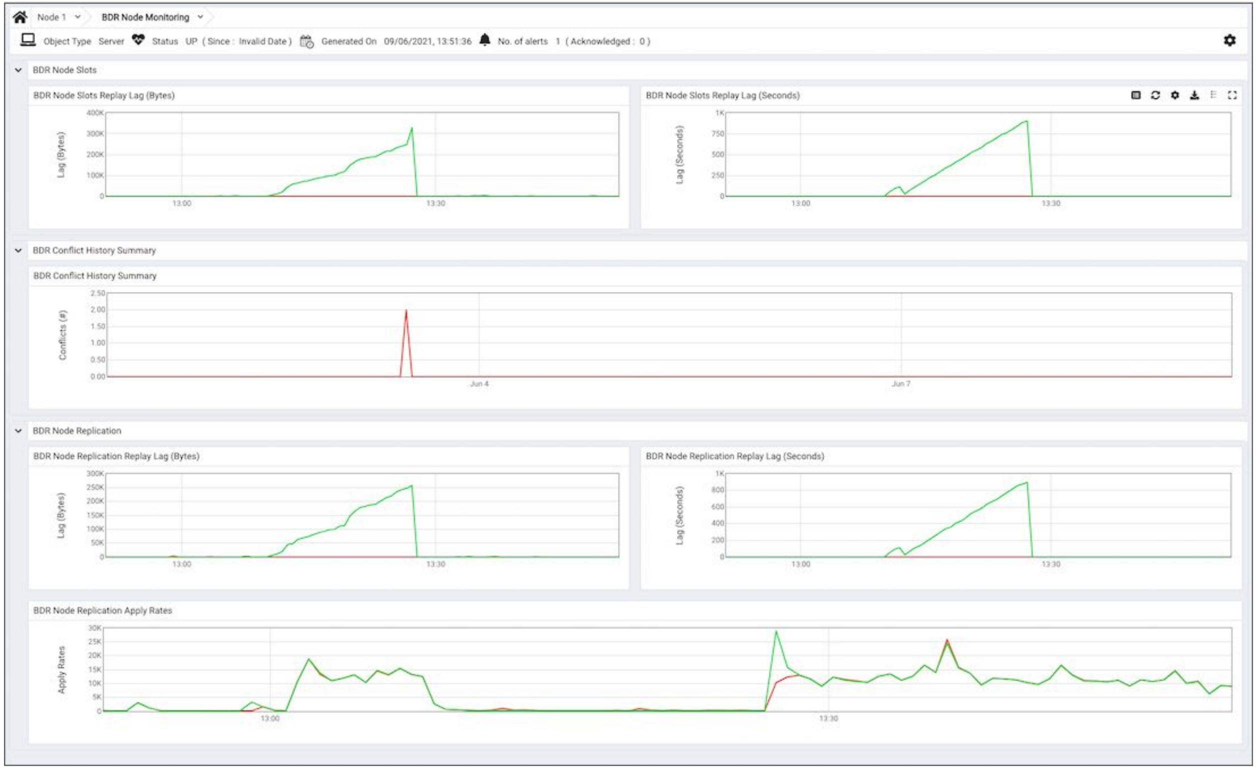
Object Type Server | Status UP (Since : Invalid Date) | Generated On 09/06/2021, 13:52:12 | No. of alerts 1 (Acknowledged : 0)

BDR Group Subscription Summary
BDR Group Subscription Lag
Lag (Seconds)
800M
600M
400M
200M
0
-200000000
13:00
13:30
BDR Group Replication Slots Details
BDR Group Replication Slots Replay Lag (Bytes)
BDR Group Replication Slots Replay Lag (Seconds)
BDR Group Replication Slots Flush Lag (Bytes)
BDR Group Replication Slots Flush Lag (Seconds)
BDR Group Replication Slots Write Lag (Bytes)
BDR Group Replication Slots Write Lag (Seconds)
BDR Group Replication Slots Sent Lag (Bytes)
Lag (Bytes)
1M
750K
500K
250K
0
2.500K
2K
1.500K
1K
500

BDR node monitoring dashboard

Node level visibility:

- For slots used by BDR in a database the outgoing replication replay lag in bytes and seconds (difference between applied LSN and current WAL write position)
- A summary of conflicts encountered
- Outgoing replication rates for a node including replay lag in bytes and seconds
- Rate in LSNs applied per second for a BDR node



Node 1 · BDR Node Monitoring

Object Type Server · Status UP (Since : Invalid Date) · Generated On 09/06/2021, 13:51:36 · No. of alerts 1 (Acknowledged : 0)

HA for Oracle SQL Compatible Applications

EDB Postgres Advanced with BDR

Benefits of Oracle SQL compatibility and 5 nines availability

Your tier 1 apps

Leverage existing infrastructure with native PL/SQL support and Oracle Call Interface (OCI) interoperability

Your people

Leverage the existing skills of your Oracle DBAs and developers

Your business

Lower costs, reduce risks, and move faster

BDR Feature Overview

A full-featured multi-master replication solution for PostgreSQL clusters

Essentials

Provides the essential multi-master capabilities for PostgreSQL clusters.

- Enables application and database upgrades without requiring downtime
- Provides clusters row level eventual consistency by default
- Tools to monitor operation and verify data consistency
- Extends PostgreSQL logical replication beyond unidirectional, standby use cases

Advanced

Includes advanced conflict management, data-loss protection, and up to 5X faster.

- Guards applications from committing transactions more than once
- Conflict-free synchronous replication with two phase commit
- Concurrent updates using conflict-free replicated data types (CRDTs)
- Configurable column level conflict resolution with customizable conflict handling and transformation

A Look Ahead

Roadmap

A vision for advancing very highly available Postgres clusters

Q2: BDR 3.7 GA

Support for v11-13

Up to 5X throughput with parallel apply

Single decoding worker streamlines upstream replication

High read scalability through data distribution tree architecture with subscribe-only nodes

Oracle SQL compatibility support with Postgres Advanced

PEM support for monitoring BDR

Q4: BDR 4.0 GA

Support for v12-14

Oracle SQL compatibility support is complete with Postgres Advanced v14

Features enabled are:

- Eager replication
- CAMO
- Single decoding worker
- Application assessment

Improved cluster management provides zero transaction lag switchover

2022 and beyond

Support for v13-15

Expand pipeline parallelism and enhance large transaction support for faster throughput

Enhance autopartition to support bigger tables in clusters

Evolve data distribution capabilities in support of bigger clusters

Develop autoscale capabilities to enable bigger databases

Thank you!

john.dalton@enteprisedb.com

EDB™