



CONFLOSS
CONFERÊNCIA DE FREE/LIBRE E OPEN SOURCE SOFTWARE

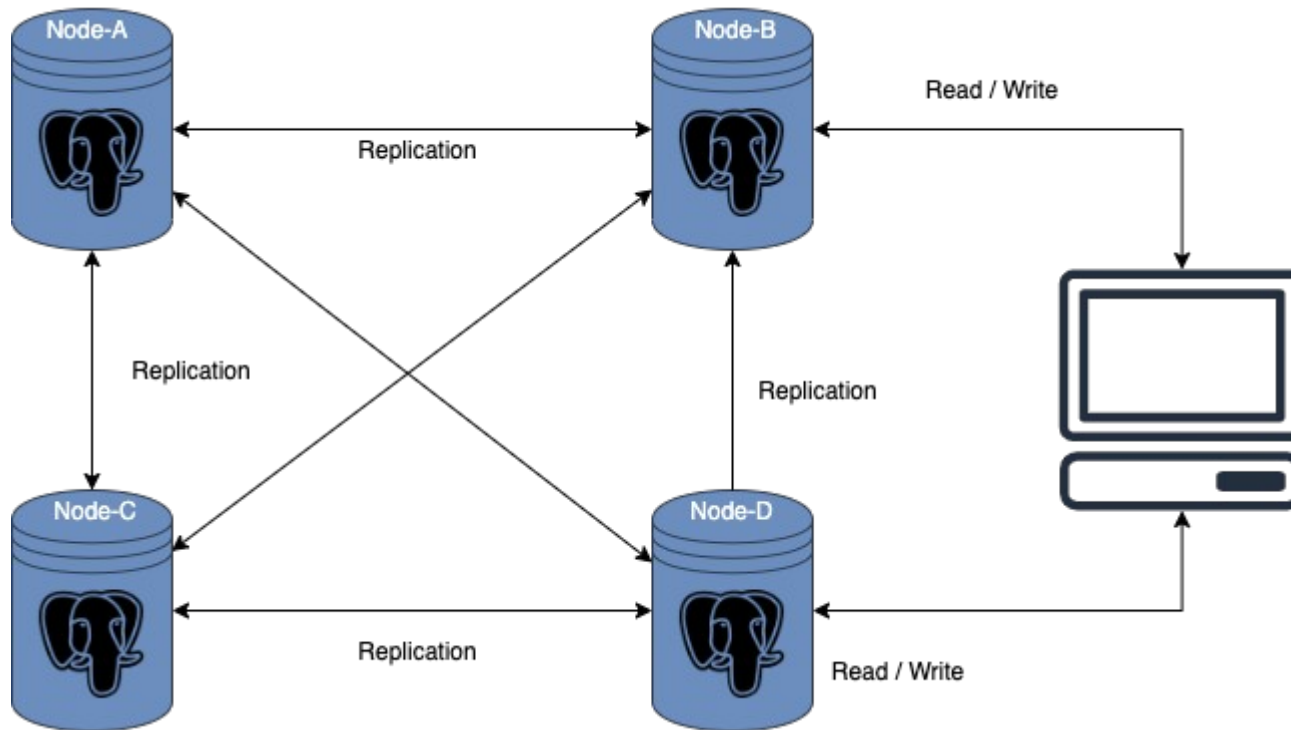
Ambiente Livre
Business Intelligence e Analytics

PostgreSQL

Implementando um Cluster Master-Master no
PostgreSQL com Bucardo
CONFLOSS 2023

Marcio Junior Vieira
CEO & Data Scientist, Ambiente Livre

Replicação Multi-Master



Prós

- No caso de um mestre falhar, o outro mestre está lá para servir a atualização e a inserção.
- Os nós mestres estão localizados em vários locais diferentes, portanto, as chances de falha de todos os mestres são mínimas.
- As atualizações de dados são possíveis em vários servidores.
- O aplicativo não requer rotear o tráfego para apenas um único mestre.



Contra

- A principal desvantagem da replicação multi-master é sua complexidade.
- A resolução de conflitos é muito difícil porque gravações simultâneas em vários nós são possíveis.
- Às vezes, a intervenção manual é necessária em caso de conflito .
- Possibilidade de inconsistência de dados.

Opções para PostgreSQL



Contra

- O PostgreSQL nativo não implementa Master-Master
- Tabela de Opções ?

Serial Number	Replication	Company	Open-Source	Language	Type	Failover	Synchronous	Version	Active
1	BDR3	2ndQuadrant	N	C	Extension	Y	ASync	PostgreSQL 12	Y
2	BDR2	2ndQuadrant	Y	C	Extension	Y	ASync	PostgreSQL 9.6	Y
3	xDB	EnterpriseDB	N	Java	Application	Y		PostgreSQL 12	Y
4	PostgreSQL-XL	2ndQuadrant*	Y	C	Fork	Y		PostgreSQL 10	Y
5	PostgreSQL-XC/XC2	EnterpriseDB and NTT	Y	C	Fork	Y			Y
6	Rubyrep	Arndt Lehmann	Y		Application	Y			N
7	Bucardo	End Point Corporation	Y	PERL	Application	Y		Postgres 12	Y

- Fonte: <https://www.percona.com/blog/multi-master-replication-solutions-for-postgresql/>



Bucardo

Multi-master Database
Replication



check_postgres

Nagios/Icinga-friendly
PostgreSQL monitoring



DBIx::Safe

Restricted Database Handles
in Perl



tail_n_mail

Live logfile monitoring



Boxinfo

Gather host info easily



pgsi

Wiki-ready query
performance reports

Conceito

- Sistema de replicação PostgreSQL assíncrono, permitindo operações multi-master e multi-slave.
- Replica dados entre bancos de dados PostgreSQL, MySQL, Redis, Oracle, MariaDB, SQLite ou MongoDB.
- Desenvolvido no Backcountry.com por Jon Jensen e Greg Sabino Mullane da End Point Corporation.
- Software Livre e de código aberto lançado sob a licença BSD.
- Desenvolvido em Perl.

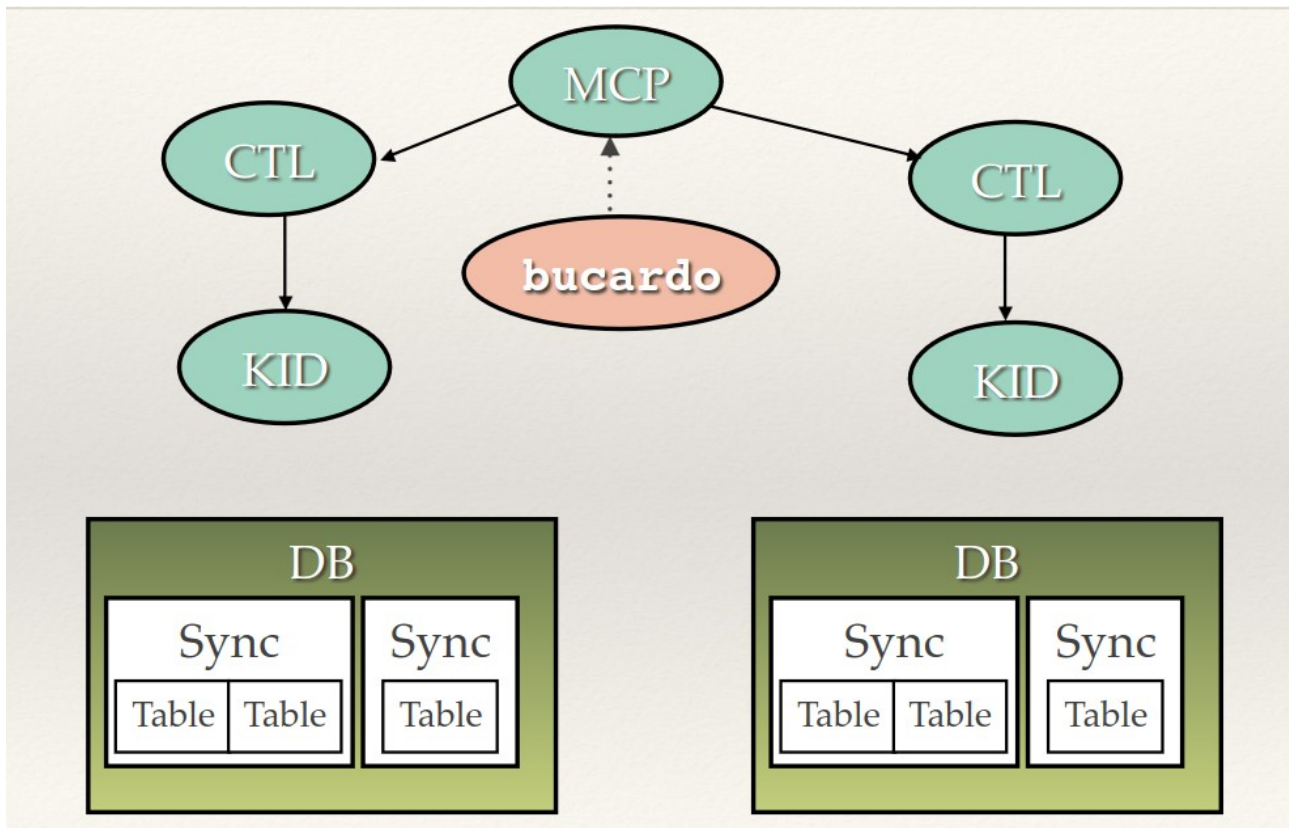
Requisitos

- Perl e libs Perl e postgresql-perl.
- Requer um banco de dados dedicado e é executado como um daemon Perl que se comunica com esse banco de dados e todos os outros bancos de dados envolvidos na replicação.
- Todas as informações específicas que o daemon precisa são armazenadas no banco de dados bucardo principal, incluindo uma lista de todos os bancos de dados envolvidos na replicação e como alcançá-los, todas as tabelas que devem ser replicadas e como cada uma deve ser replicada.

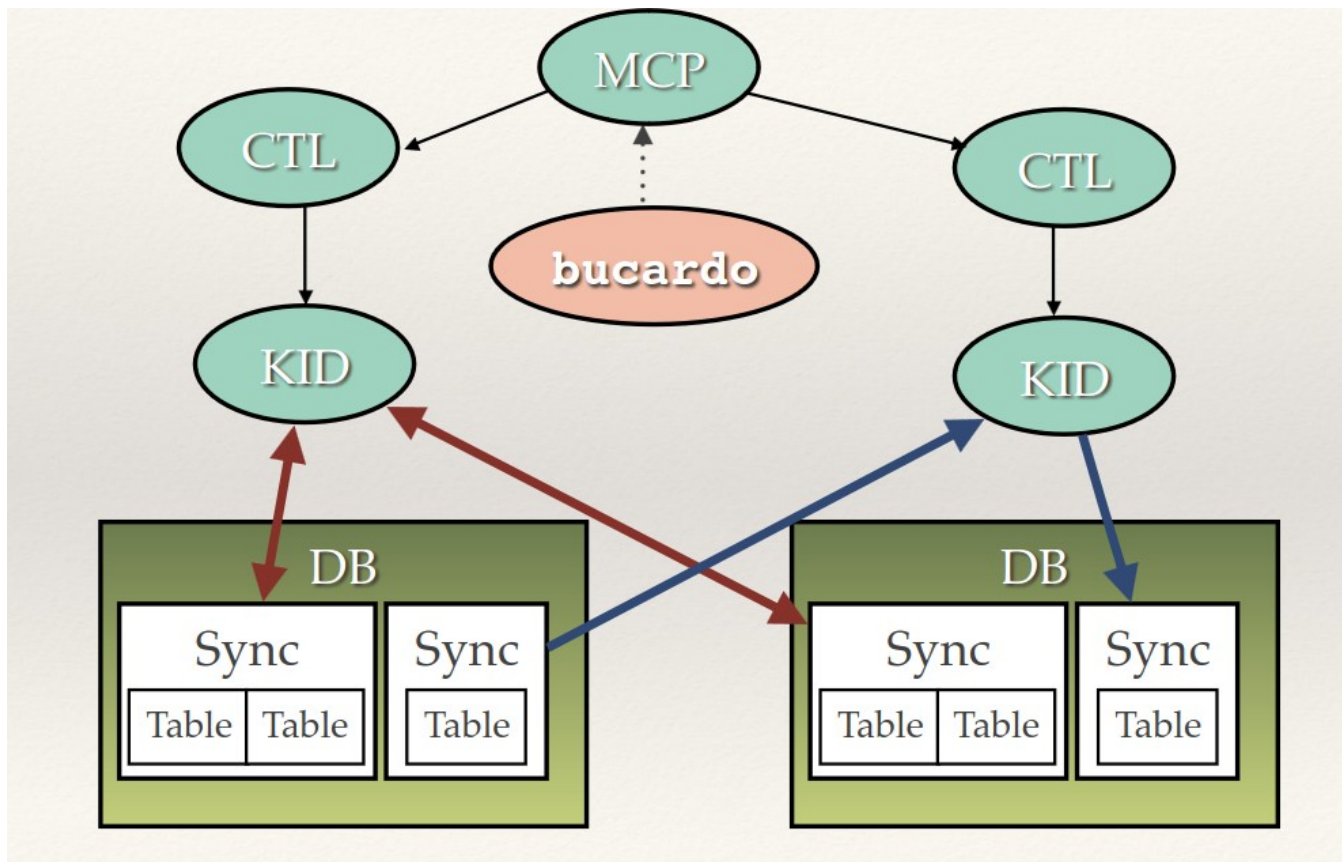
Multi-master

- A replicação multi-master usa dois ou mais bancos de dados, com resolução de conflitos (opções padrão ou sub-rotinas personalizadas) para lidar com a mesma atualização em ambos os lados.
- A replicação mestre-slave envolve uma ou mais origens indo para um ou mais destinos.
- A origem deve ser PostgreSQL, mas os destinos podem ser PostgreSQL, MySQL, Redis, Oracle, MariaDB, SQLite ou MongoDB.

Bucardo - Master-Master



Bucardo - Master-Master



Workflow

- A primeira etapa na execução é adicionar dois ou mais bancos de dados ao banco de dados principal do Bucardo.
- Feito isso, são adicionadas informações sobre quais tabelas devem ser replicadas, bem como quaisquer agrupamentos de tabelas.
- Em seguida, as sincronizações são adicionadas. As sincronizações são ações de replicação nomeadas, copiando um conjunto específico de tabelas de um servidor para outro servidor ou grupo de servidores.
- Uma vez configurado o Bucardo, as triggers começam a armazenar informações sobre quais linhas foram alteradas em todas as tabelas de interesse.

Sincronização de troca (multi-master)

- Uma alteração é feita na tabela e registrada na tabela bucardo_delta.
- Um aviso é enviado ao daemon principal, informando alteração.
- O daemon notifica o controlador para essa sincronização e volta a escutar.
- O controlador cria um “kid” para lidar com a replicação.
- O kid inicia uma nova transação e desabilita triggers e regras nas tabelas.
- Em seguida, reúne uma lista de quais linhas foram alteradas desde a última replicação e compara as duas para descobrir o que deve ser feito.
- Se houver um conflito, o manipulador de conflitos padrão ou personalizado, definido por tabela, será executado para resolver as coisas.
- Triggers e regras são reativados e a transação é confirmada.
- Se a transação falhar, todos os manipuladores de exceção personalizados serão executados.
- O kid sinaliza para o controlador que terminou.



Cli commands

- bucardo install
- bucardo add
- bucardo status
- bucardo kick
- bucardo help

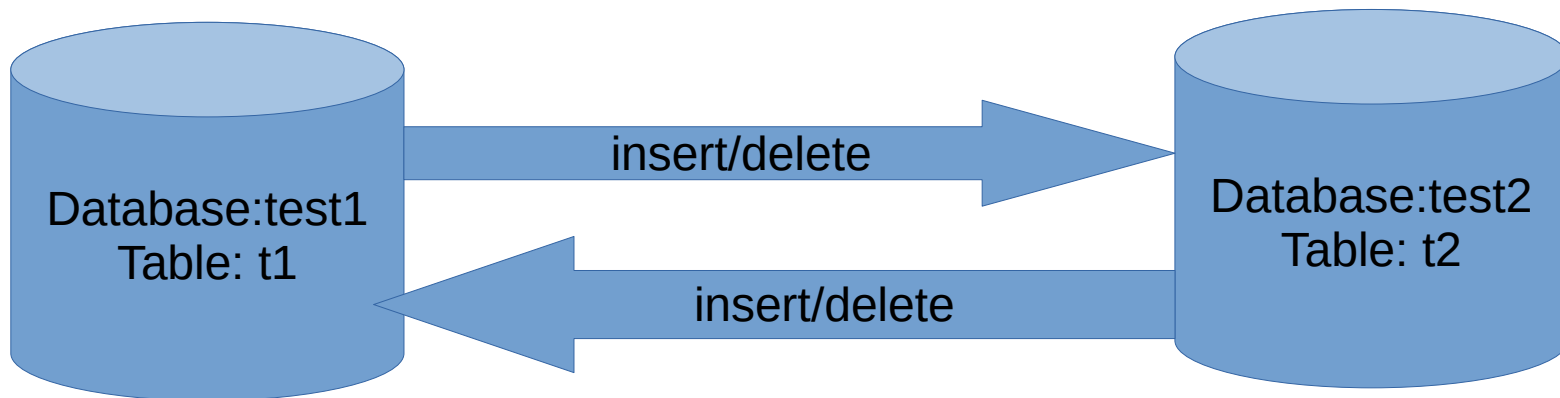


Limitações

- Sem manipulação automática de DDL
- Requer PKs

O que faremos

- Instalar o Bucardo.
- Criar dois databases (no mesmo server para teste)
- Apresentar Replicações.



Obrigado

Marcio Junior Vieira

marcio@ambientelivre.com.br

@marviojvieira @ambientelivre

<https://www.linkedin.com/in/mvieira1/>