



PGCONF.BRASIL 2024
BELO HORIZONTE, MG
7 E 8 DE NOVEMBRO



Implementando um Cluster Master-Master no PostgreSQL com Bucardo

PGCONF Brasil 2024

Marcio Junior Vieira
CEO & Data Scientist, Ambiente Livre

Mini-CV

- 22 anos de experiência em TI, vivência em desenvolvimento e análise de sistemas de gestão empresarial e ciência de dados.
- CEO da Ambiente Livre atuando como Cientista de Dados, Engenheiro de Dados e Arquiteto de Software.
- Professor dos MBAs em Big Data & Data Science, Inteligência Artificial e Business Intelligence e Analytics da Universidade Positivo.
- Professor do MBA Artificial Intelligence e Machine Learning da FIAP.
- Pesquisador do Laboratório de Tecnologias para Tomada de Decisão da Universidade de Brasília (Unb/Latitude).
- Trabalhando com Free Software e Open Source desde 2000 com serviços de consultoria e treinamento.
- Graduado em Tecnologia em Informática(2004) e pós-graduado em Software Livre(2005) ambos pela UFPR.
- Palestrante FLOSS em: FISL, TDC, Latinoware, Campus Party, Pentaho Day, Ticonova, PgDay e FTSL.
- Organizador Geral: Pentaho Day 2017, 2015, 2019 e apoio nas ed. 2013 e 2014.
- Data Scientist, instrutor e consultor de Big Data e Data Science com tecnologias abertas.
- Ajudou a capacitar equipes de Big Data na IBM, Accenture, Tivit, Serpro, Natura, MP, Netshoes, Embraer entre outras.
- Especialista em implantação e customização de Big Data com Hadoop, Spark, Pentaho, Cassandra e MongoDB.
- Contribuidor de projetos internacionais, tais como Pentaho, LimeSurvey, SuiteCRM e Camunda.
- Especialista em implantação e customização de ECM com Alfresco e BPM com Activiti, Flowable e Camunda.
- Certificado (Certified Pentaho Solutions) pela Hitachi Vantara (Pentaho).
- Membro da The Order Of de Bee (comunidade Alfresco para desenvolver o ecossistema Alfresco independente)
- Trabalha profissionalmente com PostgreSQL desde 2006.



Sobre a Ambiente Livre

Open Software for Business

- Fundada em 2004 com foco em consultoria com FLOSS.
- Experts em 34 soluções para geração de negócios com Software Livre/Código Aberto.
- Atualmente estamos sediados no Hub de Inovação Mindhub em Curitiba (FAE).

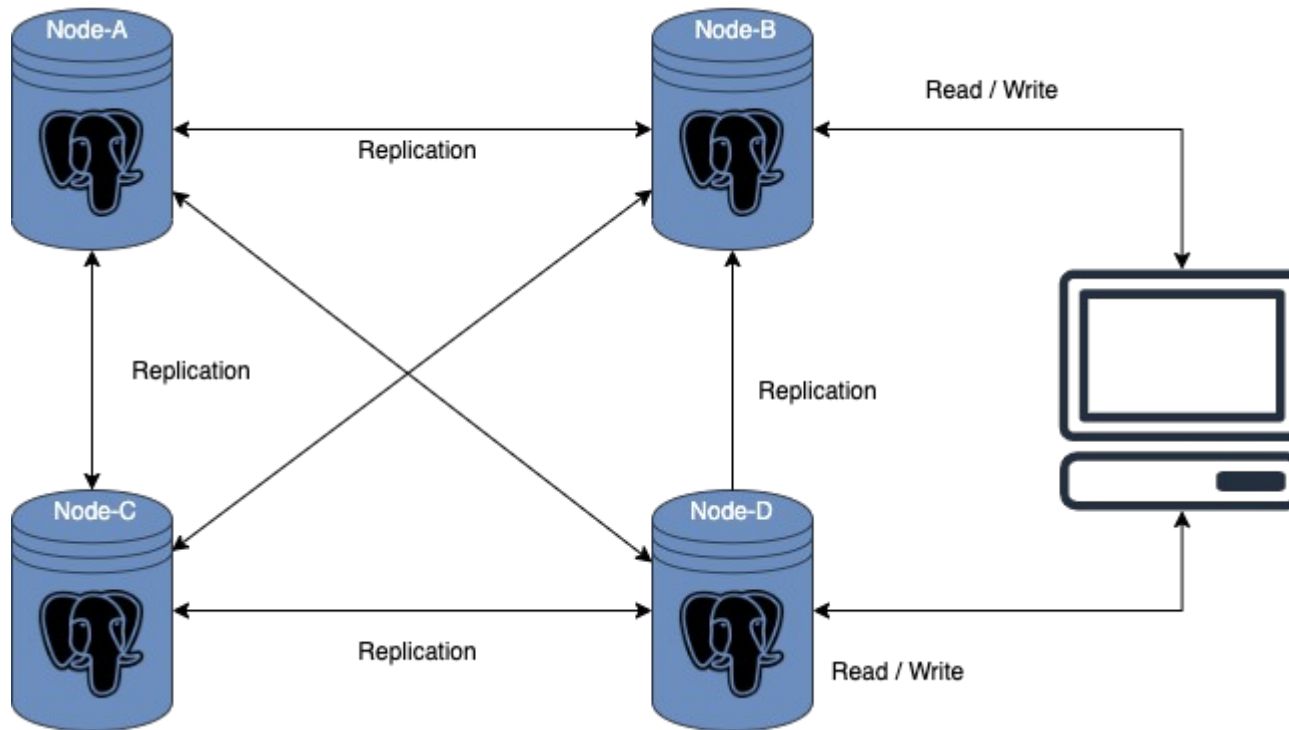


Nosso Ecossistema de Serviços

Data Driven	CRM, CMS e ITSM	ECM e BPM	Infra - IAC - DevOps
<p>Painéis de Indicadores Cubos e Relatórios Análise Preditiva Processamento Distribuído Banco de Dados Colunares</p> <p>Dashboards e OLAP Data Integration e Data Mining Big Data & Data Lake Machine Learning Business Intelligence & Analytics</p> <p>Consultoria Treinamento Projeto</p>	<p>Help Desk e Inventory Pesquisas Online Marketing e Vendas SAC e Pós-vendas Portais de Conteúdo</p> <p>IT Service Management Customer Relationship Management Content Management System Content Management Framework EAD e LMS</p> <p>Consultoria Treinamento Projeto</p>	<p>Gestão de Documentos Gerenciamento de Mídias Processo de Negócio BPMN e BPMS Microserviços</p> <p>Enterprise Content Management Records Management Business Process Management Microservices Orchestration</p> <p>Consultoria Treinamento Projeto</p>	<p>DepOps DevSecOps MLOps e DataOps Native Cloud Distributed Systems</p> <p>Web Server Kubernetes-as-a-Service Object Storage Containers Building Blocks</p> <p>Consultoria Treinamento Projeto</p>



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.
- 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.

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 para o PostgreSQL assíncrono, permitindo operações multi-master e multi-slave.
- Desenvolvido no Backcountry.com(Jon Jensen) e Greg Sabino Mullane (End Point Corporation) em 2002 usando PG7.
- Release publicada em 2007.
- Open Source, lançado sob a licença BSD*.
 - * 2 clausulas (citar autor e isenção de responsabilidade)
- Desenvolvido em Perl.
- Versão mais atual 5.6.0.



Curiosidade

- Referência ao Bucardo, uma subespécie de cabra selvagem que vivia nas montanhas dos Pireneus.
- Foi declarado extinto em 2000, mas, em 2003, cientistas tentaram reviver a espécie com técnicas de clonagem.
- A experiência teve um sucesso temporário, criando o primeiro animal “desextinto”, mas o clone viveu apenas alguns minutos.



Conceito

- **Multi-Master** - Suporta somente PostgreSQL.
- **Multi-Slave** - A origem deve ser PostgreSQL, mas os destinos podem ser PostgreSQL, MySQL, Redis, Oracle, MariaDB, SQLite ou MongoDB.

Multi-Master



Multi-Slave



ORACLE



redis



Vantagens

- Poucos requisitos.
- Não é necessário configurar nada no PostgreSQL (somente ter um usuário de replicação para o daemon)
- 1 Daemon para todos casos (Master ou Slave)
- Rápido.
- Bom CLI de monitoramento.
- Fácil de Instalar e Operar.



Requisitos

- PostgreSQL 8.3+
- Perl e libs Perl e postgresql-perl.
- Um banco de dados dedicado para seu database (DB bucardo).

Serviços

- Daemon Perl que se comunica o banco de dados e todos os outros bancos de dados envolvidos na replicação.
- Todas as informações específicas do daemon são armazenadas no banco de dados bucardo.
- O BD Bucardo tem uma lista de todos bancos de dados envolvidos na replicação, todas as tabelas a serem replicadas e como cada uma deve se comportar.



Multi-master

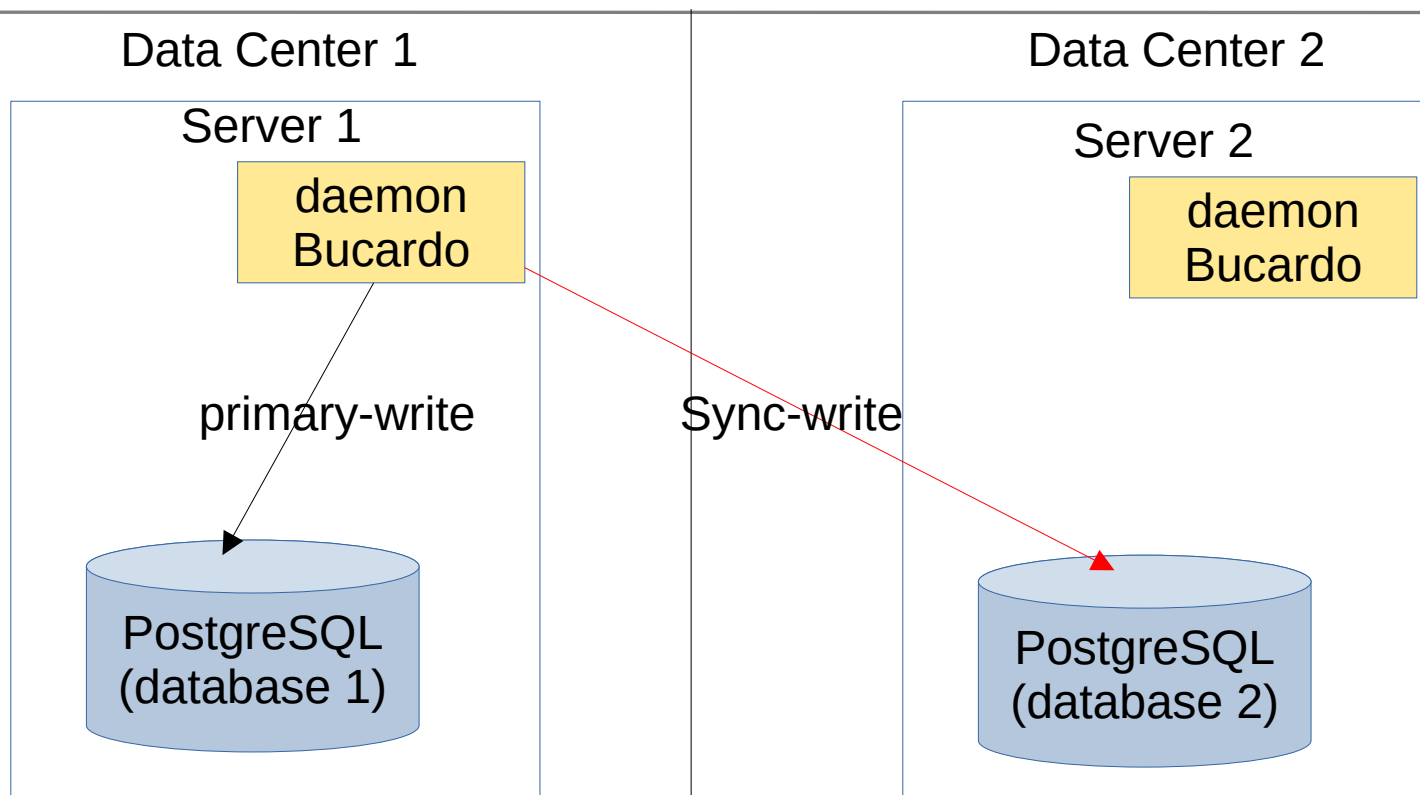
- A replicação multi-master usa dois ou mais bancos de dados, com resolução de conflitos (opções padrão ou sub-rotinas personalizadas) para lidar com a mesma atualização em ambos os lados.

Master-slave

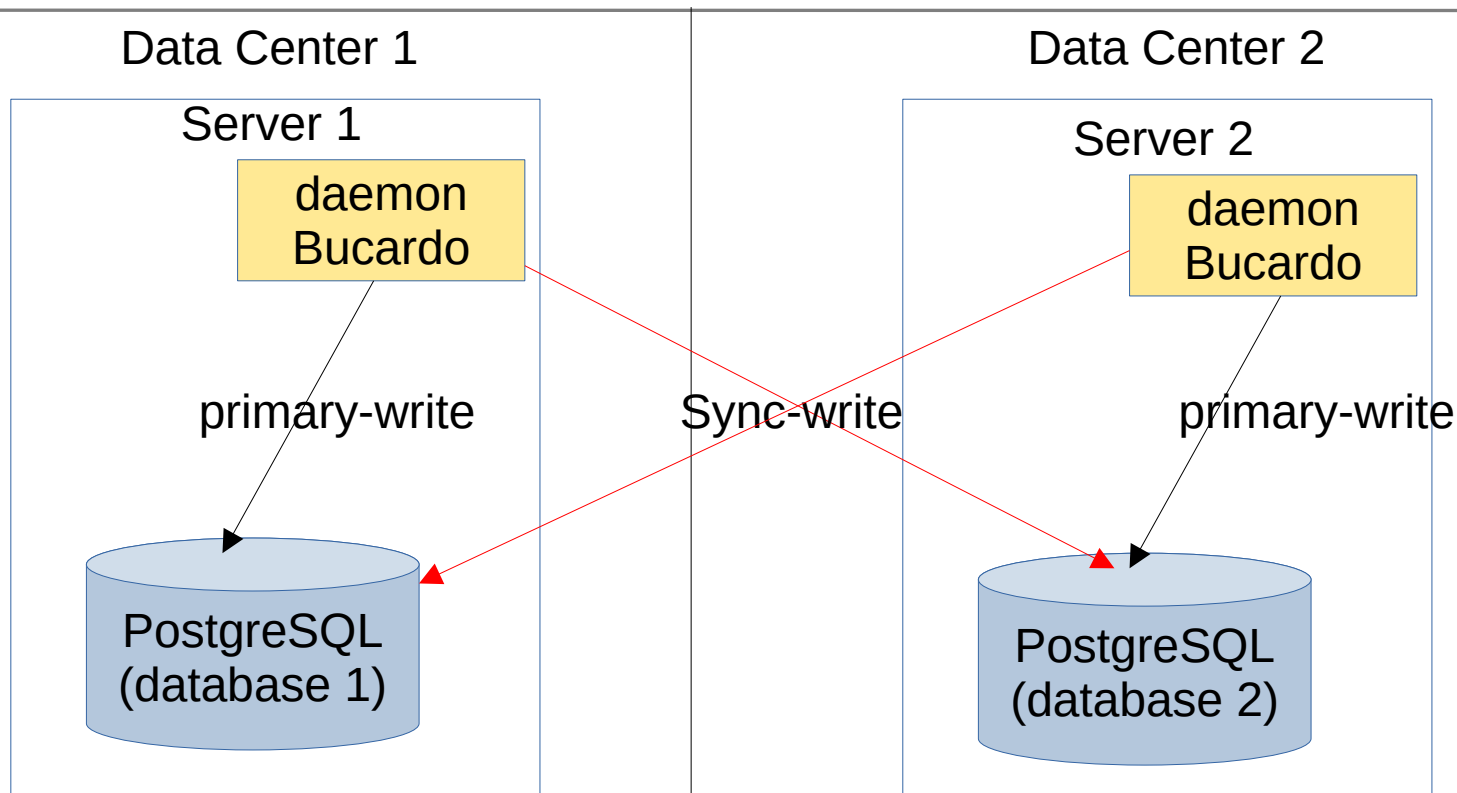
- A replicação master-slave envolve uma ou mais origens indo para um ou mais destinos.



Bucardo - Master-Slave



Bucardo - Master-Master



Workflow

- 1) Na execução é adicionar dois ou mais bancos de dados ao banco de dados principal do Bucardo.

bucardo add database **dbtest1 dbname=test1**

- 2) Adicionar informações sobre quais tabelas devem ser replicadas, bem como quaisquer agrupamentos de tabelas.

bucardo add **all tables db=dbtest1**

bucardo add all tables db=dbtest1 **--herd=dbtest1_dbtest2**



Workflow

- 3) As sincronizações são adicionadas que são ações de replicação nomeadas, copiando um conjunto específico de tabelas de um servidor para outro servidor ou grupo de servidores.

```
bucardo add sync sync_dbpmis1 relgroup=dbtest1_dbtest2  
dbs=dbtest1,dbtest2
```

- Etapa 4)Uma vez configurado o Bucardo, as triggers começam a armazenar informações sobre quais linhas foram alteradas em todas as tabelas de interesse.

```
bucardo start
```



Sync

- Immediate
- Timed
- Manual

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.

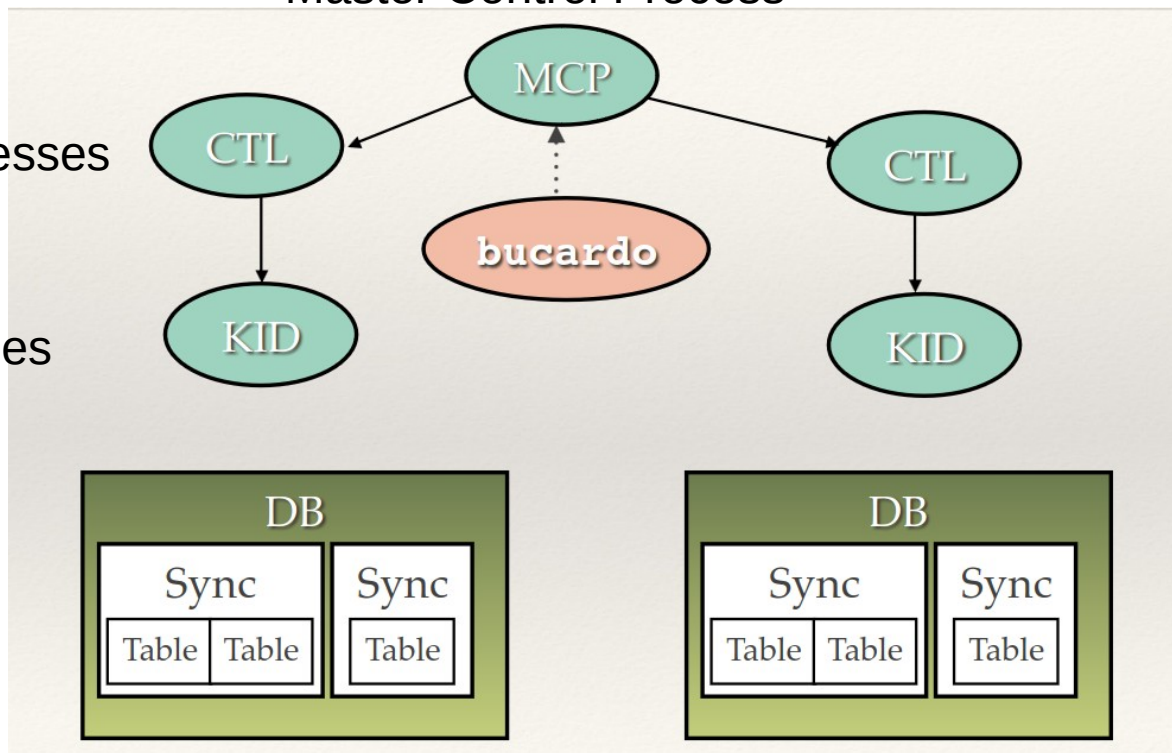
O kid sinaliza para o controlador que terminou.



Master Control Process

Controller Processes

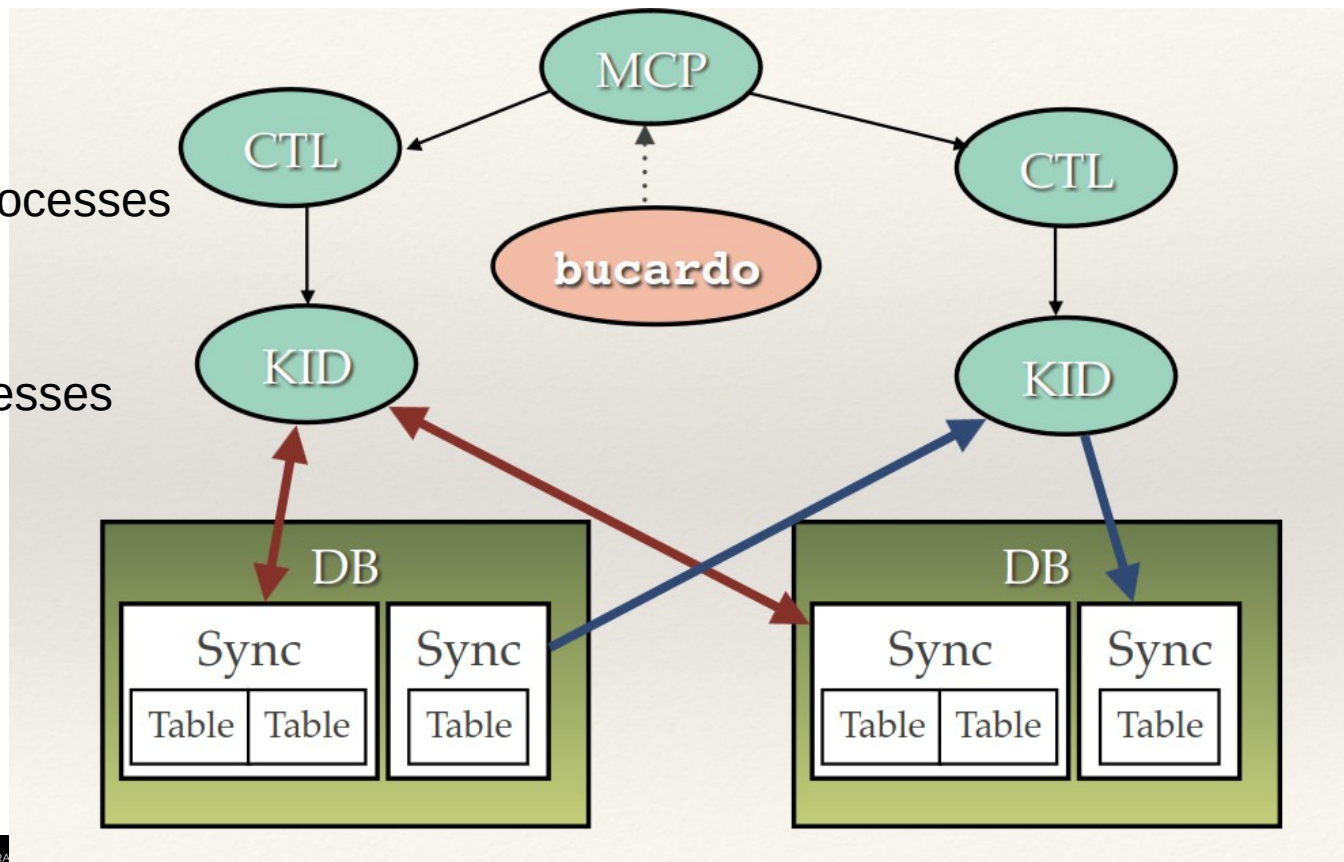
KID processes



Master Control Process

Controller Processes

KID processes



Cli commands

- bucardo install
- bucardo add {database, table, all tables}
- bucardo status
- bucardo kick
- bucardo help
- bucardo list {tables,herd,sync}
- bucardo {start/stop}

Custom

- Quando acontece um conflito o bucardo executa um customcode onde você pode escrever sua lógica de resolução.
- No Master-Master temos mais potencial de conflitos, uma modificação da mesma linha gera um conflito.
- O Bucardo tem alguns métodos de resoluções de conflitos padrão
 - * source
 - * target
 - * random
 - * latest

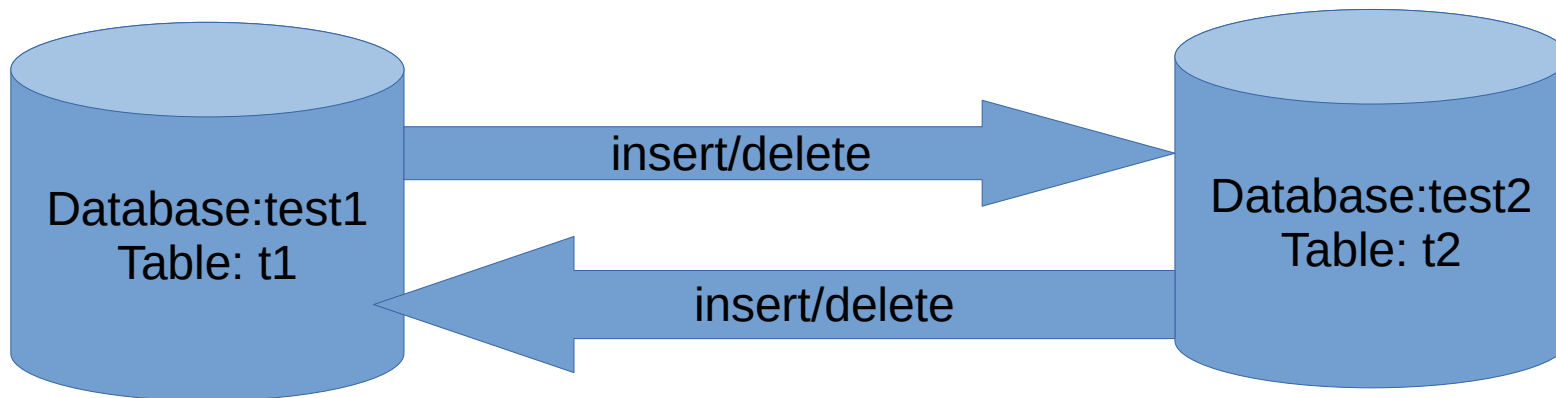


Limitações

- Sem manipulação automática de DDL
- Requer PKs

O que faremos

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



Obrigado

A decorative white wave graphic consisting of two stylized, overlapping curves, positioned on the left side of the slide.

Marcio Junior Vieira
marcio@ambientelivre.com.br
[@marviojvieira](#) @ambientelivre
<https://www.linkedin.com/in/mvieira1/>

Palestra e Tutorial:

**[github.com/ambientelivre/labs/
talks/pgconf/2024/](https://github.com/ambientelivre/labs/talks/pgconf/2024/)**