

# PostgreSQL'de Uygulamalı Gömülü Replikasyon (Streaming Replication ve Hot Standby)

Devrim GÜNDÜZ  
Principal Systems Engineer  
EnterpriseDB  
devrim.gunduz@EnterpriseDB.com

# Bu anlatan da kim?



## Bu anlatan da kim?

- Zararlı fanatik seviyesinde açık kaynak kod kullanıcısıyım.
- PostgreSQL'in ana katkıcılarından birisiyim.
- 1997 yılından beri masaüstünde Linux kullanıyorum. Fedora geliştiricisiyim.
- Kod yazmıyorum (pek).
- EnterpriseDB'de çalışıyorum. Eğlenceli bir işim var.
- İstanbul'da yaşıyorum.
- Metal dinliyorum (genelde)
- Bir oğlum var.



- ▶ **PostgreSQL'de WAL kavramı ve arşivleme temelleri**
- ▶ **Streaming Replication (SR) kavramı**
- ▶ **Hot Standby (HS) kavramı**
- ▶ **SR+HS için ana (master) sunucuda yapılacaklar**
- ▶ **SR+HS için standby sunucuda yapılacaklar**
- ▶ **Sorular**



## PostgreSQL'de WAL kavramı ve arşivlemesi

# PostgreSQL'de WAL kavramı



postmaster

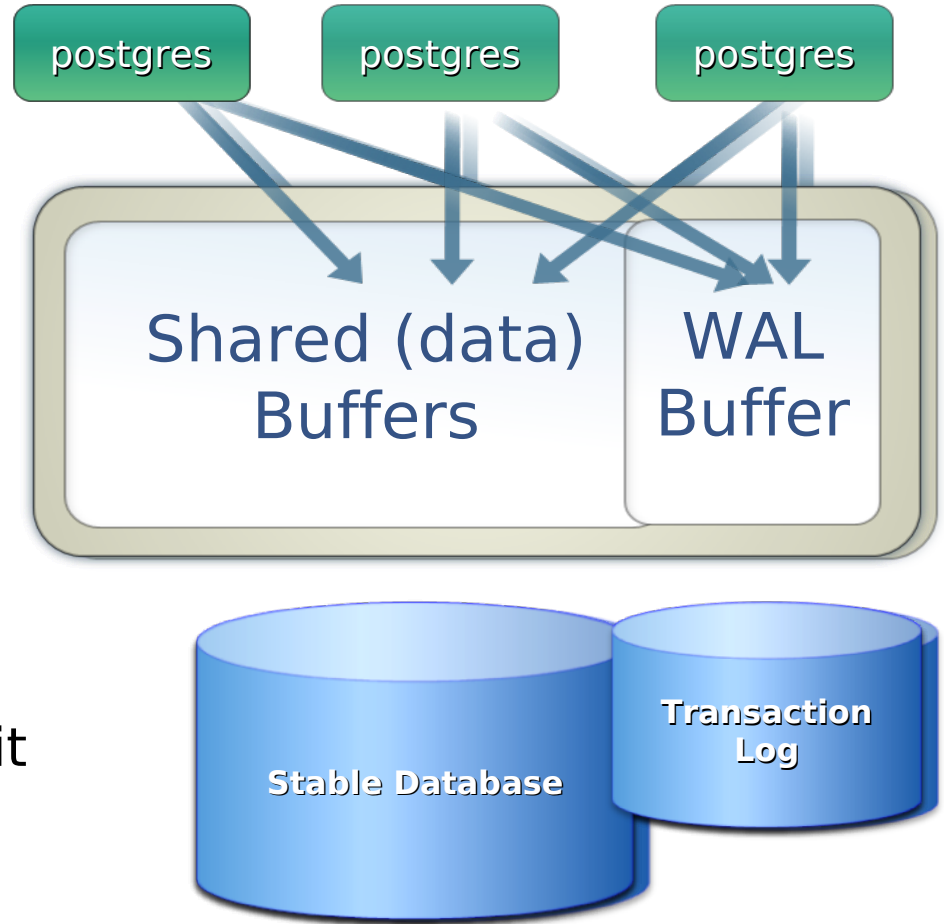
postgres süreçleri

Veri veritabanı tamponuna  
(buffer) yazılır.

Transaction log verisi WAL  
tamponuna yazılır.

Commit gerçekleşince wal  
buffer diske yazılır (flush,  
pg\_xlog)

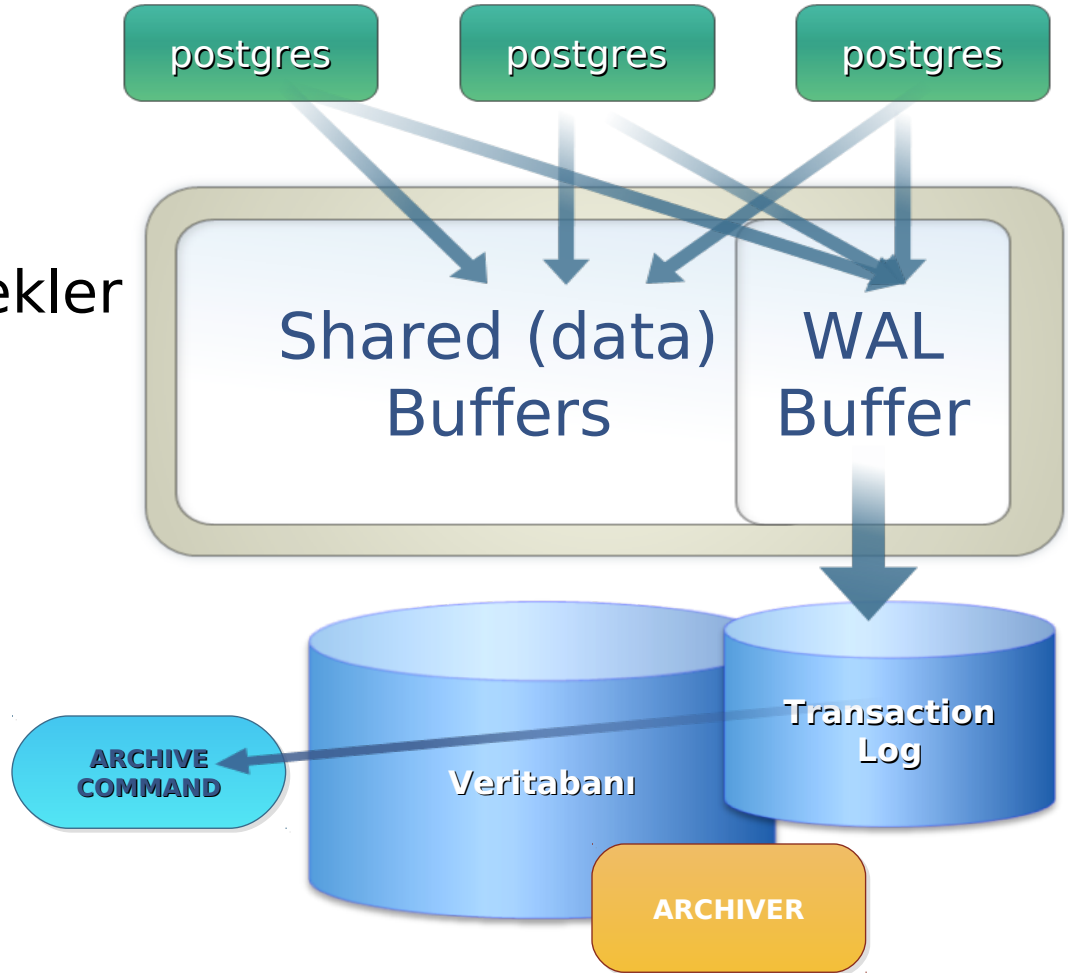
Cache dolu ise group commit



# WAL Arşivleme temelleri



- Postmaster
- Archiver süreci
- Xlogların dolmasını bekler
- Dosya dolunca kopya işlemi başlar





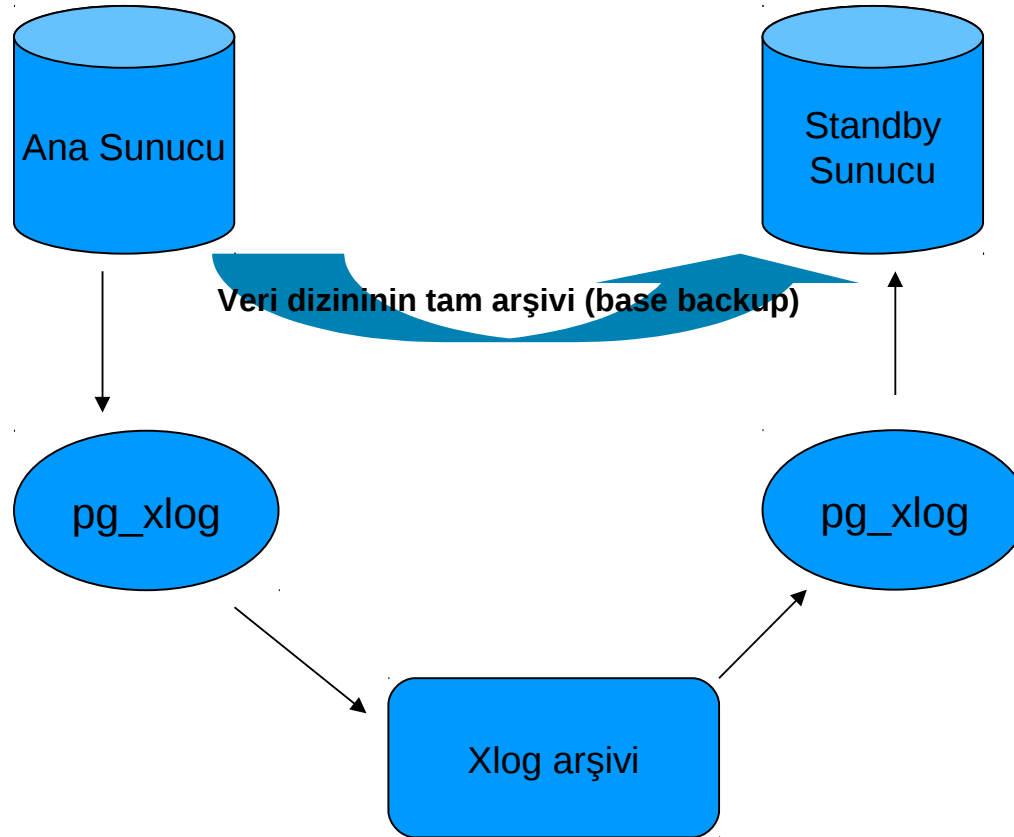
# Streaming Replication (SR) ve Hot Standby (HS) Temelleri



- SR ve HS PostgreSQL 9.1'in ana özelliklerinden birisidir.
- SR, replikasyon yükünü çok azaltan bir yapıdır.
- HS ise standby sunucuda salt okunur sorgulara izin verir.
- WAL kayıtları “stream” edilir, böylece xlog taşımaktan kaynaklanan gecikmeler ortadan kalkar.
- WAL dosyaları her zaman gereklidir -- o yüzden ayrıca arşivlemek gereklidir.



# SR ve HS temelleri





## SR ve HS için ana sunucuda yapılması gerekelenler

# Ana sunucuda yapılması gerekenler



## ► wal\_level=hot\_standby

wal\_level transaction logları içindeki transaction bilgisinin miktarını belirtir. Hot standby için, transaction loglarında en fazla bilgiye gereksinim duyulur. Bu nedenle bu seçeneği hot\_standby yapmalıyız.

# Ana sunucuda yapılması gerekenler



## ▶ `max_wal_senders=2`

En fazla walsender süreci sayısıdır. Genelde slave sunucu sayısı +1 şeklinde hesaplayabilirsiniz.

## ▶ `archive_mode=on`

xlog arşivlemesini etkinleştirir.

## ▶ `archive_command='...'`

xloglar arşivlenirken çalıştırılacak komut ya da komutlar. Bu komutu doğrudan `postgresql.conf`'a yazmamak, bizi reload sürecinden kurtarır.

– `%p`, `%f`

Bu değişikliklerin etkin olması için PostgreSQL'i restart etmeniz gerekmektedir.

# Ana sunucuda yapılması gerekenler



- ▶ Replikasyon kullanıcısını yaratmak  
CREATE ROLE replicouser REPLICATION LOGIN;  
ALTER ROLE postgres NOREPLICATION;
- ▶ pg\_hba.conf içinde replikasyon desteği  
host replication replicouser 10.1.1.2/32 md5



## Standby sunucusunda yapılması gerekenler

# Standby sunucuda yapılması gerekenler



- ▶ `hot_standby=on`

HS özelliğini açar

- ▶ `max_standby_archive_delay`

- ▶ `max_standby_streaming_delay`

Standby sunucu, elindeki veriyi xlog arşivinden ya da streaming replication'dan eşlerken sorguların en uzun süresini belirtir.



## Base backup alma



# Base backup alma



## ► Verinin fiziksel kopyası

Bu sürede veritabanını durdurmanıza gerek yoktur. PostgreSQL bu süreçte eksik olacak bilgileri xloglardan tamamlayacaktır.

```
postgres=# SELECT pg_start_backup('test');
```

```
pg_start_backup
```

```
-----
```

```
0/2000020
```

```
(1 row)
```

# Base backup alma



Ardından dosyaları uzaktaki sunucuya atalım. İşlem basit olarak şudur:

```
rsync -ave ssh /var/lib/pgsql/9.1/data/* \  
relika.sunucusu:/var/lib/pgsql/9.1/data/
```

xlogları da arşivlediğimiz için, oluşacak veri değişiklikleri bizi etkilemeyecektir. Yedek sürecini şimdi bitirebiliriz:

```
postgres=# SELECT pg_stop_backup();  
NOTICE: pg_stop_backup complete, all required WAL  
segments have been archived  
pg_stop_backup
```

```
-----  
0/20000D8
```

```
(1 row)
```



## recovery.conf dosyası



- ▶ **recovery.conf** dosyasının oluşturulması  
Postgresql.conf ile aynı dizinde olmalıdır.

```
standby_mode = on
```

```
primary_conninfo = 'host=192.168.100.1 port=5432  
user=replicauser password=mypass'
```

```
restore_command = '/bin/cp /var/lib/pgsql/xlogarchive/%f %p'
```

```
archive_cleanup_command = '/usr/pgsql-  
9.1/bin/pg_archivecleanup /var/lib/pgsql/xlogarchive %r'
```

```
trigger_file = '/var/lib/pgsql/9.1/data/finish.replication'
```

# Sırada ne var?



- Standby sunucuyu artık başlatabilirsiniz.
- Hadi uygulayalım!



## Sorular

# PostgreSQL'de Uygulamalı Gömülü Replikasyon (Streaming Replication ve Hot Standby)

Devrim GÜNDÜZ  
Principal Systems Engineer  
EnterpriseDB  
devrim.gunduz@EnterpriseDB.com