

# PostgreSQL'de Ölçeklenebilirlik, Yüksek Erişilebilirlik ve Yedeklilik

...

Atıf Ceylan  
@atifceylan



Yurdum Yazılım

# Sunum İçeriği

Ölçeklenebilirlik (Scalability)

Yedeklilik (Replication & Backup)

Yük Dengeleme & Yüksek Erişilebilirlik (Load Balancing & High Availability)

# Ölçeklenebilirlik

Artan kapasite ihtiyacının,  
performanstan ödün vermeden  
karşılanabilmesidir.



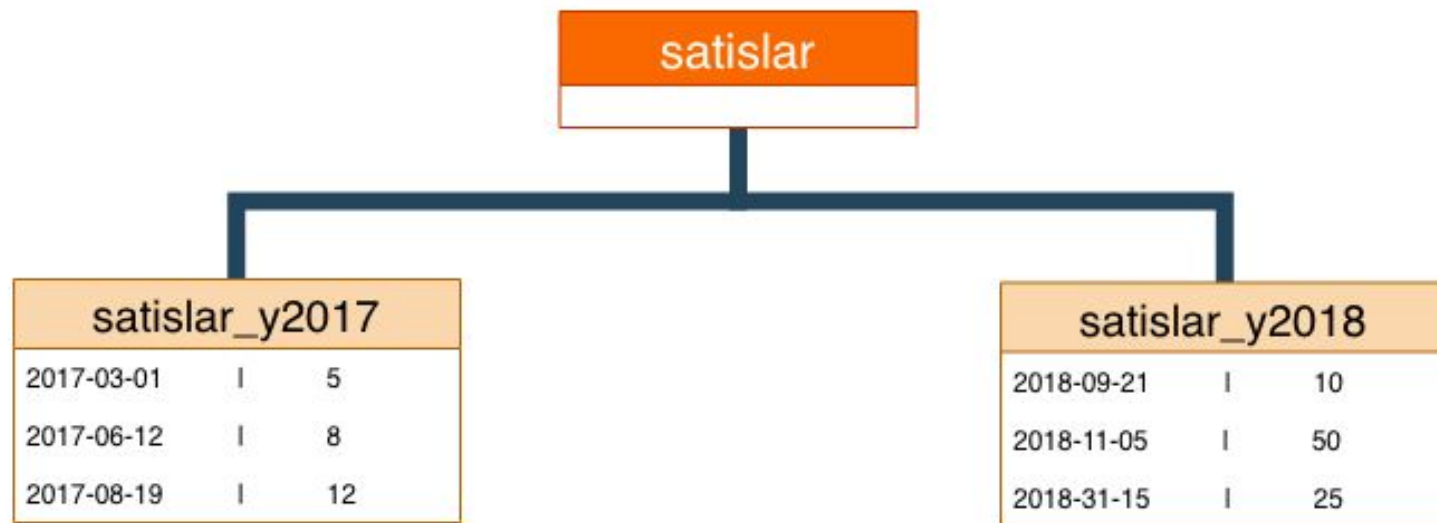
# Doğru Teşhis Doğru Reçete

Arkadaşının hastalığına iyi gelen ilaç  
senin için zehir olabilir.

- Uygulama/SQL optimizasyonu.
  - Düzgün konfigürasyon, doğru index kullanımı, metin araması için FTS kullanımı.
  - DBA, önemsemediğin kadar önemlidir.
-

# Partitioning

Büyük tablo verilerinin, belirli kriterlerle göre alt tablolara bölünmesidir.



# Partitioning

## Örnek:

```
CREATE TABLE satislar (  
    ekleme_tarihi    date not null,  
    satis_miktari     int  
) PARTITION BY RANGE (ekleme_tarihi);
```

```
CREATE TABLE satislar_y2017 PARTITION OF satislar  
FOR VALUES FROM ('2017-01-01') TO ('2018-01-01');
```

```
CREATE TABLE satislar_y2018 PARTITION OF satislar  
FOR VALUES FROM ('2018-01-01') TO ('2019-01-01');
```

# Partitioning

```
INSERT INTO satislar (ekleme_tarihi, satis_miktari) VALUES ('2017-03-01', 5);  
INSERT INTO satislar (ekleme_tarihi, satis_miktari) VALUES ('2017-06-12', 8);
```

```
INSERT INTO satislar (ekleme_tarihi, satis_miktari) VALUES ('2018-09-21', 10);  
INSERT INTO satislar (ekleme_tarihi, satis_miktari) VALUES ('2018-11-05', 50);
```

```
explain select * from satislar;
```

QUERY PLAN

```
-----  
Append (cost=0.00..87.80 rows=4520 width=8)  
  -> Seq Scan on satislar_y2017 (cost=0.00..32.60 rows=2260 width=8)  
  -> Seq Scan on satislar_y2018 (cost=0.00..32.60 rows=2260 width=8)  
(3 rows)
```

# Partitioning

```
EXPLAIN SELECT * FROM satislar WHERE ekleme_tarihi BETWEEN '2017-03-01' and '2017-10-01';
```

## QUERY PLAN

---

```
Append (cost=0.00..43.95 rows=11 width=8)
  -> Seq Scan on satislar_y2017 (cost=0.00..43.90 rows=11 width=8)
        Filter: ((ekleme_tarihi >= '2017-03-01'::date) AND (ekleme_tarihi <= '2017-10-01'::date))
(3 rows)
```

Partition table olmayan bir aralığa ekleme yapmak istediğimizde hata alırız.

```
INSERT INTO satislar (ekleme_tarihi, satis_miktari) VALUES ('2019-03-01', 5);
```

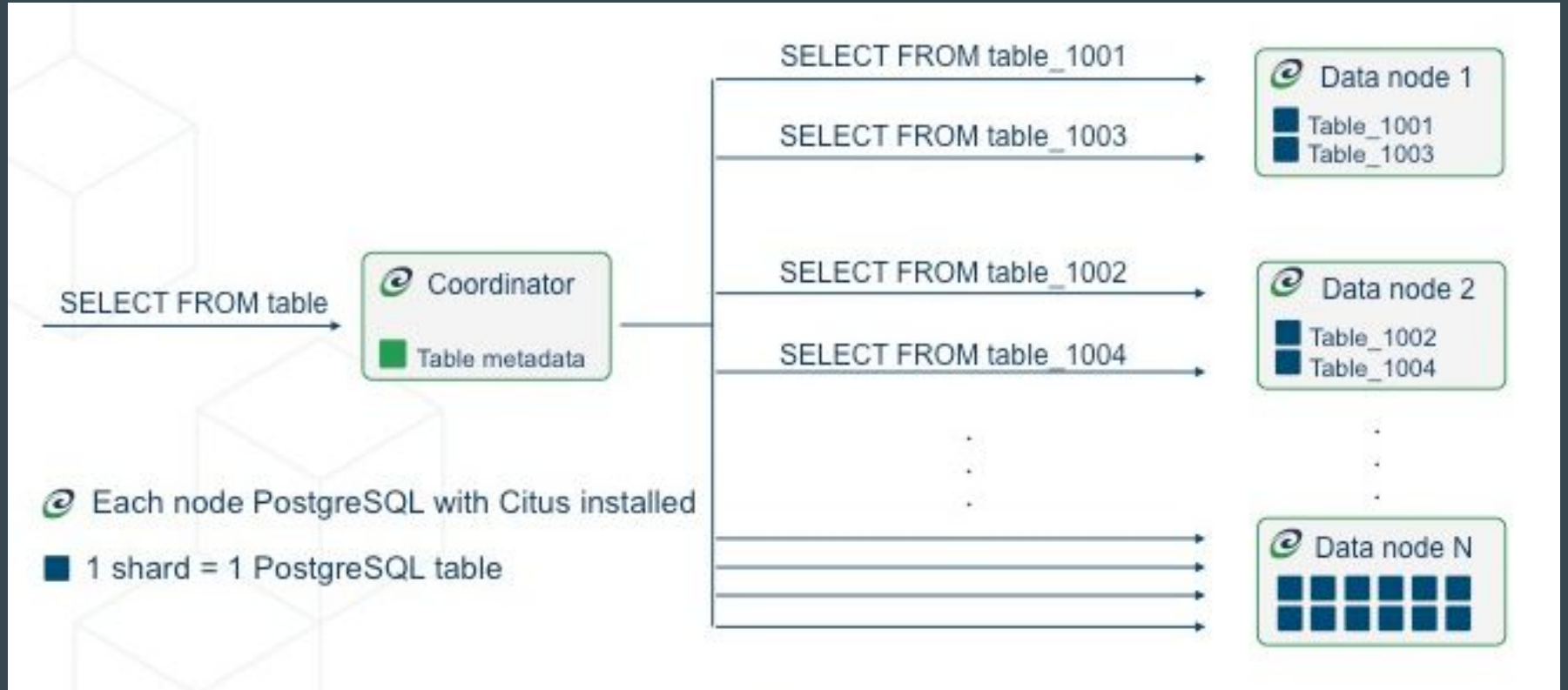
ERROR: no partition of relation "**satislar**" found for row

DETAIL: Partition key of the failing row contains (**ekleme\_tarihi**) = (2019-03-01).



# Sharding

Büyük tablo verilerinin, belirli kriterlerle (key) göre birden çok servise bölünmesidir.



# Citus

- Sharding
- Açık kaynak
- PostgreSQL Extension
- Distributed (dağıtık) Data ve SQL (parallelism) Desteği
- Distributed Aggregate
- Distributed Transaction
- Real-Time Analytics

# Citus Kullanan Büyük Veri Örnekleri

- **Heap**  
500+ billion events  
900TB of data on a 40-node Citus database cluster
- **Chartbeat**  
>2.6B rows of data added per month
- **Pex**  
30B rows updated/day  
20-node Citus database cluster on Google Cloud  
2.4TB memory, 1280 cores, and 60TB of data
- **Mixrank**  
1.6PB of time series data

# Citus

Co-Located Join

Update

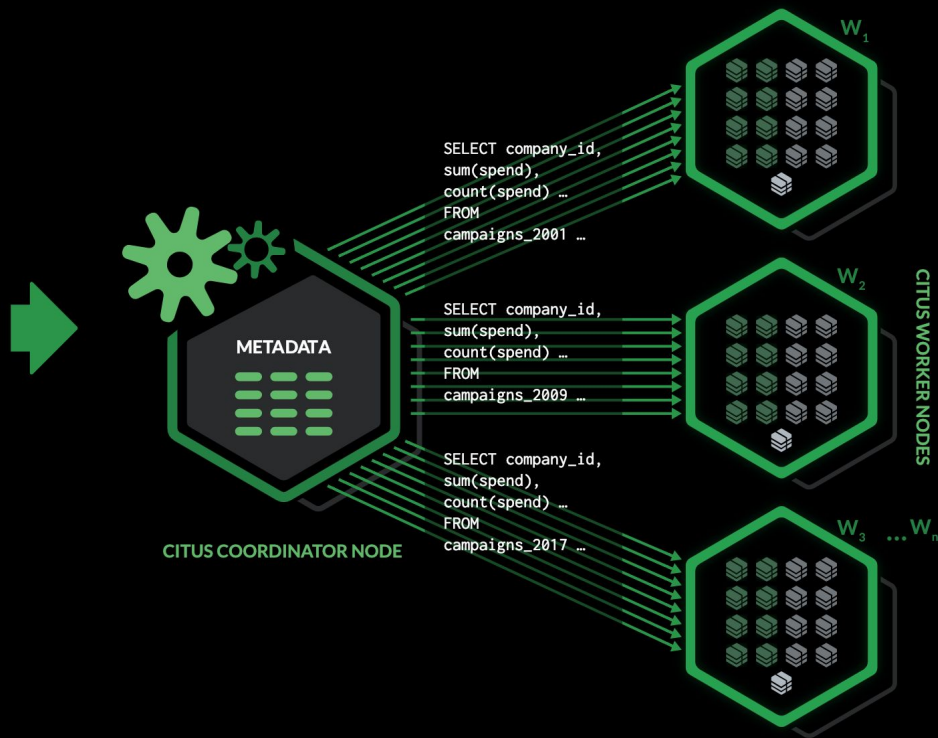
Distributed Aggregate

Distributed Transaction

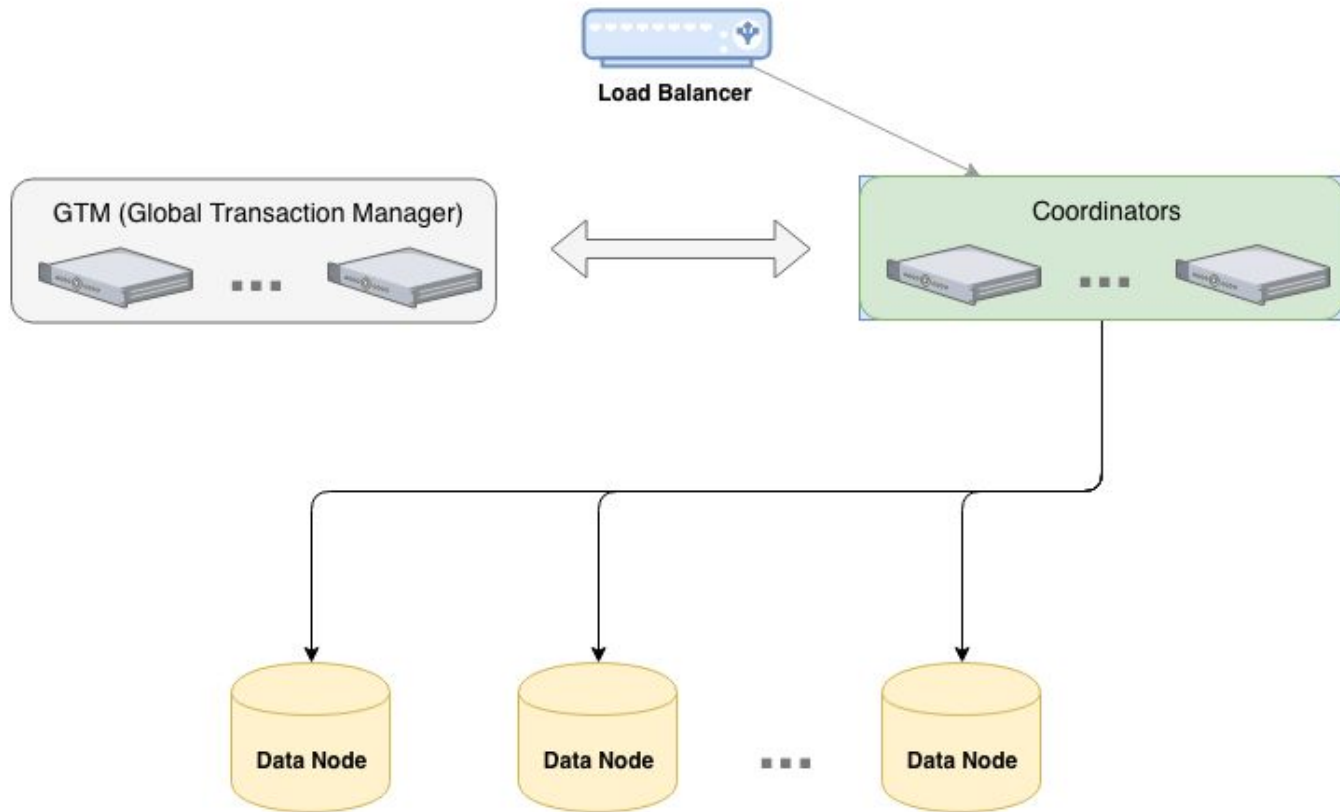
Schema Change

## APPLICATION

```
SELECT company_id,  
       avg(spend) AS avg_campaign_spend  
FROM campaigns  
GROUP BY company_id;
```



# PostgreSQL-XL



# PostgreSQL-XL

- Açık kaynak
- Tam ACID desteği
- Küme (cluster) çapında tutarlılık (consistency)
- Yazma ağırlıklı OLTP iş yükleri
- MPP ile OLAP desteği
- Key-value store (json)
- GIS (PostGIS)
- Replicated/Distributed çalışma modeli
- 16 data node'a kadar genişleyebilme

# Replikasyon (Master-Slave)

Fiziksel ve Mantıksal Replikasyon



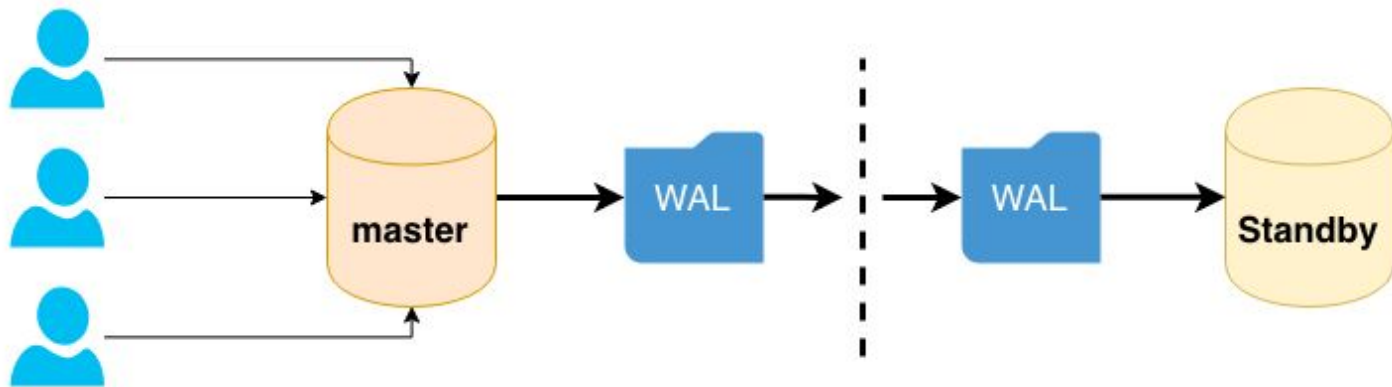
# Replikasyon (Master-Slave)

1. Fiziksel Replikasyon (physical replication)
  - a. WAL Transferi (log shipping)
  - b. Streaming Replication
    - i. Asenkron (asynchronous)
    - ii. Senkron (synchronous)
2. Mantıksal Replikasyon (logical replication)



# Physical Replication - Log Shipping

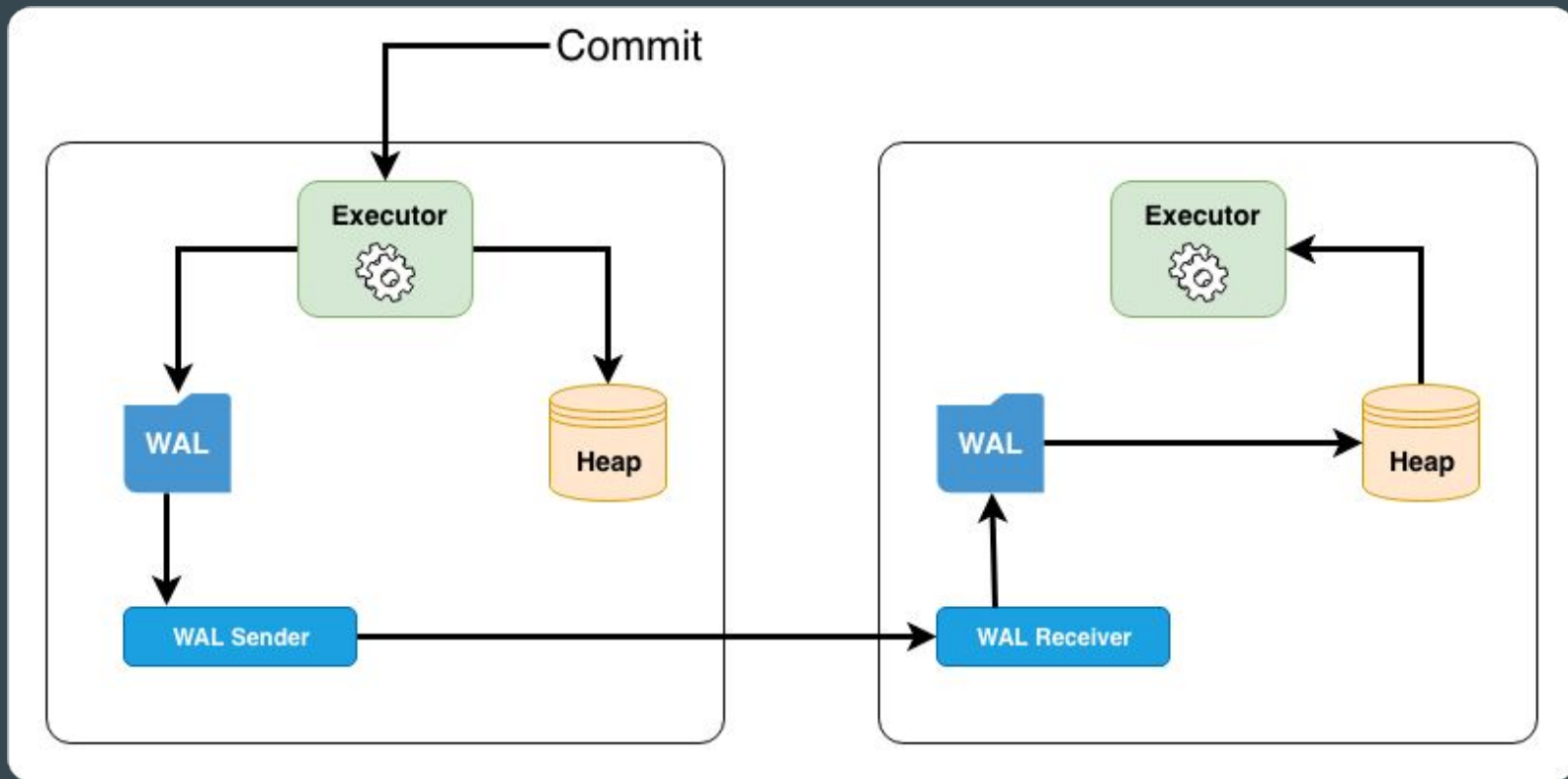
- `archive_mode`, `archive_command`
- `rsync`, `scp` vb. yöntemlerle WAL dosyaları transfer edilebilir
- NFS, GFS2, Samba vb. paylaşımlı disk/klasör yapıları kullanılabilir



# Physical Replication - Streaming Replication (asynchronous)

synchronous\_commit = **off** / **on** / **local**

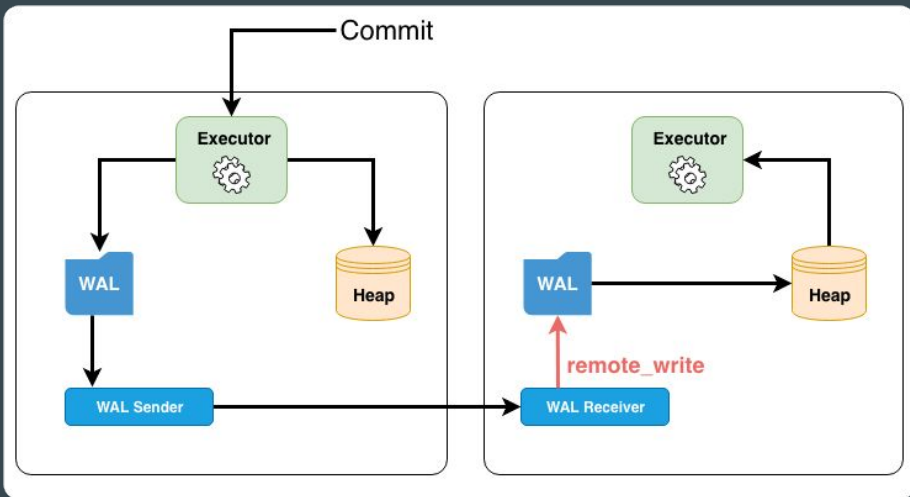
synchronous\_standby\_names ?



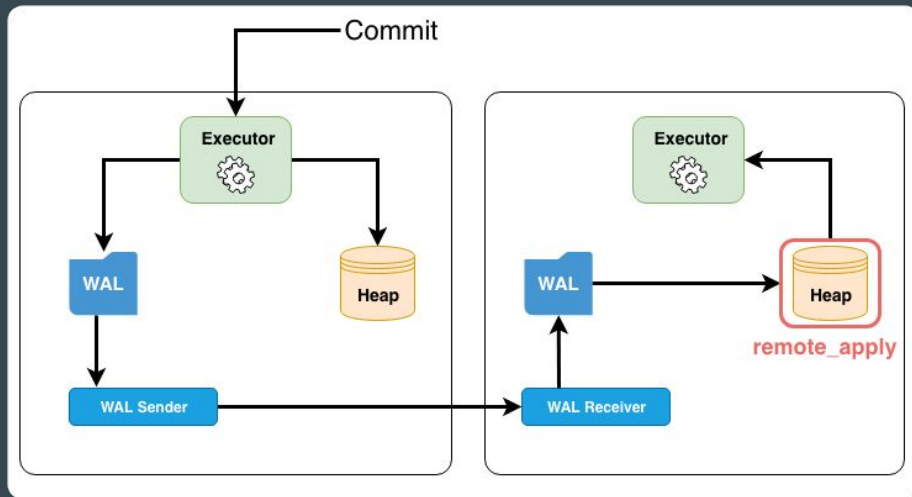
# Physical Replication - Streaming Replication (synchronous)

`synchronous_standby_names` parametresi doldurulmalıdır

`synchronous_commit = remote_write`



`synchronous_commit = remote_apply`



# Logical Replication

- Fiziksel replikasyondaki gibi raw block paylaşımı yerine statement/row tabanlı çalışır.
- Publication / Subscription yöntemi kullanılır.
- Fiziksel replikasyonda cluster görüntüsü replike edilirken, mantıksal replikasyonda veritabanı veya alt kümeleri (base tables) replike edilir.
- Şema değişiklikleri, DDL, sequence verisi, large objects, view, materialized view, partition root tabloları replike edilmez.

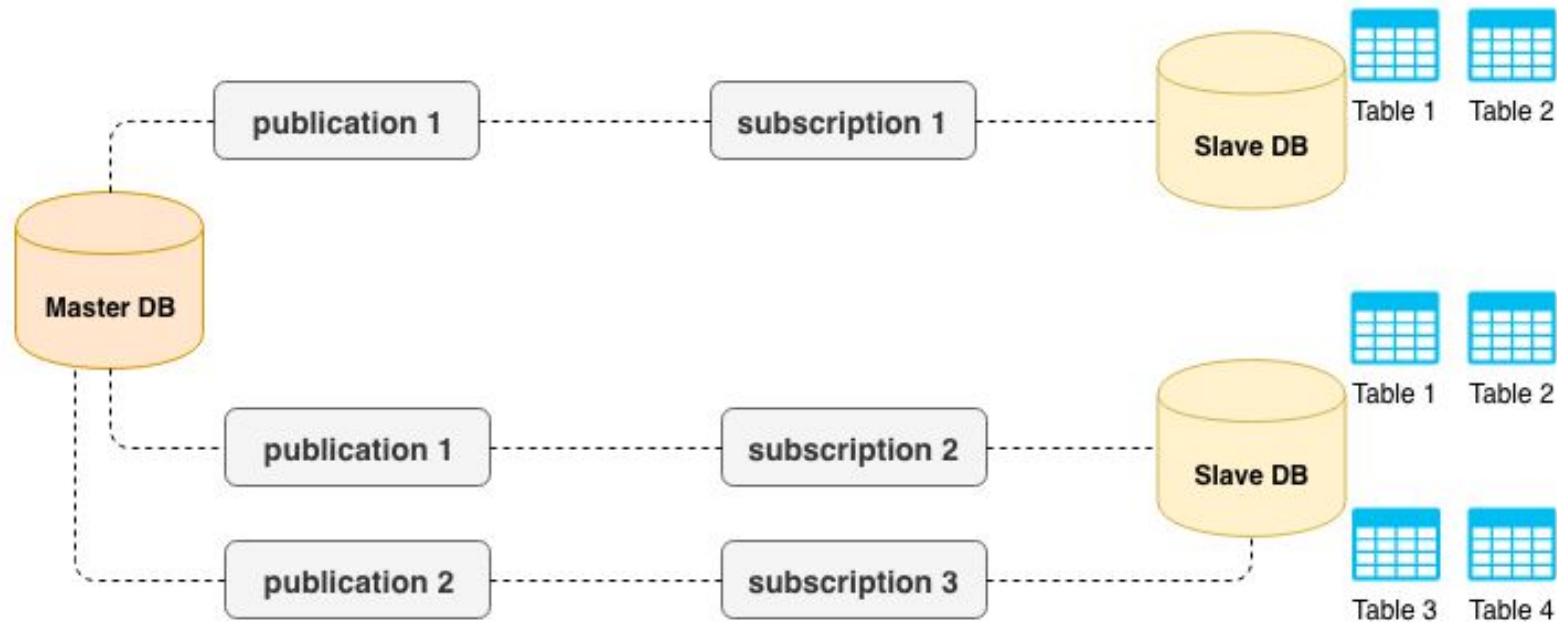
## **master:**

```
CREATE PUBLICATION mypub1 FOR TABLE article;
```

## **slave:**

```
CREATE SUBSCRIPTION mysub1 CONNECTION '<connection string>' PUBLICATION mypub1;
```

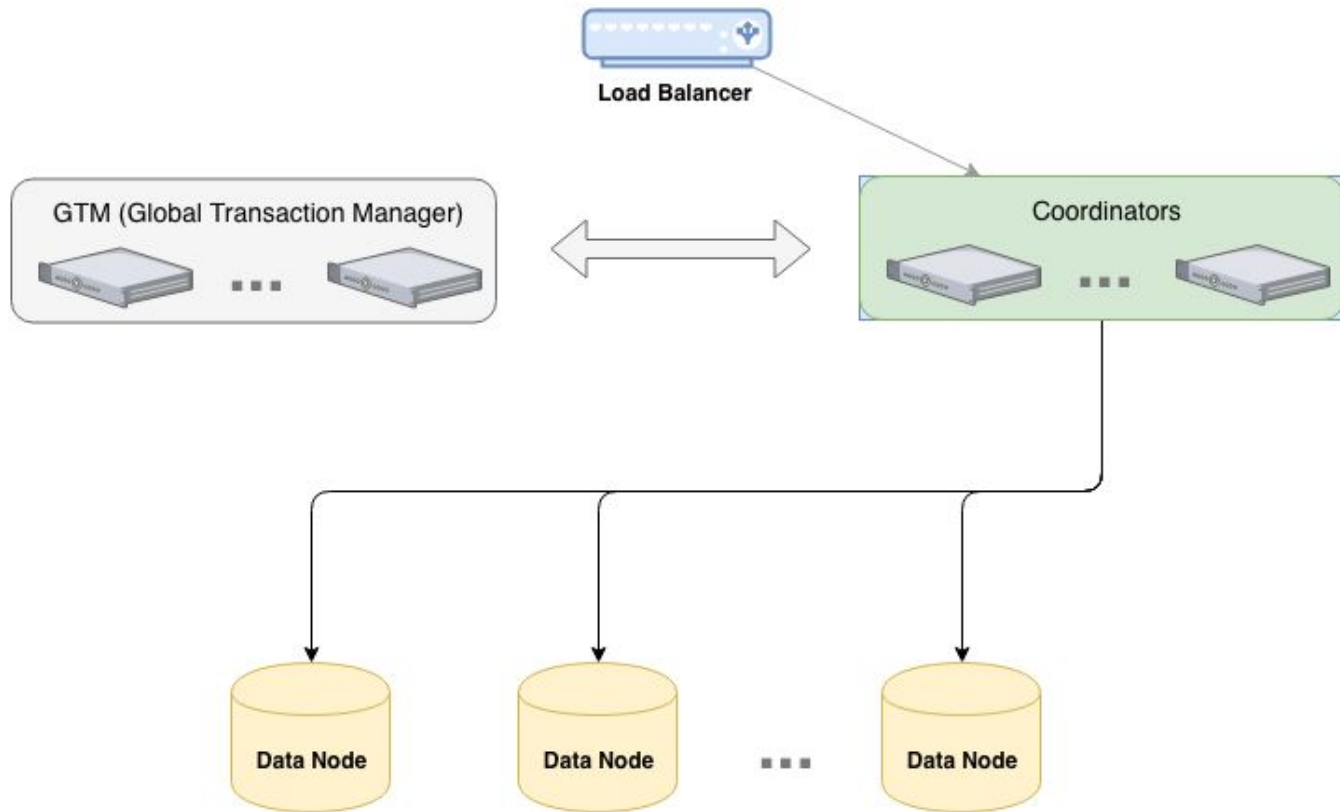
# Logical Replication



# Logical Replication Tercih Nedenleri

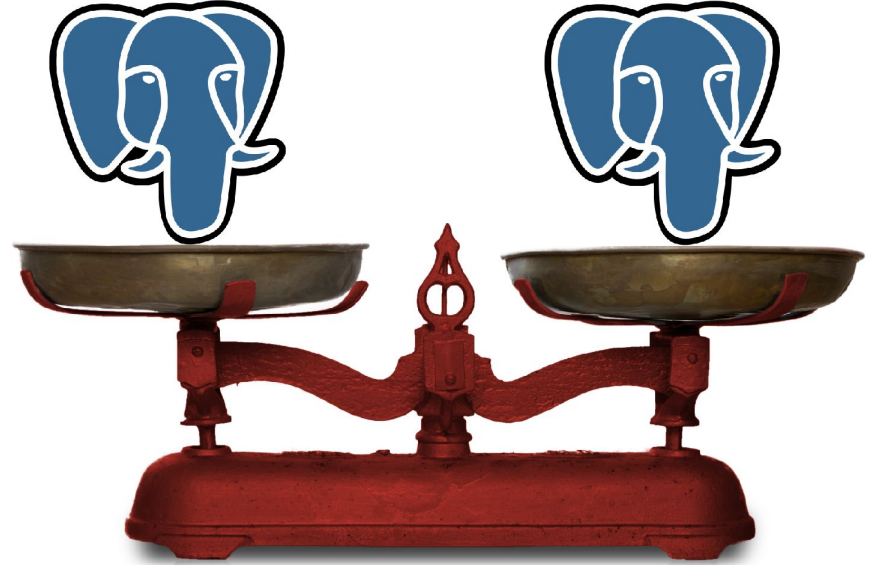
- Bir veritabanı veya alt kümelerine ait değişimleri abonelere (subscriber) gönderme
- Farklı majör versiyonlar arasında replikasyon olanağı
- Farklı platformlar arasında replikasyon olanağı
- Replike edilen veriler için farklı kullanıcı gruplarına yetki verilmesi
- Bir veritabanının alt kümelerini birden çok veritabanına replike edebilme

# PostgreSQL-XL



# Yük Dengeleme

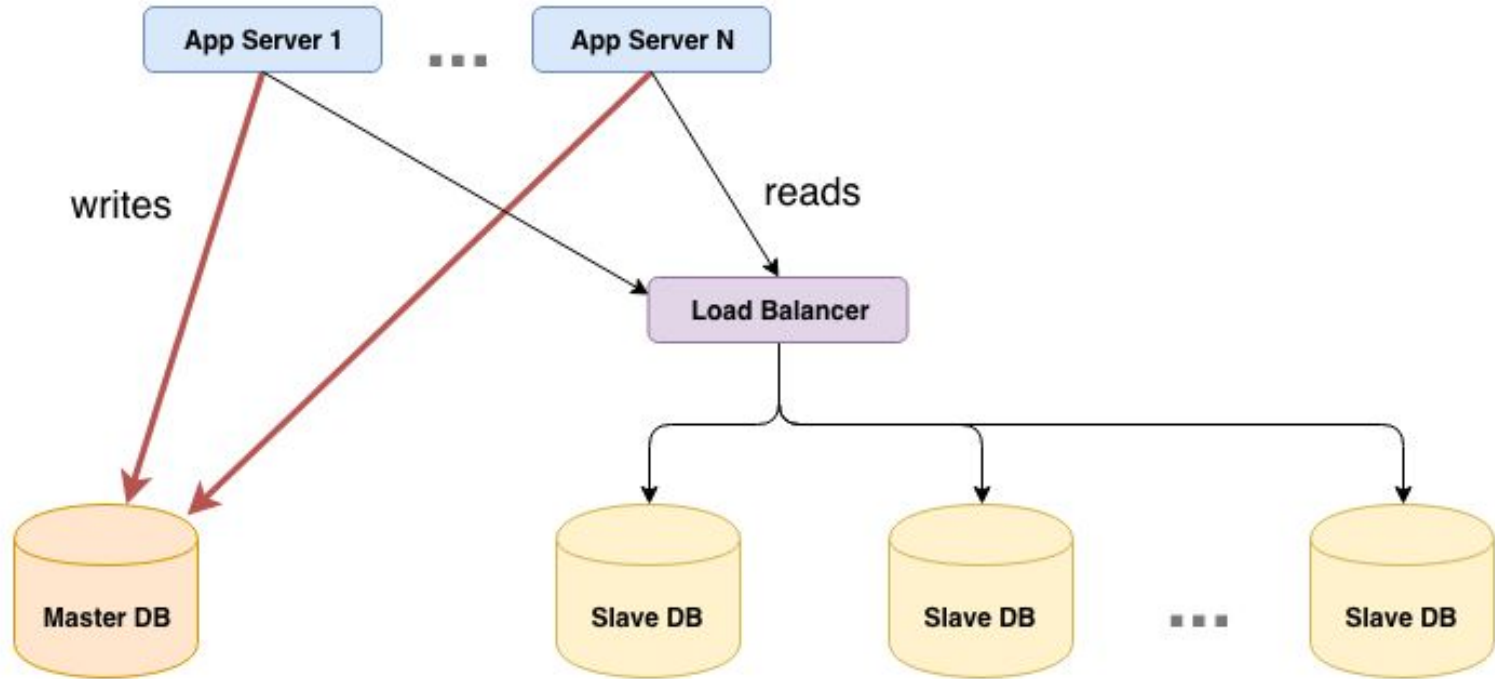
Okuma, yazma veya ortaya karışık



---



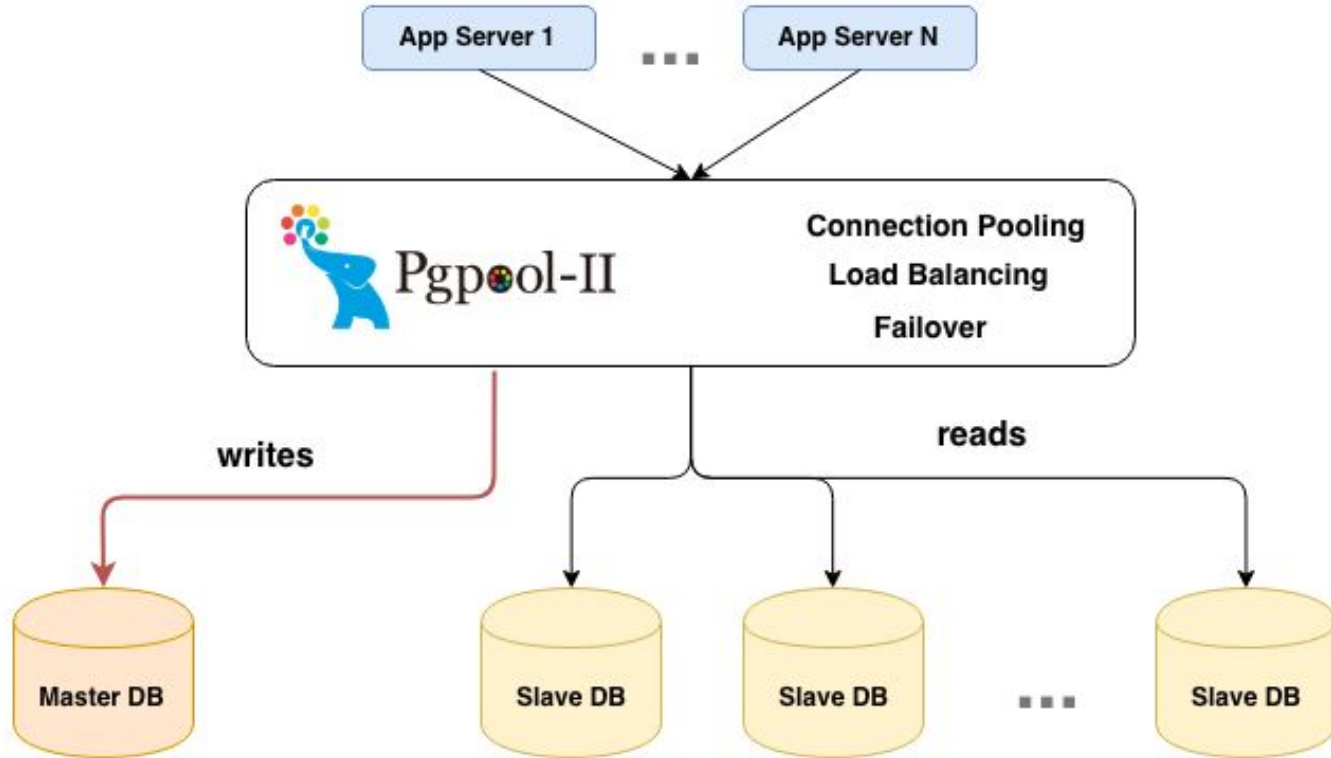
# Okuma Ağırlıklı Yük Dengeleme



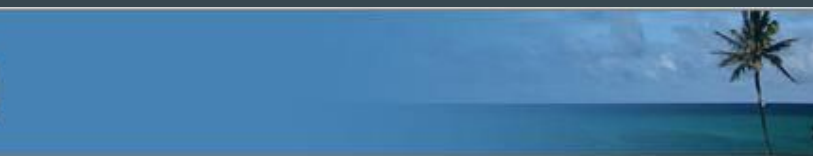

# pgpool-II

- Açık kaynak
- Connection Pooling
- Replikasyon
- Yük Dengeleme
- Otomatik Failover

# pgpool ile yük dengeleme



# pgpoolAdmin



- pgpool Status
- Node Status
- Query Cache
- Partitioning Rule
- pgpool.conf Setting
- pgpoolAdmin Setting
- Change Password
- Logout

[? Help](#)

pgpool Status

Summary

Process Info.

Node Info.

Node Info.

IP Address	Port	Status	Weight	
localhost	5432	Up, Connected	0.333	<div>Disconnect</div>
localhost	5433	Up, Connected	0.333	<div>Disconnect</div>
localhost	5434	Up, Connected	0.333	<div>Disconnect</div>

Summary

Process Info.

Node Info.

pgpool

Stop pgpool

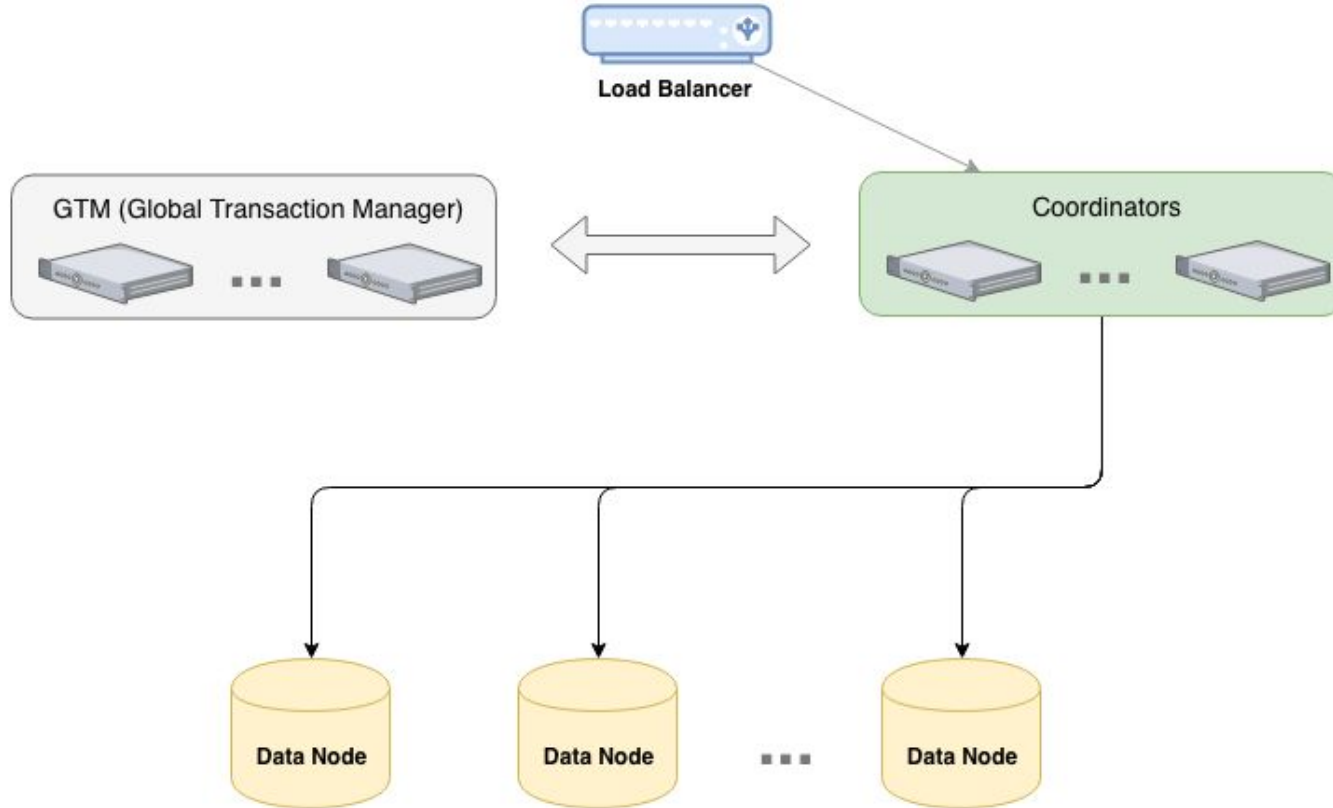
Restart pgpool

Reload

Version 2.0

Copyright © 2006 - 2007 pgpool Global Development Group. All rights reserved.

# PostgreSQL-XL ile Yazma Ağırlıklı Yük Dengeleme



# Teşekkür

...

Atıf Ceylan  
@atifceylan



Yurdum Yazılım