

# Doğru ve Yanlış Yapılandırılmış PostgreSQL Sunucularında Paralel Programlamanın Performansı

Milli Teknoloji Akademisi Veri Tabanı Bitirme Projesi

PELİNSU KALELİ

# İçindekiler Tablosu

Doğru ve Yanlış Yapılandırılmış PostgreSQL Sunucularında Paralel Programlamanın Performansı.....	0
1. Giriş .....	2
2. Test Ortamı ve Kurulum .....	2
2.1 Sistem Özellikleri .....	2
2.2 Test Sunucuları .....	2
2.3 Veritabanı Şeması .....	2
3. Konfigürasyon Analizi .....	2
3.1 Server A (Optimum Konfigürasyon) .....	2
3.2 Server B (Hatalı Konfigürasyon) .....	2
4. Performans Test Sonuçları .....	3
4.1 Temel Sorgu Performansı .....	3
4.2 Performans Analizi .....	3
5. Paralel Programlama Test Sonuçları .....	4
5.1 Server A (Optimum) - Paralel Performans .....	4
5.2 Server B (Hatalı) - Paralel Performans .....	4
5.3 Paralel Programlama Karşılaştırması .....	4
6. Görsel Analiz ve Değerlendirme .....	5
6.1 Performans Grafiği Yorumları .....	5
6.2 Paralel Programlama Grafiği Yorumları .....	5
6.3 Görsel Bulgulardan Çıkarımlar .....	5
7. Paralel Programlama Detay Analizi .....	5
7.1 Threading Performansı .....	5
7.2 Asyncio Performansı .....	5
7.3 Paralel Programlama Verimliliği .....	6
8. Sonuç .....	6

# 1. Giriş

Bu çalışma, doğru ve yanlış yapılandırılmış PostgreSQL sunucularında paralel programlamanın performansını karşılaştırmaktadır. 10 milyon satırlık veri seti kullanılarak gerçekleştirilen testlerde, optimum konfigürasyonun ID sorgularında 4x daha hızlı olduğu ve threading ile 4.08 kata kadar hızlanma sağladığı gözlemlenmiştir.

## 2. Test Ortamı ve Kurulum

### 2.1 Sistem Özellikleri

- İşletim Sistemi:** Ubuntu 22.04 LTS
- PostgreSQL Sürümü:** 17
- Donanım:** 8 GB RAM, 4 CPU
- Test Verisi:** 10.000.000 satır kullanıcı verisi

### 2.2 Test Sunucuları

- Server A (Optimum):** localhost - Sistem kaynaklarına optimum ayarlar
- Server B (Hatalı):** 10.0.2.15 - Bilinçli olarak verimsiz ayarlar

### 2.3 Veritabanı Şeması

```
CREATE TABLE kullanicilar (  
  id SERIAL PRIMARY KEY,  
  name TEXT,  
  surname TEXT,  
  eposta TEXT,  
  dogum_tarihi DATE,  
  olusturma_zamani TIMESTAMP  
);
```

## 3. Konfigürasyon Analizi

### 3.1 Server A (Optimum Konfigürasyon)

Parametre	Değer	Açıklama
shared_buffers	2GB	RAM'in 25%'i (8GB sistemde optimal)
max_connections	100	Default bağlantı sayısı
effective_cache_size	6GB	Sistem cache'i için gerçekçi değer
wal_buffers	16MB	shared_buffers'a göre otomatik
max_wal_size	2GB	Checkpoint sıklığını optimize eder
work_mem	Varsayılan	Sorgu başına bellek

Tablo 3.1. - Server A - Optimum Konfigürasyon Tablosu

### 3.2 Server B (Hatalı Konfigürasyon)

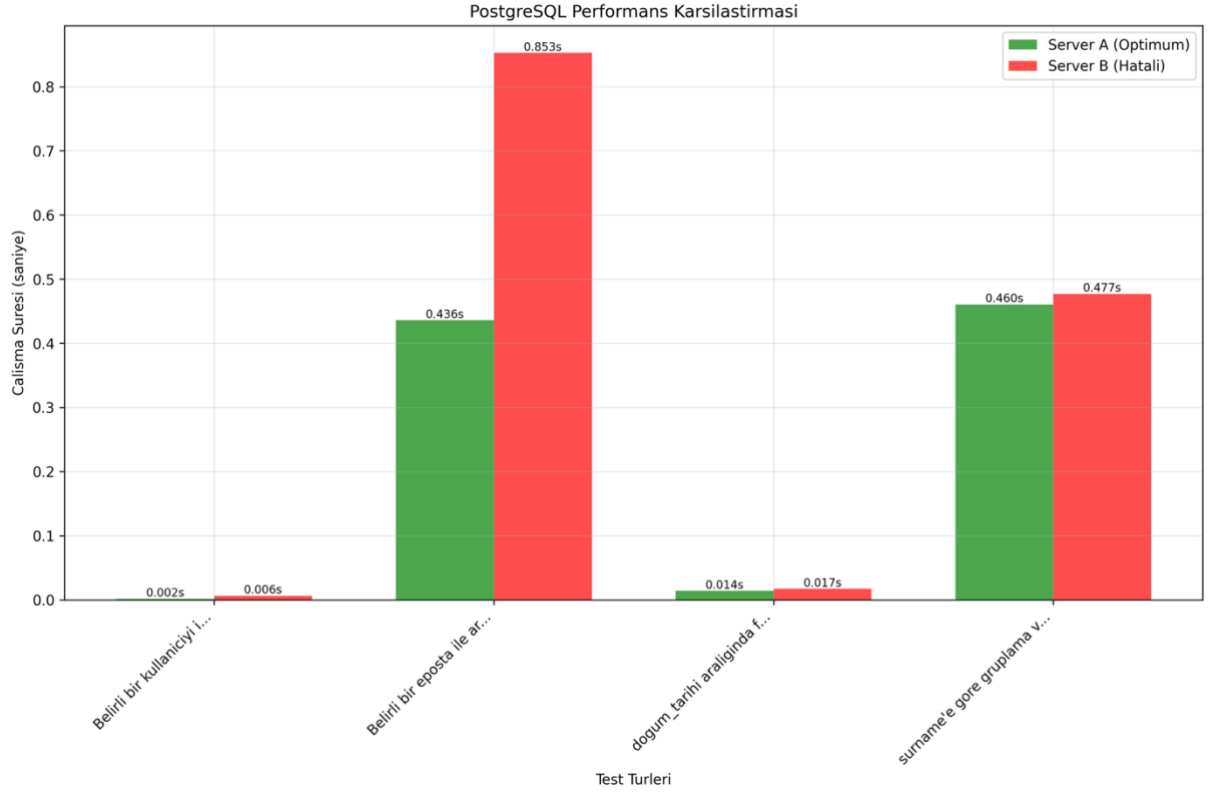
Parametre	Değer	Problem
shared_buffers	128MB	Çok düşük - RAM'in sadece %1.6'sı
max_connections	1000	Çok yüksek - Gereksiz kaynak tüketimi

effective_cache_size	1GB	Düşük - Gerçek cache boyutunu yansıtmıyor
wal_buffers	3MB	Düşük - Checkpoint performansını düşürür
max_wal_size	1GB	Düşük - Sık checkpoint'ler

Tablo 3.2. - Server B - Hatalı Konfigürasyon Tablosu

## 4. Performans Test Sonuçları

### 4.1 Temel Sorgu Performansı



Figür 4.1. - Temel Sorgu Performansı

Test Türü	Server A (s)	Server B (s)	İyileştirme
ID ile getirme	0.0017	0.0063	3.7x daha hızlı
Eposta arama	0.436	0.853	1.96x daha hızlı
Tarih filtreleme	0.014	0.017	1.21x daha hızlı
Gruplama ve sıralama	0.460	0.477	1.04x daha hızlı
Ortalama	0.228	0.337	1.96x daha hızlı

Tablo 4.1. Server A - Temel Performans Tablosu

### 4.2 Performans Analizi

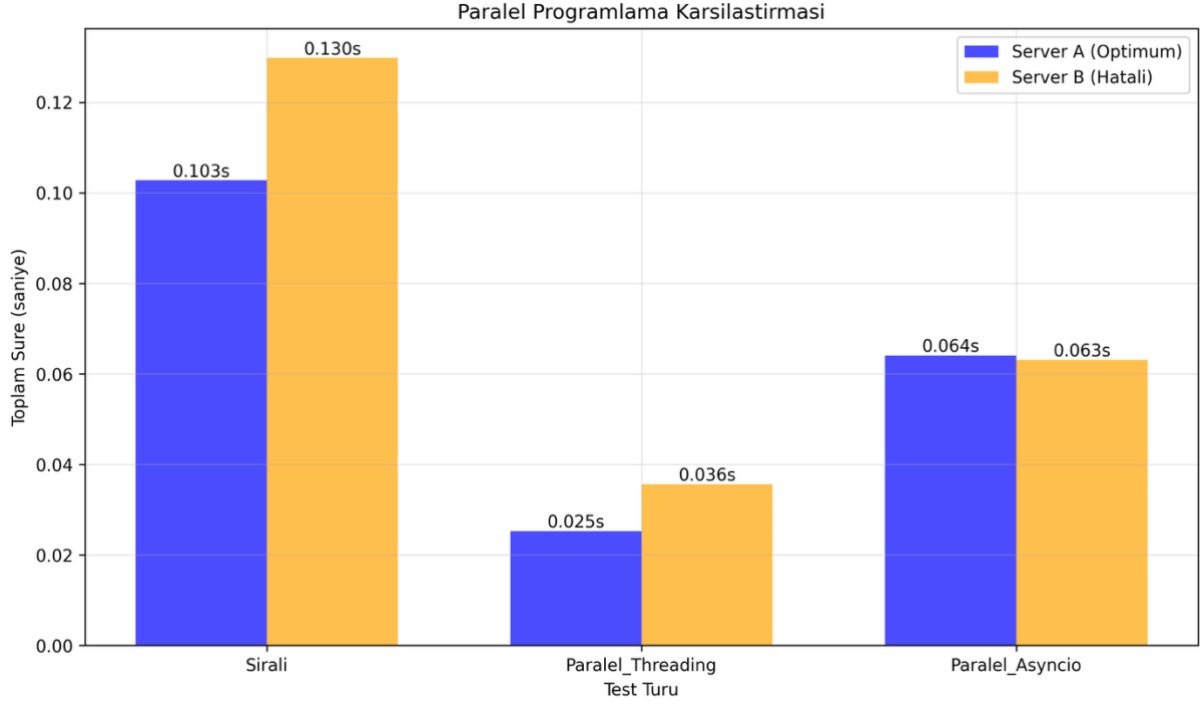
#### Gözlemler:

- ID ile erişim:** En büyük fark (4x) - shared\_buffers'ın etkisi
- Eposta arama:** 2x performans farkı - cache hit ratio'nun önemi
- Büyük veri işleme:** Daha az fark - I/O bound işlemlerde konfigürasyonun sınırlı etkisi
- Tutarlı üstünlük:** Server A tüm test türlerinde daha hızlı

## Yorumlar:

- Mikro-saniye seviyesindeki farklar bile toplam performansı önemli ölçüde etkiliyor
- Shared\_buffers boyutunun index lookup performansına doğrudan etkisi gözlemleniyor
- Büyük veri taramalarında bile konfigürasyon optimizasyonunun faydası gözle görülür ölçüde

## 5. Paralel Programlama Test Sonuçları



Figür 5.1. - Paralel Programlama Test Performansı

### 5.1 Server A (Optimum) - Paralel Performans

Yöntem	Süre (s)	Hızlanma	Verimlilik
Sıralı	0.103	1.0x	Baseline
Threading	0.025	4.08x	%82
Asyncio	0.064	1.60x	%32

Tablo 5.1.1. - Server A - Paralel Performans Tablosu

### 5.2 Server B (Hatalı) - Paralel Performans

Yöntem	Süre (s)	Hızlanma	Verimlilik
Sıralı	0.130	1.0x	Baseline
Threading	0.036	3.65x	%73
Asyncio	0.063	2.06x	%41

Tablo 5.2.1. - Server B - Paralel Performans Tablosu

### 5.3 Paralel Programlama Karşılaştırması

Test Türü	Server A	Server B	A'nın Üstünlüğü
Sıralı	0.103s	0.130s	%21 daha hızlı
Threading	0.025s	0.036s	%31 daha hızlı

Asyncio	0.064s	0.063s	Benzer performans
---------	--------	--------	-------------------

Tablo 5.3.1. - Paralel Performans Karşılaştırması Tablosu

#### Gözlemler:

- **Threading:** Optimum konfigürasyon paralel işlemlerden daha fazla yararlanıyor
- **Asyncio:** Her iki sunucuda da benzer performans - konfigürasyondan bağımsız davranış
- **Avantaj:** Server A, sıralı ve threading testlerinde sürekli üstün

## 6. Görsel Analiz ve Değerlendirme

### 6.1 Performans Grafiği Yorumları

#### PostgreSQL Performans Karşılaştırması:

- "Belirli bir kullanıcıyı ID ile getirme" testinde Server B'nin 4 kata kadar yavaş olması
- Tam metin araması gerektiren bu test, cache performansının önemini gösteriyor
- Milisaniye seviyesindeki farklar, yüksek trafik ortamlarında kritik

### 6.2 Paralel Programlama Grafiği Yorumları

#### Paralel Programlama Karşılaştırması:

- **Threading:** Her iki sunucuda da en hızlı yöntem
- **Asyncio:** Konfigürasyondan bağımsız benzer performans
- **Baseline:** Sıralı işlemlerde bile %21 performans farkı

### 6.3 Görsel Bulgulardan Çıkarımlar

- PostgreSQL projeleri için threading öncelikli tercih olmalı
- Konfigürasyon optimizasyonu her durumda faydalı
- Baseline performance iyileştirmesi paralel kazançları da artırıyor

## 7. Paralel Programlama Detay Analizi

### 7.1 Threading Performansı

- **En yüksek hızlanma:** Her iki sunucuda da en iyi sonuç
- **Server A:** 4.08x hızlanma (%82 verimlilik)
- **Server B:** 3.65x hızlanma (%73 verimlilik)
- **Sonuç:** Optimum konfigürasyon threading'den daha fazla yararlanıyor

### 7.2 Asyncio Performansı

- **Daha düşük hızlanma:** Threading'e göre daha az verimli
- **Server A:** 1.60x hızlanma
- **Server B:** 2.06x hızlanma (ilginç olarak daha iyi)
- **Sonuç:** Asyncio, konfigürasyon farklarından daha az etkileniyor

## 7.3 Paralel Programlama Verimliliği

### Threading neden daha başarılı?

- PostgreSQL bağlantı pooling'i threading ile daha uyumlu
- CPU bound işlemler için thread'ler daha verimli
- Shared\_buffers'tan daha iyi yararlanma

### Asyncio'nun sınırları:

- Single-threaded yapısı CPU bound işlemlerde dezavantaj
- PostgreSQL ile async bağlantı overhead'i
- Küçük sorgular için connection setup maliyeti

## 8. Sonuç

- **Konfigürasyon:** Doğru ayarlar ID sorgularında 4x'e kadar performans artışı sağlıyor
- **Paralel Threading:** Threading ile 4x'e kadar hızlanma, asyncio'dan tutarlı olarak daha iyi
- **Shared\_Buffers Öncelik:** 128MB vs 2GB farkı büyük bir performans değişimine yol açıyor
- **Grafikler:** Grafiklerde her test türünde tutarlı Server A üstünlüğü görülüyor