

# İleri Seviyede PostgreSQL Yönetimi

Devrim GÜNDÜZ

PostgreSQL Geliştiricisi

PostgreSQL DBA @ Markafoni

devrim@gunduz.org

devrim@postgresql.org[.tr]

<http://twitter.com/devrimgunduz>

# İleri Seviyede PostgreSQL Yönetimi

Bu seminer 1 saatte bitmez  
sanıyorum; 3 saatlik  
seminer slotu istiyoruz! :)

# Aklıma gelmişken...

- Markafoni'de çalışmak ister misiniz? Java, Python, Django, PostgreSQL gibi sözcüklerden birisi ya da birden fazlası sizin için anlamlı ise [ik@markafoni.com](mailto:ik@markafoni.com) adresine özgeçmişlerinizi gönderebilirsiniz.

# Ajanda

1. PostgreSQL yönetim temelleri
2. Donanım
3. İşletim sistemi
4. PostgreSQL bileşenleri
  1. Bellek kullanımı
  2. Bgwriter
  3. Checkpoint
  4. İstatistiklerin kullanımı
  5. Autovacuum

# Ajanda

1. PostgreSQL yönetim temelleri
2. Donanım
3. İşletim sistemi
4. PostgreSQL bileşenleri
  1. Bellek kullanımı
  2. Bgwriter
  3. Checkpoint
  4. İstatistiklerin kullanımı
  5. Autovacuum

# PostgreSQL yönetimi temelleri

- Sistem yönetimi
- Donanım bilgisi
- Network bilgisi
- ...
- Veritabanı yönetimi

# PostgreSQL yönetimi temelleri

- Tüm ekip ile beraber çalışabilme
- Teknolojileri düzgün takip etme
- ...

# Ajanda

1. PostgreSQL yönetim temelleri
2. Donanım
3. İşletim sistemi
4. PostgreSQL bileşenleri
  1. Bellek kullanımı
  2. Bgwriter
  3. Checkpoint
  4. İstatistiklerin kullanımı
  5. Autovacuum



# PostgreSQL Donanımı

- Genel kural: Üretim ortamı yükleri gerçekten hiçbir şekilde kestirilmez.
- Donanım planlaması yapmak bu açıdan zordur.
- Donanım alırken ilerisini de düşünerek hareket etmek gerekli.

# PostgreSQL donanımı

- I/O
  - En önemli sorun
  - Veritabanları I/O bağımlıdır.
- RAID!
  - RAID 0
  - RAID 1
  - RAID 5
  - RAID 10
- Donanımsal ve yazılımsal RAID

# PostgreSQL Donanımı

- BBU (Battery Backed Unit)
  - Olmazsa olmaz
  - Arttırılmış başarıım
  - Arttırılmış veri güvenliği
- RAID kartları
  - HP P400 ve üzeri (400, 410 ve 800)
  - LSI

# PostgreSQL Donanımı

- Disk
  - Bitmeyen tartışma: SATA ve SAS
  - Para mutluluk getirir :-)
  - 6 Gbit'lik SAS diskler ile sırt ağrılarına son.
- Fiber? ISCSI?
- DAS ve SAN

# PostgreSQL donanımı

## İşlemci

- 64 bit!
- Giriş kapısını geniş tutun: Cache
- Bol miktarda core
- RAM
  - Evdeki makinamda 8, işyerindekinde de 16 GB RAM var.
  - RAM ucuz. Bol bol alın.
- Anahtar kural: Bilgiyi işlemcinin yakınında tutun.

# Ajanda

1. PostgreSQL yönetim temelleri
2. Donanım
3. İşletim sistemi
4. PostgreSQL bileşenleri
  1. Bellek kullanımı
  2. Bgwriter
  3. Checkpoint
  4. İstatistiklerin kullanımı
  5. Autovacuum

# İşletim Sistemi

- İşletim sistemi seçim kriterleri
  - Kurumsal destek
  - 64 bit
  - Uzun süreli destek

# İşletim sistemi

- Linux!
  - RHEL/CentOS
  - Ubuntu LTS/Debian
  - SLES
- Solaris bir başka alternatif.
- Üretim ortamında Windows, Ubuntu, Fedora ve OpenSuSE'den uzak durmak iyi olacaktır.



# İşletim sistemi

- Scheduler
  - deadline scheduler şu anda çok dedicated ortamlarda daha iyi başarımlar sunuyor.
  - cfq ise paylaşılmış ortamlarda iyi bir denge sağlıyor.
  - grub.conf: elevator=deadline

# Dosya sistemi seçimi

- Linux size birçok alternatif sunuyor.
- Ext2, ext3, ext4 ve diğerleri.
- Partitioning
  - Xloglar **mutlaka** ayrı bir diskte olmalıdır. Sıralı yazma ve okuma başarımı önemlidir.
  - Data ayrı bir disk grubuna gitmelidir. Raslansal olarak yazma ve okuma hızlı olmalıdır.
  - Yoğun ortamlarda indexler ayrı disklere gidebilir.

# Dosya sistemi

- Xloglar ext2 **olabilir**. Xlogların zaten kendisi journal'dır. Bu sayede %40 başarımlar artışı sağlarsınız. xloglar RAID1 olmalıdır. SATA kullanıyorsanız 4 disk ile RAID 10 yapıp başarımlar dertlerini çözebilirsiniz.
- Eğer BBU varsa data için data=writeback kullanın, başarımlar artsın.
- Hadi benchmark görelim :)

# Ajanda

1. PostgreSQL yönetim temelleri
2. Donanım
3. İşletim sistemi
4. PostgreSQL bileşenleri
  1. Bellek kullanımı
  2. Bgwriter
  3. Checkpoint
  4. İstatistiklerin kullanımı
  5. Autovacuum

# PostgreSQL bellek kullanımı

- Bellek kullanan 3 yer var:
  - Bakım işlemleri  
(maintenance\_work\_mem)
  - Shared buffer (shared\_memory)
  - Çalışma anında sorgu bazlı kullanılan bellek (work\_mem)

# PostgreSQL'in bellek kullanımı

- Shared memory ayrıntıları
  - Shmmax!
  - İdeal değer ve yüksek shared\_buffers sorunları
- work\_mem
  - ORDER BY, DISTINCT, agregateler, hash join
  - Dikkat: **Her** işlem için bu bellek ayrı ayrı kullanılacaktır.
  - İdeal ayar
  - log\_temp\_files (8.3+)

# PostgreSQL'in bellek kullanımı

- Örnek bir log\_temp\_files çıktısı:
- Mar 5 06:00:04 dbhost  
postgres[21556]: [16-1]  
user=xxx,db=xxx LOG: temporary  
file: path  
"base/pgsql\_tmp/pgsql\_tmp21556.0",  
size 146538496
- Bu sorgular için work\_mem özel olarak ayarlanmalıdır:  
SET work\_mem TO '150MB';  
SELECT ...

# PostgreSQL'in bellek kullanımı

- EXPLAIN ANALYZE ile work\_mem analizi yapabilirsiniz.

Örnek:

Sort Method: **external merge** Disk:  
15672kB

ya da

Sort method: **quicksort**  
Memory:1826Kb



# PostgreSQL'in bellek kullanımı

- maintenance\_work\_mem
  - CREATE INDEX, VACUUM, ALTER TABLE ADD FOREIGN KEY
  - İdeal değer?
  - Autovacuum'a dikkat!

# Ajanda

1. PostgreSQL yönetim temelleri
2. Donanım
3. İşletim sistemi
4. PostgreSQL bileşenleri
  1. Bellek kullanımı
  2. Bgwriter
  3. Checkpoint
  4. İstatistiklerin kullanımı
  5. Autovacuum

# Bgwriter

- Background writer
- Dirty buffer kavramı
- Yararları:
  - Sunucu süreçlerinin daha az dirty buffer temizleyecek olmaları
  - Daha az ve yumuşak checkpoint baskısı
- Olumsuz tarafı: Sürekli ve düzenli bir I/O baskısı çıkartması
- Olumsuz mu demiştik? Aslında değil.

# Bgwriter

- Bgwriter parametreleri:
  - `bgwriter_delay`
  - `bgwriter_lru_maxpages`
  - `bgwriter_lru_multiplier`

# Bgwriter

- Kullanılan I/O'yu hesaplamak:
- $1000 / \text{bgwriter\_delay} * \text{bgwriter\_lru\_maxpages} * 8192 =$   
Kullanılan I/O (byte cinsinden)
- Öntanımlı değerlerle:  
 $1000/200*100*8192 = 4\text{MB}$
- Bu hesap %100 doğru değildir. Ancak %100'e yakındır.

# Bgwriter

- pg\_stat\_bgwriter (8.3+)
  - checkpoints\_timed
  - checkpoints\_req
  - checkpoint\_timeout
  - buffers\_checkpoint
  - buffers\_clean
  - maxwritten\_clean
  - buffers\_backend
  - buffers\_alloc

# WAL

- Write Ahead Log
- PostgreSQL'in transaction logu
  - Her işlem xloga yazılmaz.
- 2 şekilde flush edilir:
  - Sync commit ve wal writer processleri ile

# WAL parametreleri

- fsync
- wal\_sync\_method
- full\_page\_writes
- wal\_buffers
- wal\_writer\_delay



# Bgwriter, Checkpoint ve WAL

- Keşke data hep ramde kalsa (savaşlar çıkmasa, her yer sevgi dolsa, lay lay lom)
- İdeal dünya üstteki gibidir. İdeal dünya olmayacaktır, ama elimizden geldiği kadar o noktaya yaklaşmalıyız.
- Checkpoint işlemine mümkün olduğu kadar az yük bırakmak gerekli.
- Wal yazma işlemi wal writer sürecine bırakmak gerekli.

# Ajanda

1. PostgreSQL yönetim temelleri
2. Donanım
3. İşletim sistemi
4. PostgreSQL bileşenleri
  1. Bellek kullanımı
  2. Bgwriter
  3. Checkpoint
  4. İstatistiklerin kullanımı
  5. Autovacuum

# Checkpoint

- Bgwriter'ın bir alt sürecidir.
- Xloglarla doğrudan ilgisi **yoktur**.
- Sadece dirty bufferların diske aktarılmasına yarar.
- Son kurtarma noktası olarak xloga yazılır.
- Çökme anında son checkpoint noktasından başlanarak xloglar okunur, veriler diske aktarılır sunucu ondan sonra başlar (REDO)

# Checkpoint

- Checkpoint parametreleri:
  - `checkpoint_segments`
  - `checkpoint_timeout`
  - `checkpoint_completion_target`
  - `checkpoint_warning`
- $(2+c_c_t)*c_s+1$  formülü
- $3*checkpoint\_segments+1$

# Ajanda

1. PostgreSQL yönetim temelleri
2. Donanım
3. İşletim sistemi
4. PostgreSQL bileşenleri
  1. Bellek kullanımı
  2. Bgwriter
  3. Checkpoint
  4. İstatistiklerin kullanımı
  5. Autovacuum

# Ajanda

1. PostgreSQL yönetim temelleri
2. Donanım
3. İşletim sistemi
4. PostgreSQL bileşenleri
  1. Bellek kullanımı
  2. Bgwriter
  3. Checkpoint
  4. İstatistiklerin kullanımı
  5. Autovacuum

# İstatistiklerin kullanımı

- İstatistikler PostgreSQL DBA'i için en önemli kontrol noktalarından birisidir.
- Datayı nereye nasıl yerleştireceğinizin ipucunu bu istatistikler verir.
- pg\_stat\* tabloları
- <http://www.gunduz.org/seminer/pg> adresinde Statistics Collector ile ilgili seminer notlarını okuyabilirsiniz.

# İstatistiklerin kullanımı

- Canlı bir örnek: Bir indexi boş bir diske/tablespace içine taşıyarak 3 sn süren bir sorguyu 0.2 saniyeye indirdim.
- Aynı zamanda düzgün I/O için de istatistikleri takip etmek önemlidir.
- Munin gibi araçlar kullanılabilir.



# Ajanda

1. PostgreSQL yönetim temelleri
2. Donanım
3. İşletim sistemi
4. PostgreSQL bileşenleri
  1. Bellek kullanımı
  2. Bgwriter
  3. Checkpoint
  4. İstatistiklerin kullanımı
  5. Autovacuum

# (Auto)VACUUM

- Dead tuple temizliği
- Daha az I/O için önemli
- ANALYZE ile beraber kullanımı
- FSM kavramı (8.4+ itibariyle artık otomatik!)
- Bloat kavramı ve VACUUM ile ilişkisi
- Autovacuum
- VACUUM FULL **kullanmayın**. 9.0'da bu vacuum modeli kaldırıldı.

# Bakım işlemleri

- CLUSTER
- REINDEX
- TRUNCATE

# Diğer önemli parametreler

- random\_page\_cost
- cpu\_tuple\_cost
- cpu\_operator\_cost
- effective\_cache\_size
- default\_statistics\_target

# İpuçları

- Çok sayıda temp table işlemi yapıyorsanız temp\_tablespaces parametresine birden fazla tablespace girebilir ve hatta bu tablespaceleri ramdiske atarak önemli bir başarımla artışı sağlayabilirsiniz.
- stats\_temp\_directory parametresini de benzer şekilde hızlı bir diske ya da ramdiske atmak isteyebilirsiniz.

...ve

# Sorular?

**03.04.2010**  
İstanbul

**Özgür Yazılım ve Linux**  
**Günleri 2010**

markafoni 

# İleri Seviyede PostgreSQL Yönetimi

Devrim GÜNDÜZ

PostgreSQL Geliştiricisi

PostgreSQL DBA @ Markafoni

[devrim@gunduz.org](mailto:devrim@gunduz.org)

[devrim@postgresql.org](mailto:devrim@postgresql.org)[.tr]

<http://twitter.com/devrimgunduz>