

POSTGRESQL

Veritabanı Sunucusu

ve

PostgreSQL ile Kurumsal Çözümler

Devrim Gündüz
devrim@tdmsoft.com
devrim@gunduz.org
devrim@PostgreSQL.org

24.06.2004
İstanbul

PostgreSQL Veritabanı Sunucusu



GİRİŞ

- Bu sunuda aşağıdaki konular anlatılacaktır:

1. PostgreSQL Nedir?
2. PostgreSQL'in gelişimi
3. PostgreSQL'in kullanım alanları
4. PostgreSQL'in bazı teknik özellikleri
 - Point In Time Recovery
 - Tablespaces
 - Nested Transactions
 - SQL standartlarına uyumluluk
 - Kullanıcı tanımlı veri tipleri ve fonksiyonları
 - PL/pgSQL SQL Yordamsal Dili
 - Index çeşitleri

GİRİŞ

5. PostgreSQL'in gelişmiş özellikleri
6. PostgreSQL'de performans artırma yöntemleri
7. PostgreSQL'i kritik ortamlarda kullanan kurumlar
8. PostgreSQL'in kullanılabileceği ortamlar
 - * Jboss
 - * C-JDBC
 - * GIS Uygulamaları (PostGIS)
9. Yedekleme ve veri kurtarma
10. Replikasyon çözümleri
11. PostgreSQL'de kümeleme

PostgreSQL nedir?

- PostgreSQL, veritabanları için ilişkisel modeli kullanan ve SQL standart sorgu dilini destekleyen bir veritabanı yönetim sistemidir.
- PostgreSQL aynı zamanda iyi performans veren, güvenli ve geniş özellikleri olan bir DBMS'tir. Hemen hemen tüm UNIX ya da Unix türevi (Linux, FreeBSD gibi) işletim sistemlerinde çalışır. Ayrıca NT çekirdekli tüm Windows sistemlerde de çalıştırılabilir. Ücretsiz ve açık kaynak kodludur.

Kaynak: <http://www.postgresql.org>

PostgreSQL' in tarihi

- PostgreSQL'in geçmişi 1977'de Kaliforniya' daki Berkeley Üniversitesinde (UCB) yapılan çalışmalara dayanır. UCB'de 1977-1985 yılları arasında Ingres adı verilen relational veritabanı geliştirildi.
- Ingres kodu Relational Technologies/Ingres Corporation tarafından satın alındı ve ilk ticari ilişkisel veritabanlarından biri oldu.

PostgreSQL' in tarihi

- Berkeley'deki ilişkisel veritabanı sunucusu üzerindeki çalışmalar 1986 – 1994 arasında devam etti ve bu veritabanı Postgres adını aldı. Bu kod ise Illustra tarafından satın alındı ve Informix olarak geliştirilmeye başlandı.
- 1994'te SQL özellikleri Postgres'e eklendi ve bu veritabanı Postgres95 adını aldı.

PostgreSQL' in tarihi

- 1996 yılında Postgres tanınmaya başlandı ve kod geliştirmesi için e-posta listesi açılmasından sonra birçok gönüllü Postgres'i geliştirmek için çalışmaya başladı. Bu aşamadan sonra Postgres son kez adını değiştirdi ve adındaki “95” ekinin yerine daha uygun olan SQL konmasına karar verildi. Bunun nedeni Postgres'in artık SQL standartlarını desteklemesiydi. Böylece PostgreSQL doğdu.
- Şu andaki sürümü 7.4.3'tür.

Neden PostgreSQL?

- PostgreSQL şirketiniz ve işiniz için diğer veritabanı sistemlerinden üstün olarak aşağıdaki özellikleri sunar:
 - PostgreSQL ile, kimse sizi lisans anlaşmalarını kırmanız durumunda, bu yazılım için lisans maliyeti olmadığı için dava edemez.
 - Patentli veritabanı satıcılarından daha iyi destek.

Neden PostgreSQL?

- Bu, bazı ek avantajları da beraberinde getirir:
 - Geniş-ölçeklenebilir yayılma ile daha karlı iş modelleri
 - Herhangi bir aşamada lisansınıza itiraz edilme olasılığının olmaması
 - Ek lisanslama maliyetleri olmadan genel araştırmalar ve denemeler yapabilme esnekliği.

Neden PostgreSQL?

- PostgreSQL, güçlü destek imkanlarına ek olarak, çalışanlarınızın katkıda bulunabileceği profesyonellerden ve renkli kişiliklerden oluşan bir takıma sahiptir.
- Personel giderlerinde çok ciddi tasarruf imkanı
- Yazılımımız, diğer önde gelen patentli veritabanlarına göre daha az ayar ve bakım gerektirme gereksinimleri için tasarlanmış ve yaratılmıştır. Bununla beraber, özelliklerinden, kararlılığından ve başarımından hiçbirşey kaybetmemiştir.

Neden PostgreSQL?

- Buna ek olarak, eğitim programlarımız diğer patentli veritabanı üreticilerine göre maliyet etkin ve pratik olarak gösterilmektedir.
- Çoğu patentli veritabanının aksine, firmalar PostgreSQL' in uzun yıllar boyunca yüksek aktiviteli işlerde hiç bir zaman göçmediğini belirtmektedirler. Bir kez bile değil. PostgreSQL sadece çalışır.
- Efsanevi güvenilirlik ve kararlılık

Neden PostgreSQL?

- Kaynak kodu herhangi bir ücret ödenmeden alınabilir. Eğer personeliniz PostgreSQL'i herhangi bir şekilde geliştirmek ya da özelleştirmek isterlerse bunu herhangi bir ek ücret ödemedi minimum çaba ile yapabilirler. Bu, PostgreSQL' in her gün dünyanın her bir tarafından gelen destekle büyümesini sağlamaktadır.
- Çoklu platformda çalışma özelliği

Neden PostgreSQL?

- PostgreSQL hemen tüm Unix türevi (son kararlı sürüm ile birlikte 34) üzerinde çalışabilmektedir.
- Aynı zamanda, doğal Windows uyumluluğu PostgreSQL 7.5 sürümü ile birlikte gelecektir.
- Yüksek hacimli işlemler için tasarlanmıştır.

Neden PostgreSQL?

- Grafik tabanlı veritabanı tasarım ve yönetim araçları
 - Veritabanını yönetmek için
 - pgAdmin
 - pgAccess
 - Veritabanı yaratmak için
 - Tora
- Çok sayıda yüksek kalitede grafik arayüzlü araçları vardır.

Teknik açıdan PostgreSQL

- Bazı teknik özellikler açısından, PostgreSQL şunları sunar:
 - %100 ACID uyumlu
 - ANSI SQL uyumlu
 - Referential Integrity
 - Replikasyon (ticari ve ticari olmayan çözümler) ana veritabanının (master) çok sayıda başka veritabanlarına (slave) çoklanmasını sağlar.

Teknik açıdan PostgreSQL

- ODBC, JDBC, C, C++, PHP, Perl, TCL, ECPG, Python, and Ruby için doğal arabirimler.
- Rules
- Views
- Triggers
- Sequences
- Inheritance
- Outer-Joins

Teknik açıdan PostgreSQL

- Stored Procedures
- Kod geliştiriciler için açık API
- Doğal SSL Desteği
- UNION, UNION ALL ve EXCEPT sorgularına destek
- Doğal Kerberos Yetkilendirmesi
- Fonksiyonel ve Partial Indexler
- Procedural Diller

Teknik açıdan PostgreSQL

- MD5, SHA1, XML ve diğer işlevsel özellikler.
- Diğer SQL-uyumlu sistemlerle paylaşabilmek için taşınabilir SQL yaratan araçlar.
- Özelleştirilmiş, kullanıcı-tanımlı veri tipleri için geliştirilebilir veri tipi sistemi ve hızla gelişen yeni veri tipleri.
- Daha az SQL-uyumlu RDBMS'lerden geçişi kolaylaştıran cross-database uyumluluk fonksiyonları

Teknik açıdan PostgreSQL

- Birden fazla işlemcide rahatlıkla çalışabiliyor.
- ODBC desteği ile MS Office ürünleri ile veri aktarımı yapabilir.
- Ticari bir veritabanı olmadığı için, hızdan ziyade güvenliğe önem veriyor.

Teknik açıdan PostgreSQL

- Point-In-Time-Recovery (PITR)
 - Veri bütünlüğü
 - Çökme anında hızlı veri kurtarma
 - Flashback Query
 - Kullanıcıdan kaynaklanan kaybolan veriyi kurtarmak
 - Incremental Backup

Teknik açıdan PostgreSQL

- Tablespace
 - Verilerin fiziksel yerleşimini kontrol etmek
 - Verileri disklere paylaşmak
 - Daha kolay yönetim
 - Daha kolay yedekleme

Teknik açıdan PostgreSQL

- Nested Transactions
- PL/pgSQL SQL Yordamsal Dili
- SQL standartlarına uyumluluk
- Kullanıcı tanımlı veri tipleri ve fonksiyonları
- Index çeşitleri
 - Hash, B-tree, R-tree, GiST
 - Partial ve Functional Indexler
- Veriyi saklama yöntemleri

Desteklenen Platformlar

- Linux (Kernel 2.0.X ve üzeri)
- AIX 4.3.2 +
- HP – UX 9.0x ve 10.20
- FreeBSD 4.X +
- IRIX 6.5.6f +
- MacOS-X Darwin +
- NetBSD 1.4, 1.4u
- QNX 4.25
- Solaris 2.5.1-2.7
- Sun OS 4.1.14
- WinNT/Cygwin
- BSDI 4.0.1
- BeOS 5.0.3

PostgreSQL' in mimarisi

- PostgreSQL'in gücü, onun mimarisinden gelir. Ticari veritabanı sistemleri ile ortak olarak PostgreSQL sunucu-istemci ortamında kullanılabilir. Bu hem kullanıcılar hem de geliştiriciler açısından oldukça fazla yarar sağlar.
- PostgreSQL kurulumunun kalbi veritabanı sunucu işlemidir (process).
 - Postmaster olarak adlandırılır . Tek bir sunucu üzerinde çalışabilir.

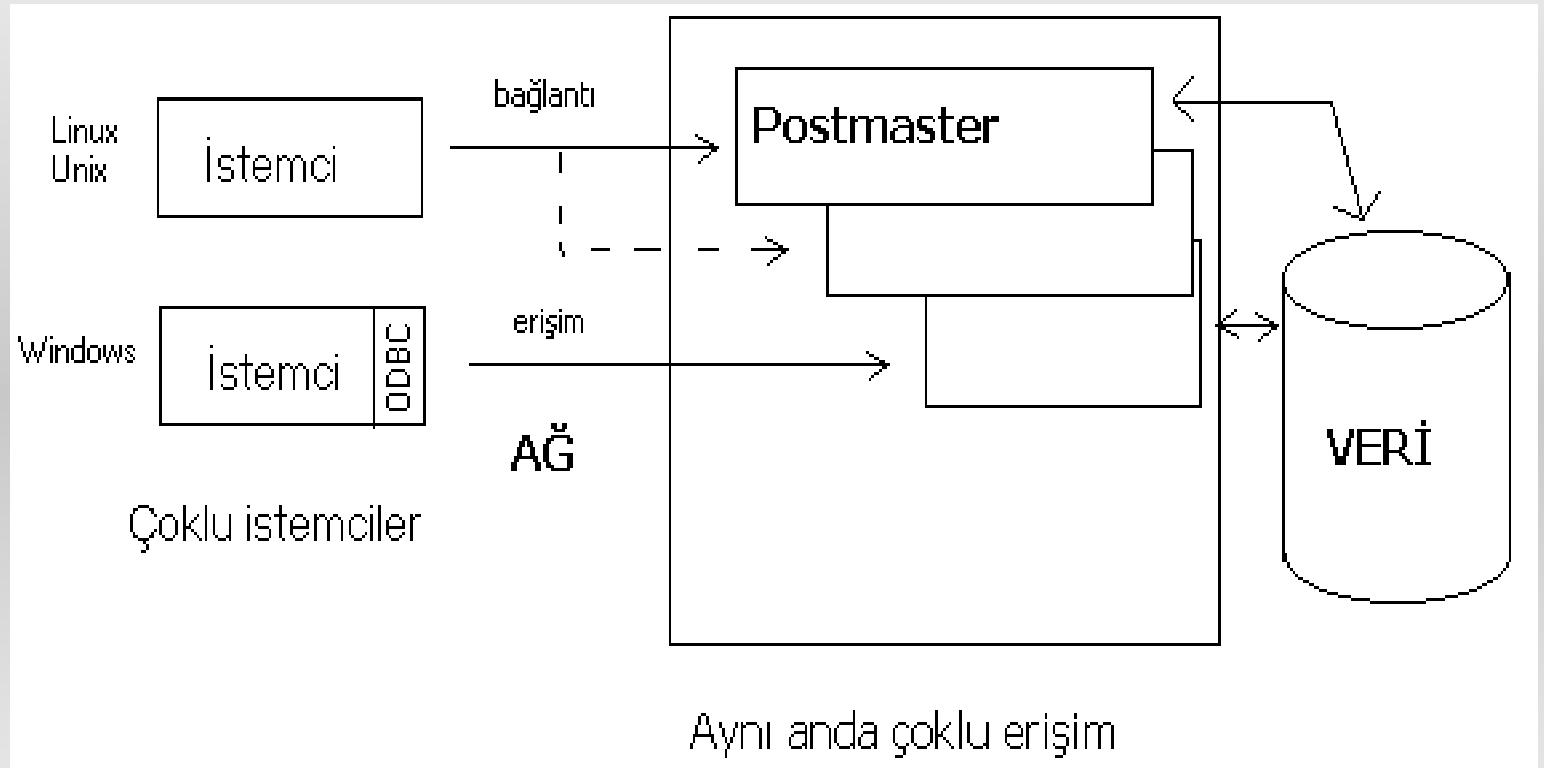
PostgreSQL' in mimarisi

- Veritabanındaki bilgilere erişebilecek programlar sunucu tarafında çalışır. İstemci tarafındaki programlar, sunucu ile aynı makinede olsalar bile veriye direk olarak erişemezler.
- İstemci – sunucu mimarisi, uygulamaların ayrı ayrı makinelerde çalışmasına izin verir. İstemcilerimizi sunucudan ayırmak için bir ağ kurabilir, ve istemci uygulamalarınızı geliştirmek için kullanıcılara uygun bir ortam kullanabilirsiniz.

PostgreSQL' in mimarisi

- İstemci-sunucu mimarisi iş gücünün bölünmesine yardımcı olur. Büyük miktarda veriyi tutabilecek ve erişilebilecek şekilde düzenlenmiş bir sunucu makinesi güvenli bir veri deposu olarak kullanılabilir.
- Gelişmiş grafiksel uygulamalar istemciler için geliştirilebilir.
- Web tabanlı uygulamalar da istemci tarafındaki işlemlerinizi görebilir.

PostgreSQL' in mimarisi



PostgreSQL Sınırlamaları

- Tablolara eklenen veriler büyüyünce, veritabanlarını kontrol etmek zorlaşır ve veritabanlarının başarımı düşer. Burada PostgreSQL'in bir farkı yoktur.
- En fazla büyüklük, disk alanı ve sanal bellekle sınırlıdır. Sınıra yaklaşıldığında, veritabanı yavaşlamaya başlar.
- Sanal belleği bile aşacak bir işlem yapıldığında, PostgreSQL'in başarımı fiziksel açıdan çok kötü olacaktır (ya da bir işlem olmayacaktır!).
- Burada bahsedilmeyen diğer sınırlamalar işletim sistemi ya da ağın veri iletme hızına bağlıdır. Örneğin, ODBC ile yapılan sorguların sürücüyeye bağlı olan sınırları vardır. Hafıza ile ilgili sınırlamalar da vardır (çok büyük bir sorgunun sonucu gibi)

PostgreSQL Sınırlamaları

- Veritabanı için max büyüklük
 - Bir tablo için max büyüklük
 - Bir row için max büyüklük
 - Bir field için max büyüklük
 - Tablo içindeki max row sayısı
 - Tablo içindeki max column sayısı
 - Tablo içindeki max index sayısı:
- Sınırsız
 - 64 TB (Tüm işletim sistemlerinde)
 - Sınırsız
 - 1 GB
 - Sınırsız
 - 1600
 - Sınırsız

Burada bahsedilen sınırlamalar, fiziksel sınırların haricindeki sınırlardır.

24.06.2004
İstanbul

PostgreSQL Veritabanı Sunucusu



PostgreSQL Veri Tipleri

- PostgreSQL, Users' Guide ve psql'deki \dT komutu ile de görülebileceği gibi oldukça fazla veri tipini destekler.
- Sadece SQL standardı değil, kendine özel başka veri tipleri de kullanır.
- Serial, inet, circle, polygon, int, varchar, oid...

PostgreSQL araçları - psql

- Oracle'daki SQL*PLUS gibi PostgreSQL'de psql adında command line aracı vardır. PostgreSQL veritabanları genellikle bu uygulama tarafından yaratılır ve yönetilir.
- PostgreSQL'in desteklediği tüm komutlar burada çalıştırılabilir.
- \h ve \?
- ...

Görsel Araçlar – PgAdminIII

- Linux, Windows, FreeBSD ve Solaris üzerinde çalışır.
- Oldukça geniş bir kullanıcı kitlesi bulunmaktadır.
- Ücretsizdir
- <http://pgadmin.PostgreSQL.org>
- <http://www.pgadmin.org>

Görsel Araçlar – PgAdminIII

The screenshot shows the PgAdmin III interface. On the left, a tree view displays the database structure for (192.168.1.2:5432). The 'public' schema is selected. The right pane shows the 'Propriétés' (Properties) tab for the 'public' schema, listing various attributes like Nom, OID, Propriétaire, ACL, etc. Below this, the 'Statistiques' (Statistics) tab is active, displaying a SQL script for creating and granting permissions to the 'public' schema.

Propriétés	Valeur
Nom	public
OID	2200
Propriétaire	postgres
ACL	{=UC}
Schéma système ?	Oui
Commentaires	Standard public namespace

```
-- Schema: "public"
-- DROP SCHEMA public;

CREATE SCHEMA public
  AUTHORIZATION postgres;
GRANT ALL ON SCHEMA public TO public;
COMMENT ON SCHEMA public IS 'Standard public namespace'
```

Restaurer le précédent environnement tel qu'il était... Réalisé. 4,46 secondes

24.06.2004
İstanbul

PostgreSQL Veritabanı Sunucusu



Görsel Araçlar – phpPgAdmin

- Web arayüzlü olduğu için, çalıştırıldığı ortamdan bağımsızdır.
- postgres kullanıcısı ile veritabanınızı yönetebileceğiniz gibi belirli bir kullanıcı ile belirli bir veritabanını da yönetebilirsiniz.
- Her türlü kayıt ekleme, silme, değiştirme; veritabanı/tablo yaratma vb işlemlerini SQL bilmeden yapabilirsiniz.

Görsel Araçlar – PHPPgAdmin

- Veritabanının içeriğini bir dosyaya boşaltabilir, daha sonra bu içeriği başka bir sunucuda da kullanabilirsiniz.
- Bunların dışında PostgreSQL'in SQL komutlarını çalıştırabilirsiniz.
- Türkçe dil desteği 2.4.2 sürümü ile gelmiştir.
- <http://phppgadmin.sourceforge.net> adresinden ücretsiz olarak indirilebilir.

Diğer görsel araçlar

- Aquafold Data Studio
- EMS PostgreSQL Manager

Vacuum

- Güncellenmiş ve silinmiş satırlardan boşalan disk alanını kurtarmak
- PostgreSQL query planner tarafından kullanılan veri istatistiklerini güncellemek
- Transaction ID sınırlarından kaynaklanan çok eski verilerin kaybolmasının engellenmesi
 - 7.2 sürümü ve sonrasındaki değişiklikler

PostgreSQL'de başarıml (performans)

- VACUUM (pg_autovacuum)
- EXPLAIN (ANALYZE)
- postgresql.conf

Bunlar, iyi bir başarıml için gerekli olabilecek takımdır.

PostgreSQL'de başarıım

- EXPLAIN
- Örnek:
- `tdmsoft=# EXPLAIN ANALYZE SELECT firm_info.full_name, cases.id, userdetails.name,userdetails.surname,cases.case_def FROM cases LEFT OUTER JOIN firm_info ON (firm_info.id=cases.firm_id) LEFT OUTER JOIN userdetails ON (user_id=userdetails.id) WHERE cases.case_closed='f' AND cases.active='t' AND cases.id='2';`
-

PostgreSQL'de başarıml

- Nested Loop Left Join (cost=0.00..3.76 rows=1 width=654) (actual time=0.054..0.054 rows=0 loops=1)
 - Join Filter: ("outer".user_id = "inner".id)
- -> Nested Loop Left Join (cost=0.00..2.56 rows=1 width=592) (actual time=0.052..0.052 rows=0 loops=1)
 - Join Filter: ("inner".id = "outer".firm_id)
- -> Seq Scan on cases (cost=0.00..1.40 rows=1 width=528) (actual time=0.050..0.050 rows=0 loops=1)
 - Filter: ((case_closed = false) AND (active = true) AND (id = 2))
- -> Seq Scan on firm_info (cost=0.00..1.07 rows=7 width=72) (never executed)
- -> Seq Scan on userdetails (cost=0.00..1.09 rows=9 width=70) (never executed)
- Total runtime: 0.198 ms

24.06.2004
İstanbul

PostgreSQL Veritabanı Sunucusu



PostgreSQL'de başarıml

- EXPLAIN çıktısını yorumlama
 - start-up-cost: Sonucun ekrana verilebilmesi için geçecek olan süre (örnek: sort işlemi varsa bunun için geçecek süre)
 - total cost: Sorgunun çalışacağı toplam süre. Eğer gereksiz sayıda satır gelecekte, LIMIT ile sorguyu sınırlamak süreyi düşürebilir.)
 -

PostgreSQL'de başarıml

- number of rows output by this plan node: Bu sorgunun planında dönecek toplam satır sayısı
- Bu plandandaki satırların ortalama “genişliğı” (width) (byte cinsinden)

PostgreSQL'de başarıml

- Örnek:
- tdmsoft=# EXPLAIN SELECT tarih FROM info WHERE tarih > '2004-01-01';
- QUERY PLAN
- -----
- Seq Scan on info
(cost=1000000000.00..100002410.40
rows=43691 width=4)
- Filter: (tarih > '2004-01-01'::date)

PostgreSQL'de başarımlar

- tdmsoft=# SELECT * from pg_class WHERE relname='info';
- relpages | 6169
- reltuples | 1.04858e+06
- relhasindex | t

PostgreSQL'de başarıml

- tdmsoft=# CREATE INDEX info_tarih_idx
ON info USING btree (tarih) WHERE tarih >
'2004-01-01';
CREATE INDEX
- test=# EXPLAIN SELECT tarih FROM info
WHERE tarih > '2004-01-01';
Index Scan using info_tarih_idx on info
(cost=0.00..1175320.50 rows=1000 width=4)
Index Cond: (tarih > '2004-01-01'::date)

PostgreSQL'de başarıml

- SET komutu
 - enable_seqscan
 - enable_nestloop
 - enable_indexscan
 - enable_sort
- EXPLAIN ile EXPLAIN ANALYZE arasındaki fark
 - BEGIN/ROLLBACK

PostgreSQL'de başarıml

- max_connections = 32
 - Baęlantı başına 14 K
 - Gerektięi kadar az tutulmalıdır.
 - Not: Aynı anda 2^{32} baęlantı imkanı

PostgreSQL'de başarımlar

- `max_fsm_relations = 1000`
 - # min 10, fsm is free space map, ~40 bytes
- FSM kavramı
- Yoğun bir veritabanında, bu değerin 1000'den çok daha yüksekte tutulması gerekir.
- `max_fsm_relations` için ise şöyle bir tanım vardır: "Sets the maximum number of relations (tables) for which free space will be tracked in the shared free-space map."

PostgreSQL'de başarımlar

- `#max_fsm_pages = 10000`
min 1000, fsm is free space map, ~6 bytes

VACUUM işlemlerinde yardımcı olan bir parametredir. Daha doğrusu, VACUUM FULL'a gereksinim duyma olasılığını azaltır. Aslında bunun değerini hesaplamak için önce bir VACUUM ANALYZE yapmak, ardından da çıkan sonuçlardaki sayfa sayısını toplamak gerekir.

PostgreSQL'de başarıml

- `#wal_buffers = 8` # min 4, typically 8KB each
- WAL (Write-Ahead-Logging), transaction logging olarak da adlandırılabilir.
 - PostgreSQL kitaplarındaki tanım

PostgreSQL'de başarıml

- `#shared_buffers = 64`
 - `# min max_connections*2` or 16, 8KB each
- 1 GB'lik ramın 60-75 MB'ı ayrılabılır.
- `ipcs` (interprocess communication facilities status)

PostgreSQL'de başarımlar

- `#vacuum_mem = 16384`
 - # min 1024, size in KB
- `vacuum_mem`'in belleğin %20'sinden büyük olmaması önerilir.
 - Belirtilen miktarda bellek “ayrılır”.

PostgreSQL'de başarıml

- #fsync = true
 - Bu parametre, verinin commit edilir edilmez diske yazılıp yazılmayacağını belirler (WAL...). Eğer donanımınıza, güç kaynağınıza vs güveniyorsanız, bunu no yapıp veri yazma hızını bol miktarda arttırabilirsiniz. Ancak unutmayın ki beklenmeyen bir durumda (veritabanının aniden kapatılması (normal yollarla değil...) gibi) sizi en son aldığınız yedeğe döndürecektir.

PostgreSQL'de başarıml

- `#effective_cache_size = 20000`
 - # typically 8KB each
- Bu parametre, PostgreSQL'in sunucunuzdaki en uygun RAM kullanımına yardımcı olur. Bu parametre, PostgreSQL'e işletim sisteminin veri cache'inin büyüklüğünü bildirir. Böylece, PostgreSQL veriye göre değişik sorgu işleme planları çıkartabilecektir.

PostgreSQL'de başarıml

- effective_cache_size için 500 MB yer ayıralım. Diyelim bir sorgu 400 MB'lık yere gereksinim duydu. PostgreSQL tüm verinin rama aktarılabilirliğini görür ve optimizasyon açısından çok iyi bir plan çıkartır (bol index kullanımı ve join işlemleri, vs...) Eğer effective_cache_size 300 MB olmuş olsaydı, sıralı bir aramanın (sequential scan) daha iyi olacağını görecekti, bu da performansın göreceli olarak düşmesine neden olacaktı.

PostgreSQL'de başarıml

- Ancak bu parametreleri ayarlarken, sunucuda çalışan diğler uygulamalar için de yer ayırmamız gerektiğini unutmayın (kernel, apache, belki php, vs...). Bu parametrenin değeri, PostgreSQL'e tüm zamanlarda ayrılalabilecek en yüksek RAM miktarı olabilir.

Ek Bilgiler

- Veritabanında açılan her bağlantıyla ilgili bilgiler shared memory de tutulur. Dolayısıyla shared memory'nin büyük olması bağlantıları daha iyi yönetilmesi demektir.
- **Ara bellek (buffer) :** Tablolardan okunan satırlar önce ara belleğe gelir . sonra kullanıcıya aktarılır; tam tersi de olur. Eğer boş RAM iniz varsa ara bellek ve shared memory ye dağıtabilirsiniz. Tablolar çok fazla hareket görüyorsa ara bellek değeri yüksek tutulmalıdır. Disk I/O oranı düşer; bu da oldukça fazla hız artışı sağlar. Genel olarak başarımda dikkat edilmesi gereken 3 unsur bulunmaktadır:
 - CPU , hafıza (RAM) ve Hard Disk.

Kimler kullanıyor?

- Türkiye'deki belli başlı üniversiteler
- Malatya İnönü Üniversitesi Turgut Özal Tıp Merkezi
- Maden Tetkik ve Arama Ens., Şeker Fabrikaları
- www.begendim.com
- .info ve .org alan adları (www.afilias.com)
- <http://www.sciencetunnel.com>
- Cisco
- BASF
- Red Hat
- Chrysler
- 3Com
- Vanten Inc.
- ...

PostgreSQL ve JBoss

- JBoss

- JBossMX - JMX management
- JBossSX - güvenlik platformu
- JBossJTA - transaction yöneticisi
- JBossJCA - Java Connector Architecture
- JBoss.NET - .NET ile entegre çalışma
- JBossMQ - JSM messaging
- Tomcat - web server ve JSP aracı
- JBossAOP - Aspect Oriented Programming
- JBossCache - gelişmiş bir cache yazılımı

PostgreSQL ve JBoss

- 2003 yılında 2 milyon download
- SMP desteği
- JavaWorld Editor's Choice ödülü
- Fortune 1000 şirketleri, ABD ve birçok Avrupa hükümetleri tarafından kullanılıyor.
- Öncül JMX desteği

PostgreSQL ve JBoss

- J2EE 1.4 standartların desteği
- JMX gibi standartların diğer uygulama sunucularından önce desteklemesi
- JBoss, standartlarından ödün vermiyor
- JBoss J2EE tüm testlerinden başarı ile geçmiştir

PostgreSQL ve JBoss

- Kümeleme desteği J2EE uygulama sunucularının ayrılmaz bir özelliğidir.
- JBoss, başlangıçtan beri kümelemeyi destekliyor.
- JBoss, denenmiş ve kendini kanıtlamış bir kümeleme desteği sağlıyor.

PostgreSQL ve C-JDBC

- Bir “middleware”
- Mevcut kodların değiştirilmesine gerek kalmaz.
- Intercommunication Overhead
- Başarım ölçeklemesi ve “fault tolerance”
- RAIDb kavramı

PostgreSQL ve GIS Uygulamaları

- PostGIS : <http://www.postgis.org>
 - PostgreSQL'e GIS desteği
 - Oracle'in Spatial eklentileri
 - OpenGIS standartlarına uygunluk
 - “OpenGIS Simple Features Specification For SQL”
 - Bilimsel alanda öncü çalışma
 - Avrupa Çevre Ajansı

Yedekleme ve Veri Kurtarma

- pg_dump
- pg_dumpall
- pg_restore
- PITR
 - Incremental Backup
 - Roll Backward/Forward
- Tablespaces
 - İşletim sistemi yedeği

Replikasyon Çözümleri

- eRServer
- Slony-I
- pgreplication
- C-JDBC
- Octobus

PostgreSQL ve Kümeleme (Clustering)

- Clusgres
- C-JDBC

24.06.2004
İstanbul

PostgreSQL Veritabanı Sunucusu



Kaynaklar

- <http://www.PostgreSQL.org>
- <http://advocacy.PostgreSQL.org>
- <http://techdocs.PostgreSQL.org>
- <http://www.PostgreSQL.org/idocs>
- <http://developer.PostgreSQL.org>

Kaynaklar

- <http://www.pgsql.com/>
- <http://foundries.sourceforge.net/databases>
- <http://gborg.PostgreSQL.org>
- www.wrox.com - Beginning Databases with PostgreSQL, 2001

POSTGRESQL

Veritabanı Sunucusu

ve

PostgreSQL ile Kurumsal

Çözümler

Devrim Gündüz
devrim@tdmsoft.com
devrim@gunduz.org
devrim@PostgreSQL.org

24.06.2004
İstanbul

PostgreSQL Veritabanı Sunucusu

