

Memcached & Php

Nedir ?

Memcached "Danga Interactive" (<http://www.danga.com/>) tarafından LiveJournal için geliştirilen dağıtık , nesne tabanlı tampon bellek sistemidir (distributed memory object caching system , biliyorum kötü bir çeviri).

Proje Live Journal için geliştirilse de , Facebook en büyük destekçisi haline gelmiştir : yazılım üzerinde yaptıkları değişikliklerle işlemci kullanım oranını %20 azaltırken, bellek kullanımında %30'luk bir iyileştirme yapmışlardır (<http://developers.facebook.com/opensource.php>) ; aynı zamanda mamcached kurulu en büyük yapı facebook'da bulunmaktadır. (<http://lists.danga.com/pipermail/memcached/2007-May/004098.html> , 200 X 16GB 4 Çekirdekli Amd64 Sunucu)

Nerede kullanılır ?

Sık talep edilen bilgileri sunmada ve veritabanını belirli ölçüde rahat bırakmada kullanılır. Hiç şüphe yokki belleğe erişim, diske erişimden daha hızlıdır. Veritabanlarının tamponlama (caching) özelliği olsada duruma göre memcached dağıtık yapısı nedeni ile tercih edilebilir.

Web sitelerinde kullanım yerleri olarak ;

- + Giriş işlemlerinde kullanıcı oturumlarının saklanması,
- + Siteiçi aramalarda , çok sık yapılan aramalar için kelimeler ve sonuçlarının hafızada saklanması,
- + Günlük yada haber sitelerinde ilk sayfada çıkan haberlerin hafızda saklanması,

örnek verilebilir.

Nasıl ?

Kurulum kullandığınız dağıtıma göre değişmekle beraber, Ubuntu üzerinde

```
#apt-get install memcached php5-memcache
```

komutu ile kurulum işlemini gerçekleştirebilirsiniz.

!!! Ubuntu & Php-5 için not : Apt-get kurarken php.ini'de ufak bir hata yapıyor , /etc/php5/apache2/php.ini 'de -e extension=memcache.so olan satırı bulup, fazlalık olan "-e "yi silmeniz gerekiyorr son hali extension=memcache.so olmalı

Apt-get yazılımları yükledikten sonra /etc/memcached.conf dosyasını açıp "-m 64" değerini bulun, varsayılan olarak memcached 64Mb ram kullanmaktadır, mevcut ram miktarınıza göre uygun bir değer girin. Dikkat: bu alana sahip olduğunuz tüm belleği girmek gibi bir hata yapmayın.

Kurulum işlemini apache'yi ve memcached'ı yeniden başlatarak tamamlayın.

```

<?
// Yeni bir memcache nesnesi oluřtur
$memcache = New Memcache();

// Sunucu üzerinde kurulu olan memcached sunucusuna baėlan
$memcache->connect('localhost', 11211) or die("Memcache:> Baėlantı kurulamadı...");

// Sunucu üzerinden "kayit" nesnesini al
$skayit = $memcache->get('kayit');

// Eėer kayit nesnesi daha önceden atanmamıřsa deėer ata
if( !$skayit )
{
    $skayit['sayac'] = 0;
    $skayit['zaman'] = time();

    // $skayit deėiřkenini, sunucu üzerinde "kayit" nesnesi olarak
    // 10sn için hafızaya ata.
    $memcache->set('kayit', $skayit, false, 10);
}
else
{
    // $skayit deėiřkenindeki sayacı arttır.
    $skayit['sayac']++;

    if( $skayit['sayac']++ == 25 )
    {
        $memcache->delete('kayit');
        echo "Memcache'den , 'kayit' nesnesi kaldırıldı..";
        return;
    }

    // sunucu üzerindeki kayit nesnesini güncelle
    // false: deėiřkenin ziplenerek saklanması istebiyorsa TRUE olmalıdır;
    // 10 : deėiřken güncellenirken, ne kadar süre daha hafızada kalacaėıda
    // ayarlanabilir
    $memcache->replace('kayit',$skayit, false, 10);
}
printf("Zaman : %s , Sayac: %s ", date("h:i:s d.m.Y", $skayit['zaman']), $skayit['sayac'] );
?>

```

Örnek-1

Örnek-1 'de

3..6. Satırlar.....:

ilk olarak bir memcache nesnesi oluřturulmakta ardından, memcache kurulu sunucuya (localhost) , memcached'ın varsayılan portu olan 11211. port'dan baėlanması saėlanmaktadır.

9. Satır.....:

Sunucu'dan 'kayit' nesnesi alınmakta ve \$skayit deėiřkenine atanmakta.

12..20. Satırlar :

Sunucu'da kayit nesnesi bulunamazsa , yeni bir dizi oluřturulup bu dizi'de 'sayac' elemanına 0, 'zaman' elemanına o anki zaman atanmakta ve \$memcache nesnesinin "set" fonksiyonu ile \$skayit dizisi , sunucuda 'kayit' nesnesi olarak oluřturulmakta. Set fonksiyonundaki 10 ifadesi , saniye cinsinden kayıdın sunucuda ne kadar süre ile hafızada tutulacaėını belirtiyor, eėer bir deėer vermezseniz bu varsayılan olarak 30 gün'dür ve 30 günün sonunda bu nesne silinecektir.

23..32. Satırlar:

Sunucu üzerinde bulunan kayit nesnesini 9. satır'da \$skayit deėiřkenine atamıřtık, bu deėiřken bir dizi ve bu dizi üzerindeki 'sayac' elemanın deėerini bir arttırıyoruz. Eėer bu deėer 25'e eřit ise memcached sunucusu üzerindeki kayit

nesnesi, \$memcached objesinin "delete" fonksiyonu ile siliyoruz. Neden böyle birşey yapma ihtiyacı içerisindeyiz ? Açıkcası bu gayet sıkıcı bir örnek ve ortama eğlence katmak lazım , bilgisayarımızı candan aşağıya atamıyorsak bizde değişkenlere saldırırız :)

37. Satır.....:

Sunucuda nesnemiz bulundu , \$kayit dizisinde sayac değerini arttırdık fakat bu değer 25'in altında o zaman yapılacak tek birşey kalıyor : yeni değeri güncellemek. \$memcache nesnesinin replace fonksiyonu tam bu işe yarıyor ve set ile benzer parametreleri alıyor. 10 değeri bu arada nesnenin hafızada kalma süresini tekrar 10 sn yapıyor.

Sanırım işleyiş mantığını anlamaya başladınız. Kullanıcı giriş bilgilerini veritabanı ile olan iletişimi en aza indirmek için memcache ile hafızada tutmak isterseniz aşağı yukarı şunun gibi bir fonksiyon kullanmanız lazım :

```
function Giriş ($kullanıcı, $parola )
{
    $memcache = New Memcache();
    $memcache->connect('localhost', 11211) or die("Memcache:> Bağlantı kurulamadı...");

    $kayit = $memcache->get( 'giris:' . $kullanıcıadi );
    if( !$kayit )
    {
        $sorgu = mysql_query("Select * From Kullanıcılar Where kullanıcı='$kullanıcı' and parola='$parola' Limit 0,1");
        if( mysql_num_rows($sorgu) < 1 )
            return FALSE;

        $kayit = mysql_fetch_array($sorgu);
        $memcache->set('giris:'.$kullanıcıadi , $kayit, false, 10);
    }
    else
    {
        if( ($kayit['kullanıcı'] != $kullanıcı) && ($kayit['parola'] != $parola ) )
            return FALSE;
    }
    return TRUE;
}
```

Örnek - 2

Kodun yaptığı kabaca , memcached sunucusunda kullanıcı bilgilerinin olup olmadığına bakmak, yoksa veritabanına bağlanıp kullanıcı bilgilerini almak ve kayıt dönerse (= kullanıcı bilgileri doğru ise) bunu memcached'da bir nesne olarak atamak. Eğer memcached üzerinde kullanıcı bilgilerini bulursa onun üzerinden kontrolleri yapmak.

Php'nin oturum bilgilerini memcache üzerinde tutmasını sağlayabilirsiniz. Yanlız burada debian/ubuntu kullanıyorsanız ufak bir sorun var : oturum bilgilerini memcached'da tutmanız için php-memcached eklentisinin versiyon numarasının 1.80 yada daha büyük olması lazım ve debian/ubuntu ile eski versiyonlar geliyor. Uzun uzun anlatmak ve hazırda olan bilgileri tekrarlamaktansa , program nasıl derlenir bildiğinizi varsayıyorum ve gerekli bilgileri almanız için sizi http://www.lullabot.com/articles/how_install_memcache_debian_etch adresine yönlendiriyorum.

Oturum bilgilerinin memcached üzerinde tutulmasını iki şekilde ayarlayabilirsiniz :

1- php.ini 'de aşağıdaki gibi satırlar ekleyip/değiştirerek

```
session.save_handler = "memcache"
session.save_path = "tcp://localhost:11211?persistent=1&weight=1&timeout=1&retry_interval=15"
```

2- .php dosyanızda , session_register'den önce şöyle bir şey ekleyerek :

```
ini_set('session.save_handler', 'memcache');
ini_set('session.save_path' , 'tcp://localhost:11211?persistent=1&weight=1&timeout=1&retry_interval=15');
```

Dökümanı bitirirken bahsetmediğim bir fonksiyon hakkında ufak bir bilgi vermem gerekli: Memcached'in dağıtık bir sistem olduğunu söylemiştik, php dosyamızın diğer memcached sunucuları ile bağlantı kurması için "addServer" fonksiyonu kullanmamız lazım. 2x1Gb ram ve 1x2Gb ram'i olan 3 makineli bir ağ için örnek :

```
$memcache = New Memcache();
$memcache->connect('localhost', 11211) or die("Memcache:> Bağlantı kurulamadı...");
$memcache->addServer( 'memc1.local.net' , 11211, TRUE, 10);
$memcache->addServer( 'memc2.local.net' , 11211, TRUE, 10);
$memcache->addServer( 'memc3.local.net' , 11211, TRUE, 20);
```

memcX.local.net : memcached kurulu olan diğer sunucu
11211 : memcached portu
TRUE : kalıcı (= persistent) bağlantı yapılacak
10/20 : işte eğlenceli kısım ; yük dağılımı.
Büyük rakam = fazla ram = Daha fazla başvurulacak makine.

Sonuç & Notlar

Memcached , yoğun trafik alan sunucularınızı biraz rahatlatmak için kullanabileceğiniz güzel bir oyuncak bununla beraber ağır çalışma ortamlarında test edildiği için deneyip kendi durumunuzda işe yaradığını görürseniz kullanmamanız için hiçbir sebep yok.

Memcached & Apache yeterli ram olduğu sürece aynı makinede çalışabilir, genel olarak memcached belleğe apache işlemciye yüklenir

Dökümanda memcached hakkında bildiklerimi işinize yaracak şekilde anlatmaya çalıştım, hatalar, değişik platformlarda değişik sonuçlar elde edilebilir, sorumluluk kabul etmiyorum. Yazıyı beğenmezseniz mail vasıtası ile kin kusmak yerine oturup daha iyisini yazın, böylesi herkes için daha faydalı olur.

Dökümanda yapmış olabileceğim imla hataları varsa şimdiden özür dilerim. Metnin bir kısmını yada tamamını kendi yazılarınızda kullanacaksanız sağda solda bir yerde siteme link verip, yazınızın tam metnini bana mail atabilirsiniz sevinirim.

Kaynaklar & İlgili Dökümanlar :

- <http://www.danga.com/memcached/>
- <http://tr2.php.net/manual/en/ref.memcache.php>
- http://www.lullabot.com/articles/how_install_memcache_debian_etch
- <http://www.rooftopsolutions.nl/article/107>

[İrtibat Bilgileri]

Web
<http://www.devmach.com>

E-Mail & GoogleTalk
root@devmach.com

16 Eylül 2007

Vrs. 1.1

vrs 1.2 - Web sitesinde yayınlanmak üzere irtibat bilgileri değiştirildi.