Web Tabanlı Uygulamalarda Performansı Etkileyen Unsurlar

Önder Şahinaslan, Emin Borandağ, Şemseddin Aksoy

Bilişim Bölüm Başkanlığı Maltepe Üniversitesi, İstanbul

onder@maltepe.edu.tr, eminb@maltepe.edu.tr, saksoy*@maltepe.edu.tr*

Özet: Web tabanlı yazılım uygulama sistemleri, teknolojik gelişmelerin yaygınlaşması ile birçok uygulamanın temel ve değişmez bir parçası haline gelmiştir. Bu durum her alanda olumlu sonuçlar getirmesi ile birlikte bazı teknolojik problemlerin ortaya çıkmasına neden olmuştur. Bu sorunların en önemlisi sistemlerde ortaya çıkan performans sorunlarıdır.

Bu çalışmada, web tabanlı sistemlerdeki performans sorunun hangi nedenlerle ortaya çıktığı irdelenerek, örnek bir sistem üzerinde web tabanlı uygulamanın performansın arttırılması ile ilgili gerçekleştirilen çalışma anlatılmıştır. Çalışma sonucunda; sistemin performansının arttırılması ile ilgili gerçekleştirilen uygulama süreçlerin ne şekilde gerçekleştirildiği anlatılmış ve elde edilen sonuçlara yer verilmiştir.

Anahtar Sözcükler: Yazılımda Performans, Yazılımda Performans Ölçümü, Normalizasyon, Pooling, Storage Procedure

Abstract: Web-based software applications system became basic and stable part of many applications with the spread of technological developments. This fact caused some negative results in terms of technology along with many positive results in many fields. One of the most important is performance problems in the systems.

In this study, web-based systems performance problem was examined. It is explained that how is web-based applications performance increased on a demo system. As a result of this study, it is explained that how the process is applied in order to improve the performance of the system and results are given.

Keywords: Software performance, software performance measurement, normalization, Pooling, Stored Procedure

1. Giriş

Internet teknolojilerinin gelişmesi ile birlikte önemini arttıran web tabanlı uygulamalar, donanım ve yazılımın sağladığı imkan ve kabiliyetlerin artışıyla pek çok şirketin kendi iç süreçlerini yönetmesinde kullanılmaktadır[1]. Web tabanlı teknolojilerden önce kullanılan sistemlerle karşılaştırıldığında birçok özelliği kendi içerisinde barındırmaktadır. Web tabanlı uygulama sistemleri, sağladığı bu özellik ve avantajlar; platform bağımsız oluşu, her yerden sisteme erişilebilir olması, sistem güncellemelerinin tek bir noktadan yapılabilir olması, standart web tarayıcılar ile kullanım, yazılım indirme, kurma veya yükleme gibi kısıtlarının olmayışıdır. Sağladığı bu avantajlar ile kurum ve kuruluşlar için tercih edilme nedenidir. Ancak kurulan bu sistemlerin performansı etkileyen bazı sorunlar meydana gelebilmektedir[1]. Bu nedenle geliştirilen sistemin etkin bir şekilde kullanılması için performansı etkileyen unsurlara dikkat edilerek geliştirilmesi gerekmektedir. Aksi durumlarda büyük miktarlarda kaynak ve zaman ayrılarak geliştirilen sistemler verimli olarak kullanılmadığı ya da "Garba-

ge Software" Çöp yazılım olarak adlandırıldığı görülmektedir[2].

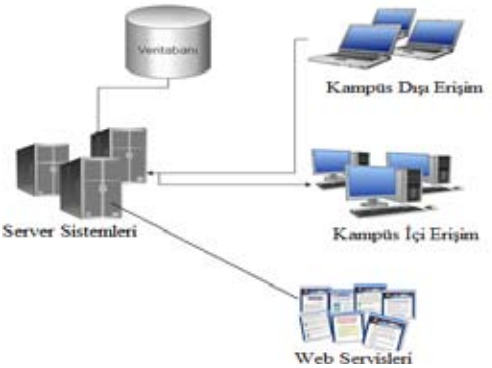
Çalışmanın 2. bölümünde kısaca Web tabanlı yazılımların çalışma mantığı anlatılacak, 3. bölümünde Yazılım performansına etki eden unsurlardan bahsedilecek, 4. Web tabanlı yazılımların performans ölçümünün hangi yazılımlarla yapılabileceği anlatılacaktır, 5.Bölümde performansa etki eden unsurlara dikkat edilerek geliştirilen sistemden bahsedilecektir, son bölümde ise elde edilen sonuçlar anlatılmaktadır.

2. Web Tabanlı Yazılımların Çalışma Mantığı

Web tabanlı sistemler, öncesinde kapalı alanlarda kısıtlı paylaşımlarla yapılan yazılım uygulamalarında karşılaşılan bazı temel sorunları aşmak için geliştirilmiştir. Bu sorunların en önemlileri hız ve konum bağımsız olarak sistemlerin çalışamamasıdır. Bu sorunları aşmak için geliştirilen web tabanlı yazılım geliştirme mimarisi ve yazılım geliştirme platformları ile mevcut sorunları aşılmaya çalışılmıştır. Aşağıda Web tabanlı sistemlerin en temel bazı avantajlar şunlardır[3].

- Web tabanlı yazılımlar, klasik uygulama yazılımlarına nazaran her istemciye ayrı ayrı yüklenmesine gerek yoktur.
- Güncelleme gerektiğinde her istemciye yazılımın yeniden yüklenmesi gerekmemektedir.
- Tarayıcılar ile veri tabanlarından alınan bilgiler istenilen şekilde kullanıcıya yansıtılmaktadır.
- Web tabanlı uygulamalar platform bağımsızdır [3].

Aşağıda Web tabanlı sistemlerde kullanılabilen Client-Server mimarisi gösterilmektedir.



Şekil 1. Net Web Mimarisi

3. Yazılımda Performansa Etki Eden Unsurlar

Kullanıcıların ihtiyaçları doğrultusunda, isterler belirlenerek yazılım teknikleri ile proje oluşturulur. Ancak yapılan bu türlü web uygulamalarında performans öncelikli bir sorun olmaktadır. Sistem trafiğindeki artışlar, kullanıcı seviyelerindeki performans problemleri, işlem süresi, fonksiyonalite ve güvenlik açısından değerlendirilmelidir.

Yazılımda performans; belirlenmiş şartlar altında, tanımlanan görevi, gösterilen süre içerisinde eksiksiz bitirilmesine bağlıdır. Yazılım mühendisliği teknolojisi ve yöntemleri kullanılarak geliştirilen web tabanlı sistemlerde performans, sisteme bağlı alt sistemlerden meydana gelmektedir. Bu sistem; veri tabanları, geliştirilen yazılım, web sunucuları, internet tarayıcılar ve donanımsal yapılardan oluşmaktadır. Aşağıda performansı etkileyen unsurdan maddeler halinde söz edilecektir.[4]

3.1 Veritabanı Performansı

Veriye birden fazla kullanıcının aynı anda güvenilir ve sağlam bir şekilde erişmesi sağlanmalıdır. Web tabanlı sistemlerde sürekli olarak bu verilere erişilip, veri alımları, güncellemeleri, bilgi ekleme işlemleri yapılmaktadır. Sürekli olarak veri alışverişi gerçekleştiğinden sistemin performansını etkileyen en önemli

kriter veri tabanı performansıdır. Veri tabanı sistem performansı, aşağıda 3 temel aşamada ele alınmıştır.

Veri tabanı Tasarımı: Veri tabanı performansını arttırmak için en etkili yöntemdir. Veri tabanı tasarımın iki temel aşaması vardır. Birincisi mantıksal veri tabanı şeması ikincisi ise fiziksel özelliklerdir. Mantıksal veri tabanı şeması tablolardan ve değişkenlerden oluşmaktadır. Fiziksel özelikleri ise; değişkenler, disk sistemi, objeler, objelerin bulundukları yerler ve indeks'lerdir. Veri tabanı tasarımının düzgün bir şekilde yapılması yapılacak yazılımda ihtiyaçların erken belirlenmesi açısından da önemlidir[5].

Q**uery Tuning**: Yazılım performansını ile ilgili bir diğer unsurda sorguların belirlenmesidir. Bu aşama sorguların nasıl geliştirileceği ve geliştirilen sorguların kaliteleri sistem performansı açısından önemlidir. Mümkün olduğu kadar Cursor özelliği kullanılmamalıdır.

Uygulama Dizaynı: Bu kısımda önemli olan mümkün olduğu kadar SQL sorgularının kısaltılmasıdır. Uygulama içerisinde küçük sonuç kümeleri kullanılmalıdır. Bu sayede I/O işlemleri azaltılır ve sistem performansı arttırılır. Yapılan sorgularda kullanıcı tarafından gönderilen iptal işlemlerinin anlık olarak veritabanına yansıtılması sistem performansı arttırır. Bazı sorguların sonsuza kadar çalışması engellemek için API ile kontroller yapılmalıdır. Transactions mümkün olan en kısa sürede gerçekleşmelidir. Stored procedureler kullanılmalıdır.

3.2 Network bağlantı hızları

Sunucu sistemlerine bağlantı erişim hızı, sistem performansını etkilemektedir. Sistem bağlantı hızı fiyat ve performans kriterlerine bakılarak belirlenmelidir.

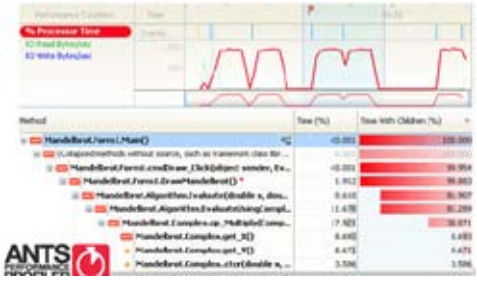
3.3 Internet Tarayıcılar Performansı

Kullanılan internet tarayıcıları bir web uygulamasına bağlanırken farklı component ve algoritmalar kullandıkları için ekrana derleme süreleri farklı olabilir. Bu bekleme süresi de sistem performansını etkilemektedir

3. 4 Yazılım Performans Ölçümü Araçları

Yazılım performans ölçümü, gerçekleştirilen yazılımın ne derece etkin olarak çalıştığının gösterilmesi açısından önemlidir. Yazılım performans ölçümü için çeşitli araçlar geliştirilmiştir. Profiler.net, Ants Performance, Pro Filter ve Firebug gibi yazılım performans ölçümü araçları kullanılarak yazılım performansı ile ilgili bilgiler elde edilebilir.

- **Ants Performance:** Web tabanlı yazılımların hızlı bir şekilde çalışma sürelerini ve performanslarını ölçmek için kullanılmaktadır. Geliştirilen yazılımlarda ortaya çıkabilen ADO.NET ile ilgili tıkanık ve gecikmelerin tespiti için kullanılabilir. Yazılım içerisinde bulunan ve performansı etkileyen prosedürler ile ilgili sorunların giderilmesinde kullanılmaktadır. Bu yazılımın kullanımı kolay ve pratiktir[6].



Şekil 2. ANTS Performance

- **FireBug:** Web sayfaları içerisinde CSS, HTML ve JavaScript kodlarını editlemek, izlemek ve yönetmek için kullanılır. Bu yazılım örnek projemizde performansın ölçümü için kullanılmıştır. Javascript panel, net paneli ve Konsol panellerinin aktif ve pasif olmasını ayarlanabilir. Konsol panelleri yapılan seçime göre aktif ve pasif hale getirilebilir.



Şekil 3. FireBug Kullanımı

Şekil 3'de, Mozilla tarayıcısından FireBug sekmesi görülmektedir. Bu eklenti, kod yazarları için karşılaşılan sorunların çabuk ve hızlı çözmek için çok büyük yarar sağlamaktadır. Hatalara tarayıcı üzerinden doğrudan müdahale etme imkanı vermektedir.

4. Gerçekleştirilen Çalışma

Bir web uygulamasının açılma süresinin %80 lik kısmı görüntü, stil, scriptler, flash, vb gibi componentlerin yüklenmesi için harcanır. Bu componentlerin azaltılması için en önemli yöntem, uygulama dizaynının sade olmasıdır. Fakat aynı zamanda zengin bir içeriğe ve hızlı bir yanıt zamanına sahip olmalıdır. Bu bağlamda performansın arttırılmasına yönelik bazı teknikler geliştirilmiştir. Örnek olarak bütün script'leri ve css dosyalarını tek bir dosya da birleştirmektir. Gerçekleştirilen örnek uygulamada yukarıda anlatılanlara uygun olarak bütün scripts'ler ve css'ler tek bir dosya da toplanmıştır[7].

Bu bölümde; web tabanlı uygulamalarda performansın nasıl arttırıldığından bahsedilmektedir. Örnek çalışma olarak; üniversite öğrenci işleri otomasyonunda karşılaşılan yazılım performans sorunlarının nasıl giderildiği ve performans artırımı için yapılan çalışma süreci anlatılmaktadır[8].

Sisteminin Optimize Edilmesi İçin Yapılanlar

- S**tored Procuder'lerin Kontrolü:** Üniversite içerisinde geliştirilmiş olan öğrenci işleri veri tabanı incelenmiştir. Mevcut prosedürler içerisinde küçük çaplı normalizasyon işlemleri gerçekleştirilmiştir.

- I**ndex'leme Kontrolü:** Veri tabanı içerisinde index'leme yapılarak yoğun sorgulama işlemleri daha da hızlandırılmıştır.

- C**ursor Kullanımının Azaltılması:** Sorgular incelenerek gereksiz cursor kullanımları ortadan kaldırılmıştır.

- C**onnection Pooling:** Connection Poolling ile basitçe veritabanı bağlantılarının cachelendiği bir havuz oluşturulmuştur. Bu sayede uygulamamızda ihtiyacı olan bağlantılar bu havuzdan kısa sürede karşılanmaktadır. Sonuç olarak sisteme yük getiren bağlantı oluşturma işlemlerinden kullanılan connection pooling vasıtasıyla büyük oranda kurtarmıştır.

```
data source=localhost;initial
catalog=northwind;integrated
security=sspi;
pooling=yes; min pool size=20;
max pool size=200;
```

- Yukarıdaki bağlantı satırında görüldüğü üzere SQL Server da herhangi bir komut çalıştırdığımızda aynı havuza ait olan 20 adet bağlantı hazır olmaktadır.

- **Http İsteklerinin Azaltılması:** Http isteklerinin azaltılması ve sayfanın daha hızlı açılması için basit ve daha anlaşılır bir tasarım ilkesi belirlenmiş ve sayfa içerisinde kullanılan componentlerin sayısı azaltılmaya çalışılmıştır.

- **GZIP Componentleri Kullanımı:** Http server'la tarayıcı arasındaki veri iletimi GZIP içerisinde kullanılan algoritma ile sıkıştırılarak iletilir. Bu sayede daha küçük veri boyutları ile aktarım sağlanmış olur.

- **JavaScript ve CSS'lerin Kodun İçerisinden Ayrılması:** Kod içerisinde gömülü

olan js ve css kodlarının alınarak ayrı dosyalarda çalışmasını sağlanmıştır.



Şekil 4. Fire Bug ile Çalışma Süresi Ölçümü

Şekil 4'te örnek öğrenci işleri yazılımına ait css, js, resim, flash ve media bileşenlerinin hangi sıra ve sürede yüklenildiği görülmektedir.

- **Yazılım Kodu Optimizasyonu:** Yazılım içersindeki geliştirilen kodlar gözden geçirilerek bu kodların optimizasyonu sağlanmıştır. Bu optimizasyon işleminde bölüm 3.4'te anlatılan yazılım araçları kullanılmıştır.

Uygulama geliştirilme sırasında .net in debug mode'u kullanılarak koddaki hataların akışı görsel bir şekilde takip edilmiştir. Ancak yazılım performansı açısından ekstra yük getiren debug mod yerine relase mod ile derleme yapılmıştır. Bunun sebebi hataları ayıklayabilmemiz için çıktı dosyalarına ekstra bilgilerin(.pdb dosyaları) eklenmesi ve çalışma sırasında bu bilgilerin yorumlanmasıdır. Ayrıca Object veri tipini ihtiyaç dahilinde kullanılmış, bir web formunun ilk defa mı yoksa yeninden bir istek sonucu mu yüklendiğini anlamak için PostBack özelliği kullanılmış ve gereksiz PostBack işlemleri kaldırılmıştır.

5. Sonuçlar

Bu çalışmada web tabanlı sistemlerin, performansın arttırılması ile ilgili bir uygulama gerçekleştirilmiştir. Uzun uğraşlar sonucunda ortaya çıkan yazılımın nihai hedefi öğrenciye hızlı ve sorunsuz hizmet vermektir. Çok sayıda kullanıcının işlem yapması nedeniyle performansa bağlı memnuniyetsizlikler en az seviyede olmalıdır. Yapılan performans testleri ile her bir işlemin hedeflenen zaman diliminde tamamlandığı incelenmiştir. Ancak yazılımda performans artırımı kesin çizgilere bağlanmış bir konu değil süreklilik arz etmektedir.

Üniversite öğrenci işleri web uygulamasında, performans artırımına yönelik özetle aşağıdaki çalışmalar yapılmıştır:

- Stored Procuderlerin Kontrolü,
- Index'leme Kontrolu,
- Cursor Kullanımının Azaltılması,
- Connection pooling,
- Http İsteklerinin Azaltılması,
- Gzip componentleri Kullanımı,
- Javascript ve csslerin Kodun İçerisinden Ayrılması,
- Yazılım Kodu Optimizasyonu

Yapılan bu çalışmalar sonunda hedeflenen amaç doğrultusunda bir web tabanlı uygulama olan üniversite öğrenci işleri otomasyon sisteminin performansında önceki duruma göre erişimde % 42 artış sağlanmıştır.

6. Kaynaklar

[1] Dr.Erhan Sarıdoğan, **"Yazılım Mühendisliği"** Papatya Yayıncılık,2004

[2]Mustafa Alkan, **"YazılımSıfınları"** http://www.csharpnedir.com/makalegoster.asp?MId=230,2004

[3] Ali Öztürk, **"Veri Tabanı Yönetim Sistemi"**, Atlas yayın dağıtımı,2003

[4] Çevrimiçi: http://www2.itu.edu.tr/~kanoglu/crs-iscpm-wbarchitectures.pdf,2010

[5]Çevrimiçi: http://msdn.microsoft.com/en-us/library /aa178014%28v=SQL.80%29.aspx,2010

[6] Çevrimiçi: http://devcity.net/Articles/101/1/ants_profiler.aspx,2010

[7] Çevrimiçi:http://developer.yahoo.com/performance /rules.html,2010

[8] Çevrimiçi:http://www.csharpnedir.com/articles/rea d/?id=272,2010