T.C.
MALTEPE ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ARAMA MOTORLARINDA KULLANILAN ARAMA ROBOTU MİMARİLERİNİN İNCELENMESİ VE URL ATAMA İÇİN YENİ BİR YAKLAŞIM SUNULMASI

Ahmet Erdem KARACA

Yüksek Lisans Tezi

Tez Danışmanı
Yrd. Doç. Dr. Şenol Zafer ERDOĞAN

İSTANBUL – 2012

T.C.
MALTEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ARAMA MOTORLARINDA KULLANILAN ARAMA ROBOTU MİMARİLERİNİN İNCELENMESİ VE URL ATAMA İÇİN YENİ BİR YAKLAŞIM SUNULMASI

YÜKSEK LISANS TEZI

Ahmet Erdem KARACA

Tez Danışmanı
Yrd. Doç. Dr. Şenol Zafer ERDOĞAN

İSTANBUL – 2012

Bu tez çalışması, Maltepe Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 19/06/2012 tarih ve 2012/12 sayılı kararıyla oluşturulan jüri tarafından **Bilgisayar Mühendisliği Tezli Yüksek Lisansı Tezi** olarak kabul edilmiştir.

JÜRİ

Yrd.Doç.Dr. Şenol Zafer ERDOĞAN
Üye
(Danışman)

Yrd. Doç. Dr. Turgay Tugay BİLGİN
Üye

Yrd. Doç. Dr. Fatih YÜCALAR
Üye

ÖZET

Yüksek Lisans Tezi, Arama Motorlarında Kullanılan Arama Robotu Mimarilerinin İncelenmesi ve URL Atama İçin Yeni Bir Yaklaşım Sunulması, T.C. Maltepe Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı.

Günümüzde her gün katlanarak büyümeye devam eden internetteki web sayfaları, insanların bilgi üretmedeki hızını gösterir olmuştur. Üretilen bu bilgi miktarı o kadar büyük seviyelere ulaşmıştır ki aranılan doğru bilgiye ulaşmak imkânsız bir hal almıştır. Doğru bilgiye ulaşmak için ortaya konulan yöntemlerden biride arama motorlarıdır. İnsanlara aradıkları bilgiyi bulmalarında yardımcı olmuştur. Arama motorları, web sayfalarını bulmak ve içeriklerini öğrenmek için arama robotlarını kullanırlar. Web botları sayfaların içeriklerini tarar ve ziyaret edilecek başka sayfaları indekslerler.

Bu tez kapsamında arama robotu mimarileri incelenmiş ve bir sonraki ziyaret edilecek adresin bulunmasında daha popüler olan adresin seçimine öncelik veren bir model geliştirilmiştir.

Bu tez 2012 yılında yapılmıştır ve 71 sayfadan oluşmaktadır.

Anahtar Kelimeler: Crawler, Arama Robotları, Arama Motorları, Web Crawler, Web Spider.

ABSTRACT

Master Thesis, The analysis of the architectural design of Search Engines that are used in Search Robots and introducing a new approach for assigning URL, T.C. Maltepe University, Graduate School of Natural and Applied Sciences, Department of Computer Engineering.

Nowadays, the number of web sites on the internet that keep on growing increasingly has begun to indicate the speed of people to generate information. The amount of information produced has reached such a huge amount that it has become impossible to find out the right information. One of the methods that have been put forwarded to reach the right information is Search Engines. They help people to find the information they look for. Search Engines usually make use of Search Robots to find the web sites and learn their contents. Web robots scan the contents of the web sites and index the other ones that will be visited.

In the light of this thesis, the architectural establishments of Search Robots have been examined and a specific model that prioritizes the choice of popular web address in finding the visited web address has been developed.

This thesis has been completed in 2012 and consists of 71 pages.

Keywords: Crawler, Search Engines, Search Robots, Web Crawler, Web Spider.

TEŞEKKÜR

Tez konusunu belirlememde beni yönlendiren, çalışmalarım sırasında tecrübelerinden ve bilgilerinden istifade ettiğim, gerekli kaynakların sağlanmasında desteğini hiç esirgemeyen tez danışmanım Sayın Yrd. Doç. Dr. Şenol Zafer ERDOĞAN' a, çalışmamın hazırlanması sürecinde çok büyük emeği olan sevgili eşim Seda KARACA' ya ve dünyanın en sevimli kızı olduğu için sevgili kızım Bilge KARACA'ya teşekkürlerimi sunarım.

İÇİNDEKİLER

KISALTMALAR

Kısaltma	İngilizcesi	Türkçesi
Arpanet	Advanced Research Projects Agency Network	Amerikan Gelişmiş Savunma Araştırmaları Dairesi Ağı
IP	Internet Protocol	İnternet Protokol
URL	Uniform Resource Locator	Standart Kaynak Bulucu
TCP/IP	Transmission Control Protocol / Internet Protocol	İletim Denetim Protokol / İnternet Protokol
GFS	Google File System	Google Dosya Sistemi
HTML	Hyper Text Markup Language	Zengin Metin İşaret Dili
URI	Uniform Resource İdentifier	Standart Kaynak Belirleyici
RSS	Really Simple Syndication	Çok Basit Besleme

ŞEKİLLER

Sayfa

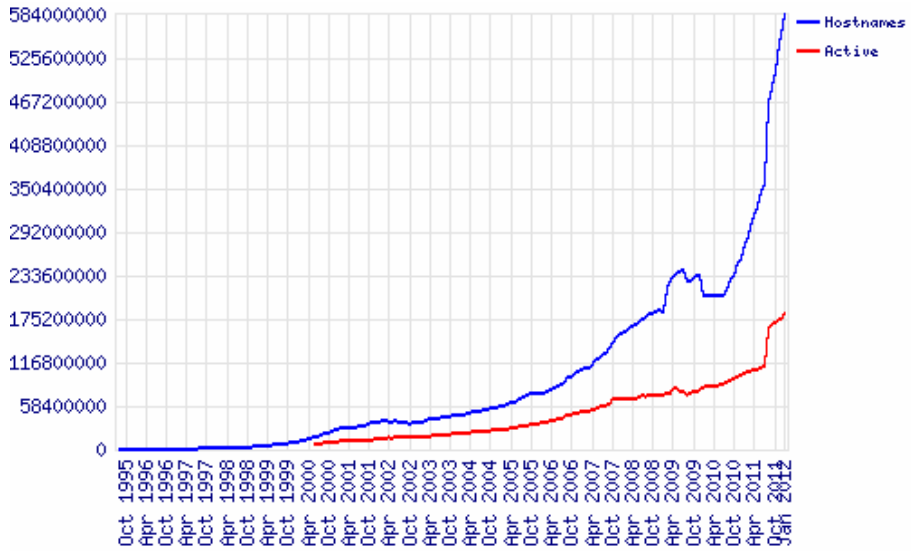
TABLOLAR

Sayfa

1. GİRİŞ

ABD Savunma Bakanlığı İleri Araştırma Projeleri Ajansı tarafından 1969 yılında geliştirilen ilk paket anahtarlamalı ağ Arpanet'tir. Üniversiteler arasında bilgi alış verişi için düşünülmüştür. Kullanımıyla beraber TCP/IP (Transmission Control Protocol / Internet Protocol) protokolünün ortaya çıkmasını sağlamıştır. Arpanet'e (Advanced Research Projects Agency Network) internetin ilk hali ya da başlangıcı denebilmektedir [1].

Arpanet'in ortaya çıkışından itibaren internet kullanımı inanılmaz seviyede büyümüştür. Günümüzde insanların çoğunun evinde internet bağlantısı mevcuttur. Bununla beraber insanlar evdeki sabit internetin yanı sıra mobil olarak da internete bağlanmakta ve bilgi üretmektedirler. Sosyal ağların gelişimi de bu bilgi üretimine büyük katkı sağlamıştır. İnternetin büyümesine paralel olarak ortaya çıkan bilgi miktarı katlanarak artmaktadır.



Şekil 1.1. Yıllara Göre Açılan Site Sayısı [2]

İlk dönemlerde bu bilgilere ulaşmak ve indekslemek nispeten kolay olsada ileriki dönemlerde gelişmeyle birlikte bu işlemlerin gerçekleştirilmesi zorlaşmıştır. İnternetteki sitelerin boyutu tam olarak tahmin edilemese de 584000000 üzerinde olduğu bilinmektedir [2]. Bu boyuttaki bilgi yığını içerisinde aranılan bilginin bulunması tabiki oldukça zor olmaktadır. Bilgi miktarındaki artışla beraber, istenilen bilgiye ulaşmak için arama motorları ortaya çıkmıştır. Arama motorları sayesinde aranılan bilgiye erişim kolaylaşmıştır.

Arama motorları, web sitelerini tarayan ve indeksleyen arama robotlarına sahiptirler. Bu arama robotları verilen ilk adresin içeriğini veritabanına kaydedip; sayfayı ziyaret ederler. Ziyaret ettikleri sayfayı tarayıp içerisindeki yeni adresleri bulup daha sonra ziyaret edilecek adres listesine yazmaktadırlar. Arama motorları, kullanıcıya kullanıcı arayüzü, aranacak anahtar kelimeleri girmesine imkan tanıyan giriş alanı ve sonuçları görüntüleme imkanı sağlamaktadır.

Bu tez çalışmasında arama robotunun ziyaret edeceği bir sonraki adresin belirlenmesinde kullanılacak modelle ilgili bir yaklaşım sunulmaktadır. İndekslenen ama daha ziyaret edilmemiş adresler, üç boyutlu uzayda ifade edilmeye çalışılmış ve yoğunluğun olduğu alanlara öncelik verilmeye çalışılmıştır. Yoğunluğun olduğu alanlardaki adreslerin daha popüler olduğu varsayılmış ve bu alanlardaki adreslerin öncelikli olarak ziyaret edilmesi sağlanmıştır.

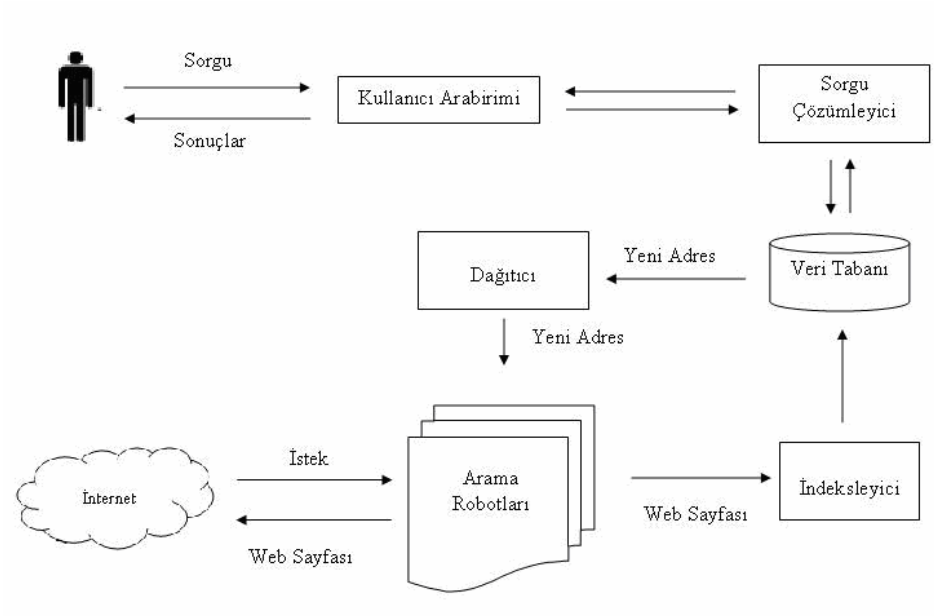
Popüler adreslerin bulunması için, ziyaret edilecek adres listesindeki adresler kök,yol ve sorgu olarak üç parçaya ayrılmıştır. Ayrılan bu parçalar kendi içlerinde tekrar edilme sayılarına göre sıralanmış ve bu sıraya göre puanlanmıştır. Bu sıralamaya göre en yüksek puanı alan adreslerin daha popüler olduğu kabul edilmiştir. Bu popüler adres alanları içerisinde olmayan, popülerlik puanı daha az olan adreslerinde indekslenebilmesi amacı ile zaman parametresi modele eklenmiştir.

Bu tez çalışmasında 2. bölümde arama motorları konusu incelenmektedir. 3. bölümde arama robotlarından bahsedilmektedir. 4. bölümde URL atama için yeni bir yaklaşım sunulmuştur. 5. bölümde ise sonuç kısmı ile tez tamamlanmaktadır.

2. ARAMA MOTORLARI

Arama motorları bilginin aranması ve gösterimi için kullanılmaktadır. Arama motorları üç parçadan oluşmaktadır. Bunlar ”Arama Robotu” , ”Arama İndeksi” ve ”Arama Motoru Arayüzü” dür.

- **Arama Robotu :** Arama robotları, web sitelerinin içeriklerini bulup getirmeye yarayan yazılımlardır. Verilen bir başlangıç adresinden başlayarak içeriğini aldığı sitedeki diğer adresleri bulur ve bunları ziyaret edilecek adres listesine ekler.
- **Arama İndeksi: Arama** robotlarından gelen bilgilerin düzenli bir şekilde kaydedilmesi, arama motoru indeksinin yöntemine göre; kullanıcıdan gelen arama sorgusuna cevap verilmesi sağlamaktadır.
- **Arama Motoru Arayüzü:** Arama motorları kullanıcılardan gelen sorguların, arama motoru indeksine iletilebilmesi ve gelen cevapların kullanıcılara gösterilebilmesi için arama motoru arayüzünü kullanırlar.



Şekil 2.1. Arama Motoru Bileşenleri

Şekil 2.1'de gözüktüğü gibi kullanıcıdan gelen sorgu kullanıcı arabirimi sayesinde sorgu çözümleyiciye iletilmektedir. Sorgu çözümleyici yöntem, veri tabanından kullanıcının sorugusuna uyan kayıtları kullanıcı arabirimi sayesinde kullanıcıya ulaştırmaktadır. Arama robotları ise dağıtıcıdan aldıkları adresi, internete ulaşarak içeriğini elde edip ulaştıkları bu web sayfalarını inceledikten sonra indeksleyiciye göndermektedirler. İndeksleyici ise bu bilgileri arama motorunun veri tabanına kaydetmektedir.

İstenilen bilginin bulunması için aranılan bilginin nerede olunduğunun bilinmesi gerekmektedir. Bunun için bir indeksleme işleminine ihtiyaç olmaktadır. Web sitelerinin durmadan artışının etkisiyle, artık bu bilgileri elle indeksleme imkanı kalmamıştır. Bu noktada arama robotları devreye girmektedir. Arama robotları, hızlı bir şekilde web sitelerini tarayarak içeriklerini veri tabanlarına kayıt etmektedir.

Buldukları içeriklerden, tarama yaparak gidebilecekleri diğer adresleride çıkararak devamlı bir şekilde taramalarına devam etmektedirler. Bu işlemlerin hızlı olması ve taranacak bilgi miktarının fazlalığı yüzünden, paralel arama robotları oluşturulmuştur.

Bu robotlar birbirileri ile haberleşerek aynı anda birden fazla adresi tarayabilmektedirler. Taranan bu bilgiler veri tabanına kaydedilmekte ve veri tabanına kaydedilen bu verilerin büyüklüğü aranan içeriğin bulunma olasığını arttırmaktadır.

İndekslenen her web sitesinin içeriğininde veri tabanında saklandığı düşünülürse oluşacak veri miktarı çok büyük olmaktadır. Bu yüzden arama motorları için çok büyük işlem gücü ve depolama kapasitesi gerekmektedir.

Arama motorlarının indeksledikleri bilgiler veri tabanına yazıldıktan sonra içeriği değerlendirilmeye alınır. Değerlendirme aşamasında içeriğin hangi kelimelerle alakalı olduğu, hangi arama sonuçlarında listeleneceği ve hangi sırada gösterileceği belirlenmektedir.

Her arama motorunun kendine özgü değerlendirme algortimaları vardır. Arama motorları, aranan kelimeler ile ilgili içerikleri eşleştirmek için kendine özel olan bu yöntemleri kullanmaktadır. Yöntemin isabet oranı ne kadar iyi ise aranan kelime ile ilgili o kadar isabetli sonuçlar getirmektedir.

Arama motoru arayüzü ise arama yapılacak kelime yada kelimeleri girerek arama motorlarının indekslerinde var olan içeriklerden aranılan bilgilerin bulunmasını sağlamaktadır. Arama motoru arayüzü, arama motorunun indeksine ulaşarak, aranan kelime ile ilgili bilgileri, yine arama motoru arayüzünde listeleyip göstermektedir.

Arama motoru arayüzü; arama yaparken farklı içerik tiplerinde arama yapılmasına da imkan verebilmektedir. Bunlar resim,video ve doküman gibi çok çeşitli içerik tipine sahip olabilmektedir.

Günümüzde bir çok arama motoru bulunmaktadır. Bunlar yatay arama yapan genel arama motorları olabileceği belirli konulara odaklanmış dikey arama motorlarıda olabilirler.

Yatay arama motorları her hangi bir konu kısıtı olmadan tüm interneti indekslemeye çalışmaktadırlar. Arama robotları sayesinde ziyaret edilen sayfalardaki tüm adresleri, takip edip indesklemektedirler.

Dikey arama motorları ise yatay arama motorlarından farklı olarak önceden belirlenmiş konular üstüne indeksleme yapmaktadırlar. Verilen konu üzerine, sayfanın içerisinde konu ile alakalı kelimeleri takip edebilecekleri gibi, aynı zamanda adreslerdeki konu ile ilgili kelimeleri de takip edebilmektedirler. Bu sayede sadece belirlenmiş konu ile alakalı adresler elde edilmektedir.

Yatay arama motorlarına Google, Yahoo, Bing ve Baidu örnek olarak verilebilir. Dikey arama motolarına örnek olaraksa yine Google'ın dikey arama hizmeti veren resim,kitap ve video gibi konu odaklı arama motorları verilebilir.

2.1 Arama Motorlarının Geçmişi

Arama motorlarının ortaya çıkış nedenlerinden biri de bilginin internette hızlı ve kolayca bulunamamasıdır. Bu bilgilerin indekslenmesine ilk başlarda imkan olsada, hızlı büyümeye paralel olarak elle indeksleme imkanı ortadan kalkmıştır.

Arama motoru terimi ilk olarak 1990 yıllarının başında archie ile başlamıştır. Archie, Alan Emtage tarafından kurulmuştur [3]. Archie popüler olunca başka arama motorlarıda ortaya çıkmıştır. Ama bunlar tam anlamıyla bir arama motorunun tüm bileşenlerine sahip olamamışlardı.

Bir arama robotu bileşenlerine sahip ilk arama motoru, 1993 yılında Matthew Gray tarafından geliştirilen ”Wandex” ’dir. Perl dilinde geliştirimiştir [4]. Yine aynı tarihlerde Aliweb arama motoru geliştirilmiş fakat herhangi bir arama robotu kullanılmamıştır.

Jumpstaion arama moturu 1993 senesinde geliştirilen arama motorlarından biridir [5]. Jumpstation bir arama robotu kullanarak siteleri indeksleyip sonuçlarını bir sorgu sayfasında sergileyebilmekteydi.

Bu tarihe kadar yapılan tüm arama motorları kısıtlı bir içeriği indeksleyebilmektedir. Tüm metin indeksleme işlemini yapan ve bir arama moturunun tüm bileşenlerine sahip ilk arama motoru 1994’de üretilen WebCrawler’dır [6]. Bu arama moturu kullanıcıya web sayfasındaki tüm kelimeler için arama imkanı sağlamaktaydı. Bu tarihten sonrada bir çok arama motoru üretilmiştir: Altavista, İnfoseek, Yahoo ve Lycos bu arama motorlarına örnek verilebilir. Bunlardan günümüzde de en bilinenleri Yahoo, Stanford Üniversitesi öğrencileri Jerry Yang ve David Filo tarafından kurulmuştur [7]. O dönemlerde oldukça popüler olan Yahoo, büyük arama motorlarından biri olarak hizmet vermeye devam etmektedir.

1999 yılında ise Larry Page ve Sergey Brin tarafından Google kurulmuştur. Google arama motoru internette her sayfayı değerlendiren *PageRank* yöntemini kullanarak büyük bir gelişme sağlamıştır [8]. Google arama moturu olmanın yanında mail, video gibi hizmetleri de vererek büyümesine devam etmiştir.

Google Çin hükümeti tarafından yasaklanınca, Baidu arama motoru 2000 yılında Robin Li ve Eric Xu tarafından kurulmuştur [9].

Bing arama motoru 2009 yılında Microsoft tarafından geliştirilmiş ve ”Windows Live Search” yerine kullanılmaya başlanmıştır [10]. 40 farklı dilde hizmet vermektedir. Bing, arama sorgusu için verilen kelimeleyi anahtar kelime olarak görmekten farklı olarak aramayı yönlendiren bir ipucu olarak görüp konu ile alakalı içerikleri getirmektedir.

2.2 Arama Motoru Bileşenleri

Arama robotları web sitelerine gidip içeriklerini indeksleyiciye gönderirler ve yeni bir adres alarak aynı işlemi yapmaya devam ederler. İndeksleyici, arama robotundan gelen bilgileri işleyerek veritabanına kaydeder. Arama motoru arayüzü ise gelen arama sorgusunu indeksleyiciye göndererek, indeksleyiciden gelen bilgileri Arama motoru arayüzünde sergilerler.

2.2.1 Arama Robotu

Arama robotları, web sitelerinin içeriklerini bulup getirmeye yarayan yazılımlardır. Verilen bir başlangıç adresinden başlayarak içeriğini aldığı sitedeki, diğer adresleri bulur ve bunları ziyaret edilecek adres listesine ekler. Bundan sonra kendi yöntemine göre ziyaret etmesi gereken bir sonraki adresi bulur ve listedeki adresler bitene kadar

bu işlemlere devam eder. İnternetin büyüklüğü düşünülürse, tek bir arama motorunun tüm interneti kontrol etmesi mümkün değildir. Bu yüzden paralel çalışabilecek arama robotları yazılmıştır. Bunlar birbirleri ile ilişkili bir şekilde çalışılmaktadırlar.

Her arama motoru kendine özgü arama robotları oluşturmuştur. Günümüzde bilinen 250'den fazla arama robotu vardır [11]. Bunlardan en bilinenleri, Google “Google Bot”, Yahoo “Yahoo! Slurp”, Baidu “Baiduspider” ve Bing “bingbot”’dur.

2.2.2 Arama Motoru İndeksi

Arama robotlarından gelen bilgilerin düzenli bir şekilde kaydedilmesi, arama motoru indeksinin yöntemine göre; kullanıcıdan gelen arama sorgusuna cevap verilmesi arama motoru indeksinin görevlerindendir.

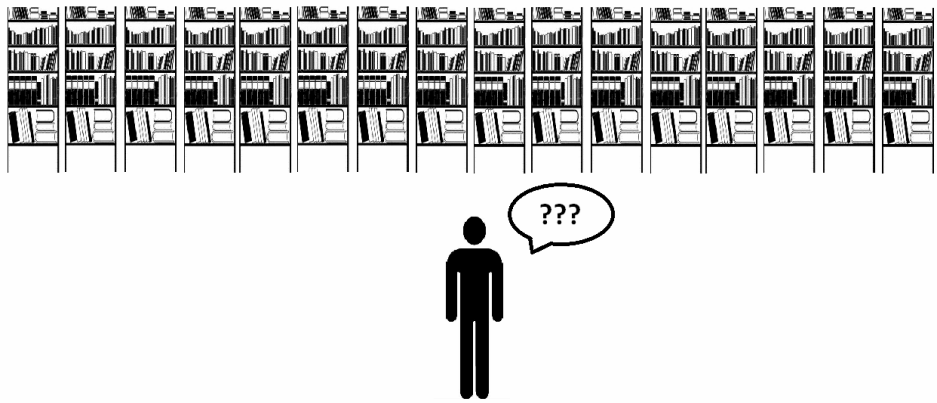
Arama robotlarının sitelerden bulup getirdiği bilgiler, arama motorunun veri tabanına kaydedilirler. Bu bilgiler arasında, arama robotu tarafından ziyaret edilen adresten ulaşılan diğer adresler ve adresin içeriği bulunmaktadır. Kullanıcı tarafından gelen sorgular, bu bilgiler ve arama motoru indeksinin yöntemi kullanılarak arama motoru indeksi tarafından cevaplandırılır.

2.2.3 Arama Motoru Arayüzü

Arama motorları kullanıcılardan gelen sorguların, arama motoru indeksine iletilebilmesi ve gelen cevapların kullanıcılara gösterilebilmesi için arama motoru arayüzünü kullanırlar. Arama motoru arayüzü, kullanıcının verdiği anahtar kelimeleri indekse gönderirken özel komutları da kabul edebilir. Bunlar ve/veya, dâhil etme gibi mantıksal aramayı güçlendiren komutlar olabilmektedir.

3. ARAMA ROBOTU

Aramak ve aranılan bilgiye ulaşmak her insan için önemlidir. Ulaşma imkanı olmadıktan sonra yığınlarca bilgi sahibi olmanın pek bir anlamı olmamaktadır. İnternet, bir bilgi havuzudur. Her gün milyonlarca içerik üretilmektedir. Bu içerik ve bilgi havuzu insanlar için ulaşılabilir olmadıkça bir anlam ifade etmemektedir.



Şekil 3.1. Kütüphane Örneği

Bir kütüphane örneğinde binlerce kitap içerisinde aranılan konu ile alakalı kitapların listesi bulunamazsa, ortadaki bilgi yığınının bir anlamı olmamaktadır. Kitapları tek tek kontrol ederek içeriklerinin ilgili olup olmadığını belirlemekte gerçekçi bir çözüm olmaktan uzaktır. Bir sonraki aramada aynı işlemlerin tekrar edeceği göz önününe alınırsa, kütüphane için bir indeks kullanmanın faydaları ortaya çıkmaktadır. İnternet gibi büyük bir bilgi kaynağı da düşünülecek olursa, buradaki bilgilerin indekslenmesinin önemi açıkça görülebilmektedir.

Önceden internetteki site indekslerinin, elle tutulması arama ve bulmayı kolaylaştırmak için yardımcı bir çözüm olarak gerekliysede; günümüzde insanlar

birkaç tıklama ile aradıkları tüm bilgiye ulaşabilmektedir. Günümüz arama motorları, aranılan bilgilere hızlı ve daha doğru cevap vermeye başlamıştır.

Bu işlerin gerçekleşebilmesi içinde arama motorları, arama robotlarını kullanmaktadır. Bunun nedeni, bir insanın kütüphanedeki tüm kitapların içerisindeki kelimeleri indekslemeye çalışması, kolay ve güvenilir bir çözüm değildir. Bunun yerine arama robotları, tek tek her sayfayı ziyaret ederek içeriklerini bizim yerimize kontrol etmekte, ilgili kelimelerle ilişkilendirmekte ve sonraki aramalarda da kullanma imkanı vermektedir.

Arama robotları internetteki sayfaları bizim yerimize ziyaret eden, bunları indeksleyen ve devamlı çalışan yazılımlardır. Bu işlemleri yapabilmek için gelişmiş teknolojilerden faydalanırlar.

Bir arama robotu, verilen bir yada birkaç başlangıç adresine sahiptir. İlk olarak verilen başlangıç sayfasını indirilmesi gerekmektedir. İndirilen sayfaların içeriği arama motorunun veritabanında saklanmaktadır. Hızlanan internet bağlantıları ile bu işlemler kolay olarak düşünülsede, arama robotunun çalıştığı sunucunun bant genişliğinin, disk hacminin ve işlemci gücünün çok büyük ve hızlı olmasını gerektirmektedir. Bu kadar büyük bir bilginin de tek bir sunucu tarafından saklanması ve işlenmesi tabiki mümkün değildir.

Bunun için arama motorları binlerce sunucu kullanmaktadır. Sunuculardaki verilerin ve işlemlerin paylaşılabilmesi için, arama motorları GFS (Google File System) gibi özel dosya sistemleri kullanmaktadırlar [12]. Bu özel dosya sistemleri sayesinde tüm sunucular, tek bir sunucuymuş gibi işlemleri paralel şekilde gerçekleştirmeye ve ortak veri havuzuna erişimi sağlamaktadır. Aynı zamanda sunucu kümeleri farklı lokasyonlara yerleştirilerek, kullanıcıya olan mesafe azaltılıp cevap verme süreleri hızlandırılmaktadır.

Arama robotları tarafından indirilen sayfaların hatalı olup olmadıkları, içeriklerinin metin yada medya olup olmadığının denetlenmesi; varsa hata kodlarının

oluşturulması gerekmektedir. Oluşturulan hata kodları arama robotlarına, tekrar ziyaret politikalarında yardımcı olmaktadır. Eğer indirilen sayfa bir web sayfası ise ve her hangi bir hata koduna sahip değilse; artık arama robotu tarafından incelenmeye hazır durumdadır.

İndirilen sayfadaki kodlar arama robotu tarafından parçalara ayrılmaktadır. Bu parçalara ayırma işlemi yapan yazılıma ”Ayrıştırıcı” denilmekte ve her arama robotunun içinde bir tane bulunmaktadır. Ayrıştırıcı, sayfanın içerisindeki HTML (Hyper Text Markup Language) etiketlerini bulup, içeriklerini bu etiketlere göre yorumlamaktadır.

Öncelikli olarak sayfanın içerisindeki ”<a href>” etiketlerindeki adresler tespit edilmektedir. Tespit edilen adreslerin doğruluğu RFC-3986 standartına göre kontrol edilip düzeltilmektedir [13]. Kontrol edilen ve düzeltilen bu adresler arama moturunun veritabanındaki ziyaret edilecek adresler listesine kayıt edilmektedir.

Günümüzdeki arama motorları, artık sadece web sayfalarını indekslememektedir. Sayfalardaki medyalarıda indekslemekte ve veritabanlarında saklamaktadırlar. Bu sayede web sayfalarının haricinde sadece resimlerin aranması gibi dikey aramalar da gerçekleşebilmektedir.

Arama robotunun ziyaret edeceği bir sonraki adres ise arama motorunun ziyaret edilmemiş adresler tablosu ve URL (Uniform Resource Locator) atamada kullanılan dağıtıcı yöntemi sayesinde yapılmaktadır. Bu yöntem arama robotlarının ziyaret edecekleri adresleri belirlediği için oldukça önemlidir. Paralel çalışan arama robotları için aynı sayfaları ziyaret etmemesini sağlamak, hangi sayfaların yeniden ziyaret edileceğinin belirlenmesi gibi önemli görevleri vardır.

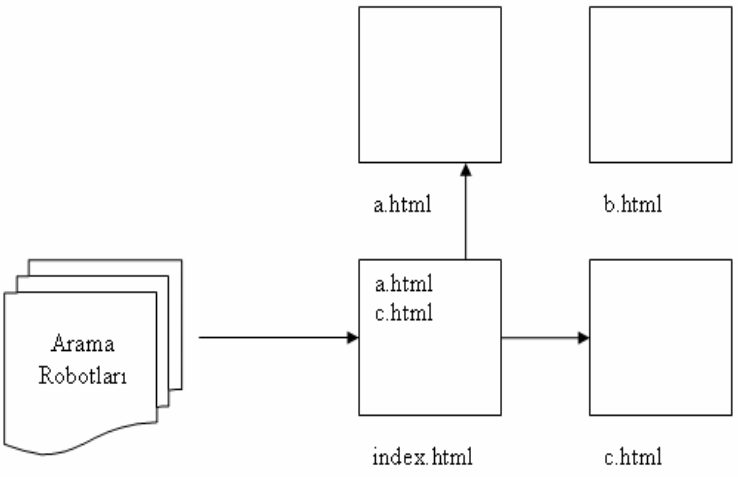
Arama robotları bu şekilde ziyaret edecekleri adresler bitene kadar devamlı olarak çalışmaktadırlar. Tekrar ziyaret politikaları sayesinde ise hem sayfaların güncellenmesini takip ederler hemde çalışmalarına devam ederler.

3.1 Derin Web

Arama motorları interneti indekslerken, arama robotları sayesinde bir başlangıç adresinden başlamak koşulu ile ziyaret ettikleri sayfadaki işaret edilen diğer adresleri de daha sonra ziyaret etmek üzere indekse yazmaktadırlar. Böylece indekse ziyaret edilmesi için devamlı yeni adresler eklenmiş olur. Bu işlemde fark edilen en büyük sorun eğer bir web sayfası diğeri tarafından işaret edilmemişse ne olacağıdır. Arama robotları işaret edilmemiş bir sayfayı bulamamaktadır. Bu işaret edilmemiş sayfalar, bu yüzden arama motorları tarafından indekslenemezler.

İşaret edilmeyen sayfalar, sayfanın dinamik bir içerik tarafından yaratılması, yani bir sorgu sonucu üretilmesi, başka hiç bir sayfa tarafından işaret edilmemesi, özel üyelik isteyen bir web sitesinin olması, belirli IP (Internet Protocol) adreslerinin girişine izin verilen siteler olması, javascript kodları ile işaret edilen sayfalar ve sayfanın html içeriğine sahip olmaması olarak ifade edilebilir. İşte internetin, bu sayfaların oluşturduğu kısmına "Derin Web" denilmektedir [14].

Şekil 3.2'de görüldüğü gibi, arama robotları index.html'i tararlarken a.html ve c.html adreslerine ulaşmışlardır, fakat hiç bir yerde adresi olmayan b.html, arama robotu tarafından bilinmemektedir.

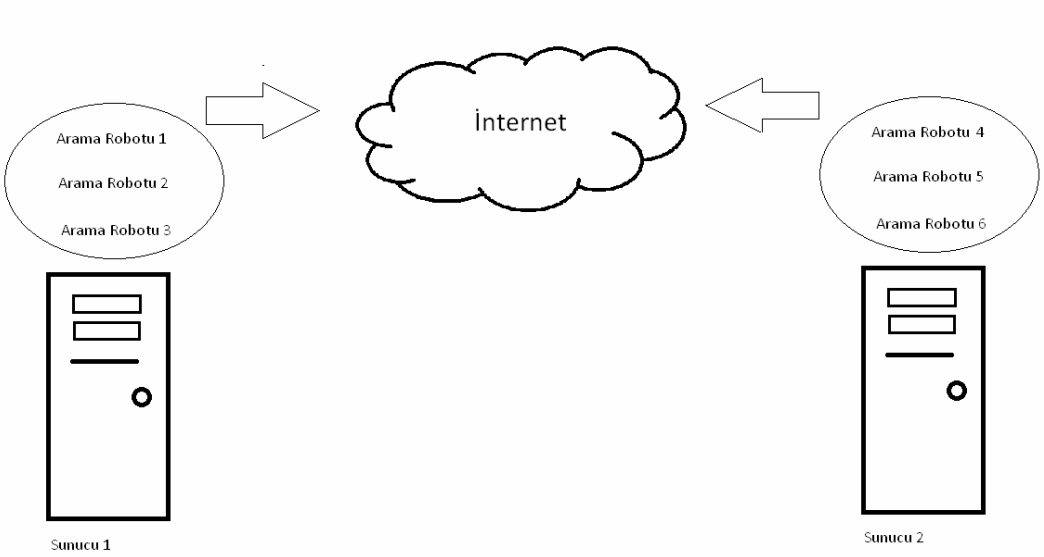


Şekil 3.2. Derin Web Gösterimi

Derin web'in ulaşılabilinen yani arama motorları tarafından indekslenen kısmına oranla çok daha büyük olduğu tahmin edilmektedir [14]. Arama motorları bu tür indekslenemeyen sayfalara ulaşabilmek için farklı yöntemler kullanabilmektedir. Eğer bir sayfa kullanıcı formundan gelen bir parametreye göre bir sorgu çalıştırıp sonuç döndürüyorsa; arama motoru sanki kullanıcı parametre göndermiş gibi sorguyu çalıştırmaya ve sayfa üretmeye zorlayabilmektedir. Bu sayede önceden ulaşamadığı sayfalara ulaşma imkânı oluşturulmaktadır [15].

3.2 Paralel Arama Robotu

İnternetin büyüklüğü düşünüldüğünde, arama motorları tarafından indekslenilmesi için tek bir arama robotunun yeterli olmadığı açıktır. Birden fazla arama robotunun çalışması ile indekslenen bilgi miktarı arttırılabilinir. Aynı anda çalışan arama motorları yazmak ve verimli kılmak arama motorlarının gereksinimlerinden biridir. Bu sayede ziyaret edilen sayfa sayısı artmakta ve arama robotları tarafından bant genişlikleri etkin olarak kullanılmaktadır. Bu tür arama robotlarına "Paralel Arama Robot" 'ları denir [16]. Şekil 3.3'de görüldüğü gibi farklı sunucular üzerinde birden fazla arama robotu paralel olarak çalışmaktadır.



Şekil 3.3. Paralel Arama Motorları

Paralel arama robotlarının yazılmasında karşılaşılan bir takım zorluklar vardır: Aynı sayfanın ziyaret edilmeye çalışılması, aynı sayfanın indekse yazılması ve arama motoru indeksine erişim sırasında ortaya çıkabilecek sıkıntılar. Bu sorunların oluşmaması, hızlı ve etkin bir paralel arama robotu geliştirmek arama motorları için önemlidir. Bilinen arama motorları, arama robotlarının paralel şekilde çalışması için tasarlamışlardır [17].

3.3 Robot Engelleme Standardı

Arama robotları bir siteyi ziyaret ederken, o sitedeki tüm sayfaları arama motoru indekslerine eklemeye çalışırlar. Herhangi bir kısıt olmaksızın yapılan bu işlem, içeriğinin diğer kişiler tarafından bulunabilir olmasını istemeyen kişileri rahatsız edebilir. Arama robotu ve site yöneticileri arasında bir protokol oluşturarak sitedeki, hangi sayfaların arama robotları tarafından ziyaret edilmemesi gerektiğinin arama robotlarına bildirilmesi için Robots.txt standardı oluşturulmuştur [18]. Bu protokol için herhangi bir resmi kurum ve çalışma grubu yoktur. Şekil 3.4'de Robots.txt örneği görülmektedir.

```
User-agent: *
Crawl-delay: 10
# Directories
Disallow: /includes/
Disallow: /misc/
# Files
Disallow: /CHANGELOG.txt
Disallow: /cron.php
Disallow: /INSTALL.mysql.txt
Disallow: /INSTALL.pgsql.txt
# Paths (clean URLs)
Disallow: /admin/
Disallow: /comment/reply/
Disallow: /filter/tips/
# Paths (no clean URLs)
Disallow: /?q=admin/
Disallow: /?q=comment/reply/
Disallow: /?q=filter/tips/
```

Şekil 3.4. Robots.txt Örneği

Bu standartta, Robots.txt adlı dosya sitenin kök dizininde oluşturulur ve arama robotu bir siteyi ziyaret etmeye başladığında Robots.txt dosyasının mevcut olup olmadığını kontrol eder. Eğer bu dosya arama robotu tarafından tespit edilirse içeriği okunarak ziyaretine izin verilen sayfalar belirlenir ve arama robotu tarafından ziyaret edilmeye başlanır. Arama robotları bu protokole uymak zorunda değillerdir. Robots.txt dosyasının kontrolü arama robotları için tamamen tavsiye niteliğindedir. Bunun için Robots.txt dosyasının konması sayfaların ziyaret edilmemesini sağlamamaktadır.

3.4 Site Haritası

Arama robotlarına bir sitedeki tüm sayfaları bildirmek için Sitemap.xml dosyası kullanılmaktadır. Bu sayede arama robotları tarafından bulunamayacak olan sayfalarda arama motoru indeksine eklenebilmektedir. Bu standart, site yöneticilerine, arama robotları için sayfaların öncelik derecelerini, güncellenme sıklığını ve en son ne zaman güncellendiği bilgisini verme imkânı sağlamaktadır. Sitemap.xml ’in en büyük faydası derin web olarak adlandırılan, arama robotları tarafından bulunamayacak olan sayfalarında arama robotlarına bildirilebilmesidir. Şekil 3.5’de bir Sitemap.xml örneği verilmektedir.

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
   <url>
      <loc>http://www.example.com/</loc>
      <lastmod>2011-01-01</lastmod>
      <changefreq>monthly</changefreq>
      <priority>0.8</priority>
   </url>
   <url>
      <loc>http://www.example.com/catalog?item=50</loc>
      <changefreq>weekly</changefreq>
   </url>
   <url>
      <loc>http://www.example.com/catalog?item=70</loc>
      <lastmod>2011-12-23</lastmod>
      <changefreq>weekly</changefreq>
   </url>
   <url>
      <loc>http://www.example.com/catalog?item=80</loc>
      <lastmod>2011-12-23T18:00:15+00:00</lastmod>
      <priority>0.3</priority>
   </url>
   <url>
      <loc>http://www.example.com/catalog?item=90</loc>
      <lastmod>2011-11-23</lastmod>
   </url>
</urlset>
```

Şekil 3.5. Sitemap.xml Örneği

Sitemap. xml dosyası web sitesindeki tüm adreslerin listelendiği bir haritadır. Google, Yahoo ve MSN tarafından desteklenmektedir. Kullanılan en son sitemap protokolü 0.9'dur [19]. Sitemap.xml dosyası web sitesinin kök dizininde olabileceği gibi, eğer başka bir dizinde bulundurulmak isteniyorsa, Robots.txt dosyası içinde yeni yerin konumu da yazılabilinmektedir.

3.5 Tekrar Ziyaret Politikaları

Arama robotları sayfaları ziyaret ettiğinde sayfanın o anki halini indekslemektedirler. İnternet ise dinamik bir yapıya sahiptir. Web sayfalarının içerikleri her gün güncellenmektedir. Haber sitelerinde ana sayfalarındaki içerik her saat başı değişmektedir. Bu kadar dinamik ve değişimin hızlı olduğu bir yerde, sayfaları bir kere indekslemek yeterli olmamaktadır. Bu yüzden arama robotları için tekrar ziyaret

politikaları oluşturulmuştur. Bu tür güncel içerik bulunduran sayfalar için daha sık güncelleme yapan, bunun yanında içeriğinin değişmesi daha az olan ya da değişim olmayan sayfalar için uzun aralıklı tekrar ziyaret yöntemlerini geliştirilmiştir.

3.6 Nezaket Politikaları

Arama robotlarının amacı, kısa zaman içerisinde çok fazla sayıda sayfayı indekslemektir. Bunu yaparken, devamlı sayfaları ziyaret etmekte ve içeriklerini sunucularına indirmektedirler. Bu işlemler sırasında, siteleri yayınlayan sunucular açısından sıkıntı verici durumlar oluşmaktadır. Bu durumlar:

- Ağ bağlantı kaynaklarının tüketilmesi,
- Sunucuların aşırı yüklenmesi,
- Kullanıcıya verilen hizmetlerin kesintiye uğraması.

Bu sorunların oluşmasını önlemek için arama robotları için nezaket politikaları oluşturulmuştur. Arama robotları durmaksızın devam eden bağlantı istekleri yerine daha istikrarlı ve kontrollü bağlantı istekleri, sunucuların yoğun olduğu zamanların dışında ziyaret zamanları gibi çeşitli yöntemleri kullanmaktadırlar [20].

3.7 Arama Robotu Çeşitleri

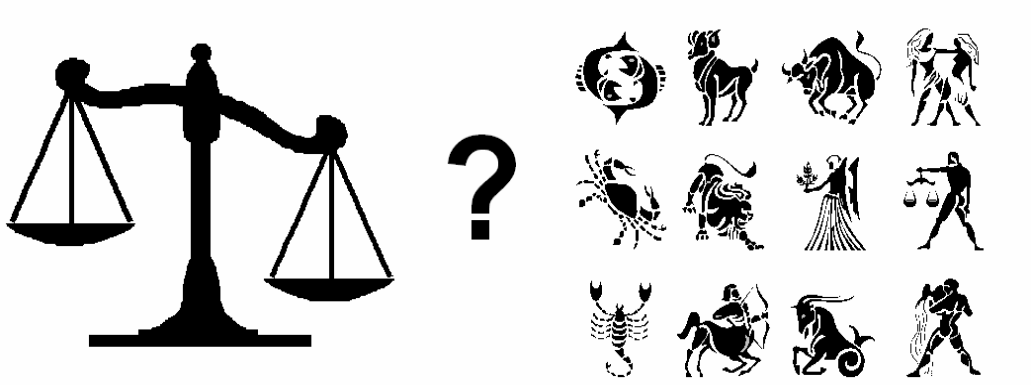
Arama motorlarının sonuç kümelerinde, aranan kelimelerle daha ilgili sonuçlar döndürebilmek için farklı türde arama robotu türleri geliştirilmiştir. Bu sayede verimin arttırılması amaçlanmıştır. Dizin arama gibi sadece bir konu üzerinde arama yapan robotlar olabileceği gibi bir siteye özel arama robotu da olabilmektedir.

Günümüzde arama motorları hem genel amaçlı arama robotlarını kullanıyorken hem de özel amaçlar için tasarlanmış arama robotlarını kullanmaktadır [21].

3.7.1 Yatay arama robotları

Evrensel arama robotu olarak da adlandırılabilinen yatay arama robotları, genel arama motorlarının temelini oluşturmaktadırlar. Adrese dayalı arama robotları verilen ilk adresten başlayarak ziyaret ettikleri sayfadaki diğer adresleri, ziyaret edilecek adres listesine eklerler ve konu ya da site ayrımı yapmadan görevlerine devam ederler. Belirli bir kısıtları olmadığından tüm internetteki sayfaları indekslemeye çalışmaktadırlar. Bu tür arama robotlarının faydası, çok geniş ölçekte sayfayı indeksleyebilmesidir. Böylece arama motoru için sorgulara cevap verirken kullanacağı büyük bir havuzun oluşmasını sağlarlar.

Adrese dayalı arama robotlarının sağladığı faydaların yanında zorlukları da vardır. Herhangi bir konu kısıtlı olmadığı için ziyaret edilecek sayfaların büyüklüğü, bu büyüklükteki bilginin saklanması ve uygun şekilde indekslenmesi arama motorları için büyük bir yük getirmektedir. Bunların yanında aranan kelimeler ile alakalı olmayan sonuçların gelmesi de mümkün olabilir.



Şekil 3.6. Sorgu Örneği

Şekil 3.6'da bir sorgu örneği görülmektedir. Eğer "TERAZİ" kelimesi ile burçlarla ilgili adrese dayalı arama motorunda arama yapılacak olursa, terazi burcuyla ilgili sonuçların yanında tartı aleti olan teraziye ait sonuçlar da getirilebilir. Bu yüzden gelen sonuçların içerisinde aranılanı bulmakta zor olabilir.

3.7.2 Dikey arama robotları

Evrensel arama robotları tüm adresleri ziyaret etmeye çalışırlarken, esasında ilgilenilen ana konuya odaklanmazlar. Adrese dayalı arama robotları bulabildikleri tüm sayfaları ziyaret etmeye çalışmaktadırlar. Dikey arama robotları, adrese dayalı arama robotlarının aksine, tüm adresleri takip etmezler. Belirlenmiş bir konu ile alakalı olan adreslere yönelirler. Bu sayede arama motorlarının sonuç kümeleri daha isabetli olabilmektedir.

Belirlenmiş konu ile ilgili sayfa içerisindeki kelimeleri takip edebilecekleri gibi adreslerdeki kelimeleri de takip edebilmektedirler. Bu sayede sadece belirlenmiş konu ile alakalı adresler elde edilmektedir.

Günümüzde popüler olan arama motorları hem evrensel bir arama hizmeti vermekte hem de dikey arama yapılmasına imkân veren hizmetler sunmaktadırlar. Örneğin, Google arama motoru üzerinden genel arama yapma imkânı vermekteyse de bunun yanında sadece resimler, haber, doküman gibi belirlenmiş konular üzerinde de arama imkânı da sunmaktadır.

3.7.3 Blog tabanlı arama robotları

Bloglar, kişilerin kendileri için günlük tuttukları web sayfalarıdır. Bloglar günümüzün en önemli bilgi paylaşım araçlarından biri haline gelmiştir. Milyonlarca

insan bloglarında kendileri ya da ilgilendikleri konularla alakalı bilgiler paylaşmaktadır. Paylaşılan blog miktarından fazla insanda bu blogları takip etmektedir.

Blogların barındırdığı bilgi miktarı düşünüldüğünde, blog tabanlı arama robotlarının ortaya çıkışı da anlaşılmaktadır. Blog tabanlı arama robotları sadece blogları indekslemektedirler. Blogların oluşturuldukları yazılımlarla daha çok uyum sağlayan, bu arama robotları daha başarılı sonuçlar ortaya koymaktadırlar [22].

3.7.4 Alan adı tabanlı arama robotları

Arama robotları, sayfaları indekslemeye başladıklarında her hangi bir kısıtlamaya sahip olmadan elde ettikleri tüm adresleri ziyaret etmeye başlarlar. Bu adreslerde gelişi güzel şekilde arama motorunun veri tabanına kaydedilmektedir. Bunun yanında alan adı tabanlı arama robotları, aramaya alan adlarını temel alarak devam etmektedirler. Bu sayede, o alan adına ait alt yollar ve sayfalar bir ağaç şeklinde ortaya çıkmaktadır. Bu tür arama robotlarının faydaları arama sonuçlarının alan adı bazında alınabilmesi ve ilgi alanlarının daraltılabilmesini sağlamaktır.

3.7.5 Dil tabanlı arama robotları

İnternetin tüm dünya tarafından kullanıldığı göz önüne alınırsa, farklı dillerde içeriklerin üretildiği de fark edilmektedir. Günümüzde en yaygın dillerden biri olan İngilizce sayfalar çoğunlukta olsa da, insanların kendi dillerinde arama yapmak istemesi ve buna uygun sonuçların ortaya konabilmesi amacı ile dil tabanlı arama motorları ortaya çıkmıştır.

Günümüzdeki arama motorları, birden fazla dilde arama yapma imkânı vermektedir. Google arama motorunun 46 [23], Bing arama motorunun ise 40 [24] dilde arama yapmaya imkân verdiğini düşünürsek dil tabanlı aramanın önemi anlaşılmaktadır.

Dil tabanlı arama robotları, indeksledikleri sayfaların dillerini tespit etmek için sahip oldukları yöntemleri kullanmaktadırlar. Dili tespit edilen sayfalar arama motorunun veri tabanında belirleyici şekilde kaydedilirler. Bu sayede arama sonuçları da istenilen dilde gelebilmektedir.

3.7.6 RSS tabanlı arama robotları

RSS'ler (Really Simple Syndication) sitelere yeni eklenen içeriklerinin kullanıcılar tarafından kolay takip edilmesini sağlamak amacı ile ortaya çıkmışlardır. Rss hizmeti veren siteler içeriklerini otomatik olarak bir xml dosyasına yazmaktadırlar. Rss okuyucu yazılımlarla bu başlıklar öğrenilmekte ve içerik hızlı bir şekilde takip edilebilmektedir. Şekil 3.7'de bir RSS dosyası örneği görülmektedir.

```
<rss version="2.0">
<channel>
<title>Güncel Haberler</title>
<link>http://ornek.com/</link>
<description>Örnek Haber Merkezi</description>
<language>tr-tr</language>
<pubDate>Tue, 10 Jun 2003 04:00:00 GMT</pubDate>
<lastBuildDate>Tue, 10 Jun 2003 09:41:01 GMT</lastBuildDate>
<docs>http://ornek.com/rss</docs>
<generator>Weblog Editor 2.0</generator>
<managingEditor>editor@ornek.com</managingEditor>
<webMaster>webmaster@ornek.com</webMaster>
<item>
<title>Trafik Haberleri</title>
<link>
http://ornek.com/haber.asp
</link>
<description>
Örnek Haberler  <a href="http://ornek.com/ornek.htm">Trafik de Son Durum</a>.
</description>
<pubDate>Tue, 03 Jun 2003 09:39:21 GMT</pubDate>
<guid>
http://ornek.com/ornek.htm
</guid>
</item>
</channel>
</rss>
```

Şekil 3.7. Örnek RSS Dosyası

Güncellenen içeriğin bulunmasını sağlayan rssler için, rss tabanlı arama robotları üretilmiştir. Bu arama robotları xml dosyasını ayrıştırarak yeni gelen içeriklerin adreslerini bulmaktadırlar [25]. Bu sayede rss arama robotları tüm sayfaları gezerek içeriklerini ayrıştırma işlemi yapmadıklarından daha hızlı ve efektif çalışmaktadırlar. Ayrıca tekrar ziyaret politikalarına büyük destek sağlamaktadır.

3.7.7 Yerel arama robotları

Depolama alanlarının büyümesi ile beraber kişisel bilgisayarlardaki saklama alanları da büyümüştür. Artık sıradan ev kullanıcıların dahi terabyte seviyesinde saklama alanları vardır. Kullanıcılar bilgisayarlarındaki bu alanlarda yüz binlerce dosya saklamaktadırlar. Bu dosyaların arasından, aranılan dosyanın bulunması kullanıcı için sorun oluşturmaktadır. Bunun için yerel arama robotları ortaya çıkmıştır. Lokal arama robotları kullanıcın bilgisayarında tüm dosyaları içeriklerine ve konumlarına göre indekslemektedirler [26]. En çok bilinen yerel arama motorları "Windows Arama Dizini" [27] ve "Google Desktop Search"'tür [28].

3.8 Açık kaynak kodlu arama robotları

Arama motorlarının internetin kullanımdaki önemi düşünüldüğünde, arama motorlarına yapılan yatırımlarda anlaşılmaktadır. Arama motoru sahibi firmalar arama motorlarının daha iyi olması için her yıl büyük yatırımlar yapmaktadır. Aynı zamanda arama motoru firmaları için büyük bir gelir kaynağı da olmaktadır. Bu yüzden arama motorlarının kodları açık kaynak kodlu değildir. Yöntemleri ve teknolojileri rekabet açısından saklanmaktadır.

Bunun yanında bu yazılımlara alternatif olması açısından açık kaynak kodlu arama motorları da ortaya çıkmıştır. Bunlar ticari firmaların arama motorları gibi kapsamlı hizmet ve büyük kaynaklara sahip olmasalar da gelişmektedirler.

Açık kaynak kodlu arama robotları, gönüllü yazılımcı toplulukları tarafından geliştirilmektedir.

3.8.1 Heritrix arama robotu

Heritrix açık kaynak kodlu, java dilinde yazılmış arama robotu ve arama motorundan oluşmaktadır. Heritrix 2003 yılında geliştirilmeye başlanmış, 2004 yılında ilk resmi sürümünü duyurmuştur. Şuanda güncel sürümü ise 3.1.1'dir [29].

3.8.2 DataparkSearch arama robotu

DataparkSearch, Maxim Zakharov tarafından 2003 yılında geliştirilmiş ve C dilinde yazılmıştır. Yazılımcısı tarafından birçok özellik eklenmiştir. Arama robotu tarafından indekslenmiş sayfaların, içeriklerindeki tüm kelimeler üzerinden arama yapma imkânı vardır. Bunun yanında ofis dosyaları, resim, müzik ve farklı birçok dosya formatında içeriği indekslemektedir.

DataparkSearch farklı dillere sahip siteleri indeksleye bildiği gibi bunların sorgulanmasına da imkân vermektedir [30].

3.8.3 AspSeek arama robotu

ASPseek C++ dilinde yazılmış genel amaçlı kullanılan bir arama motorudur ve paralel çalışmayı desteklemektedir. Mantıksal aramayı destekleyen bir kullanıcı arayüzüne sahiptir. Bu arayüzden ve/veya, dâhil etme gibi mantıksal aramayı güçlendiren komutlar verilebilmektedir.

Arama robotu, Robots.txt standardını desteklemektedir. Dokümanlar ve medyalar gibi içerikleri de indeksleyebilmektedir. Nezaket politikalarına sahiptir. Böylece sunuculara aşırı yük getirmemektedir [31].

3.8.4 HTTrack arama robotu

HTTrack, Xavier Roche tarafından C ve C++ yazılım dillerinde geliştirilmiştir. Javascript ve gömülü java kodlarını da tespit edebilmektedir. Kullanıcı arayüzü birçok dili desteklemektedir.

Çoklu bağlantılara izin vermesinden dolayı bant genişliğini etkin kullanmaktadır. Aynı zamanda IPv6 protokolünü desteklemekte ve dosya tiplerine göre filtreleme seçenekleri bulunmaktadır. En son sürümü 3.45-4 olup 50000 satır civarında yazılım koduna sahiptir [32].

3.8.5 Nutch arama robotu

Nutch, Apache Software Foundation tarafından geliştirilmektedir. Yazılım dili Java'dır. Yazı tabanlı arama imkânı vermektedir. Paralel çalışma, dağıtık veritabanı kullanma imkânına sahiptir [33].

4. ARAMA ROBOTU İÇİN URL ATAMADA YENİ BİR YAKLAŞIM

Arama robotları ziyaret ettikleri sayfalardan topladıkları yeni adresleri, arama motorunun veri tabanındaki ziyaret edilecek adres listesine kaydetmektedirler. Bir sonraki ziyaret edilmesi gereken adresi URL atayıcı bu listeye bakarak dağıtıcıya vermektedir. Dağıtıcıda aldığı bu adresleri arama robotlarına iletmektedir. Arama robotları da ziyaret ettikleri sayfalardan topladıkları yeni adresleri ziyaret edilecek adres listesine kaydetmektedir. Bu işlemler listedeki tüm adresler ziyaret edilinceye kadar devamlı tekrarlanmaktadır.

Arama robotlarının ziyaret edecekleri adreslerin belirlenmesi arama motorları için büyük önem arz etmektedir. Bu sayede önemli sayfaları indekslerine alabilmektedirler. Arama motorları daha önce veritabanına ilk kaydedilen adresten başlayarak adres dağıtımı yapmaktaydı. Bunun dezavantajı eğer diğerlerinden daha önemli sayfalar, veri tabanına daha sonra kaydedilmişse sıranın ona gelmesi daha uzun süre alacaktır.

Bu sıradaki sorgularda, sayfa indekslenemediği için listelenemeyecektir. Bu sorunu çözmek için PageRank yöntemi gibi farklı yöntemler kullanılarak URL ataması da yapılmaktadır [37]. Arama motorları için URL atamasını iyileştirmek arama robotlarının daha önemli verileri arama motorunun indeksine kaydetmesini sağlayıp, sonuç listelerinin daha iyi olmasını sağlamaktadır.

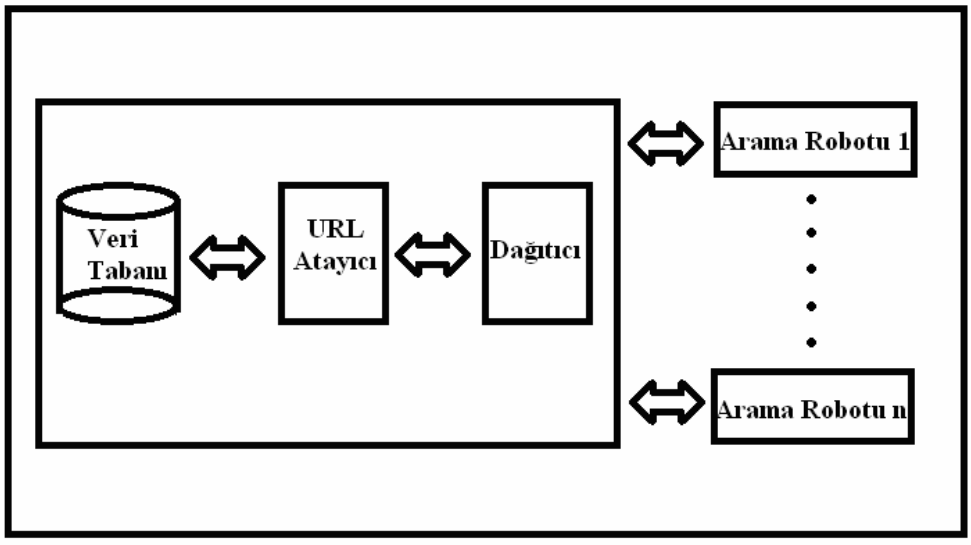
Bu tez çalışmasında, arama robotlarına URL atama yönteminde kullanılması için yeni bir yaklaşım sunulmuştur. Bir sonraki ziyaret edilecek adresin bulunması noktasında daha popüler olan adresin seçimine öncelik veren bir model geliştirilmiştir.

Bu sayede daha önemli sayfaların arama robotları tarafından ziyaret edilmesi için geçen süre kısaltılmaya çalışılmış; bunun yanında sadece önemli sayfaların ziyaret edilmesini önlemek için modele zaman faktörü de eklemiş böylece tüm sayfaların ziyaret edilmesi hedeflenmiştir.

URL ataması için kullanılacak yeni yöntemin sonuçlarının elde edilmesi için yeni bir arama robotu geliştirilmiştir. Geliştirilen arama robotunun, paralel çalışabilmesi sağlanmış böylece daha çok sayfayı ziyaret edip adres toplaması amaçlanmıştır. Temel amaç yöntemin çalışmasını görmek olduğu için kelime ayrıştırma işlemine yer verilmemiştir.

4.1 Sistem Mimarisi

Arama motorları birçok bileşenden oluşmaktadır. URL atama işlemi yapabilmek için ise arama motorunun veri tabanı, URL atayıcı, dağıtıcı ve arama robotları kullanılmaktadır. Bu bileşenler birbirleri ile iletişime geçip bilgi aktarmaktadırlar. Şekil 4.1' de sistem mimarisi yer almaktadır.



Şekil 4.1. URL Atama Mimarisi

Sistem mimarisinde yer alan bileşenler:

Veri Tabanı: Arama motorları, elde ettikleri bilgileri muhafaza edebilmek için veri tabanlarını kullanmaktadır. Elde edilen bilgi miktarı arttıkça, veri tabanında bilgileri saklamak ve ulaşmak zorlaşmaktadır. URL atayıcıya ihtiyacı olan adres listesi veri tabanı tarafından sağlanmaktadır.

URL Atayıcı: Veri tabanı tarafından sağlanan adreslere üzerindeki yöntemi kullanarak sıralama yapmaktadır. Öncelik sırasına dizdiği bu adresleri dağıtıcıya göndermektedir.

Dağıtıcı: URL atayıcıdan gelen adresleri arama robotlarına dağıtmaktadır. Arama robotu yeni bir sayfayı ziyaret etmek için dağıtıcıya istek göndermektedir. Dağıtıcıda bu gelen isteklere URL atayıcıdan aldığı adresler ile cevap vermekte, bu sayede arama robotları da sayfaları ziyaret etmektedir. Arama motorlarının birçok arama robotu kullandığı düşünüldüğünde, tüm isteklere en kısa zamanda cevap vermek dağıtıcının görevlerindendir.

Arama Robotu: Arama motorlarının web sayfalarını ziyaret etmek ve onları indekslemek için kullandığı yazılımlardır. Bu yazılımlar ziyaret ettikleri sayfaların içeriklerini ayrıştırarak içeriklerini arama motorunun veri tabanına gönderirler. Yine bu ayrıştırma işlemi esnasında elde ettikleri yeni adresleri de arama motorunun veritabanındaki ziyaret edilecek adres tablosuna kaydetmektedirler.

Arama robotunun bir sayfayı ziyaret edebilmesi için sayfanın adresini bilmesi gerekmektedir. Arama robotu adres isteklerini dağıtıcıya bildirmektedir. Dağıtıcı arama robotundan gelen bu istekleri cevaplamak için arama motorunun URL atayıcısından adres listesini talep etmektedir.

URL atayıcı, dağıtıcıdan gelen isteği cevaplayabilmek için arama motorunun veri tabanından ziyaret edilecek adres tablosuna ulaşır ve oradaki bilgileri kullanarak sahip olduğu yöntemi kullanarak adres listesini önceliklendirmektedir

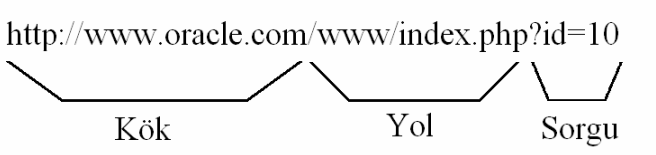
Önceliklendirilmiş bu listeden en yüksek puan alan adres en önce olmak üzere, dağıtıcıdan gelen talebi karşılamak için yeniden sıralanmış listeye göre ziyaret edilmesi gereken sayfanın adresi yollanmaktadır.

Dağıtıcı, URL atayıcıdan gelen adres bilgisini talebi yapan arama robotuna gönderir ve bir sonraki arama robotunun talebi karşılamak üzere devam eder. Arama motoru da gelen adresi ziyaret edip içeriğini ayrıştırdıktan sonra bulduğu yeni adresleri veri tabanının ziyaret edilecek adres listesine kaydetmektedir. Bu işlemler ziyaret edilmesi gereken sayfalar bitinceye kadar devam eder.

4.2 URL Atama İşlemi için Yeni Bir Yaklaşım

Bu tez çalışmasında, URL atama işleminde kullanılmak üzere yeni bir yöntem ortaya konulmaktadır. Bu yöntemde ziyaret edilen adresler (domain ve alt domain) ve adresin veritabanına kayıt tarihi bilgileri kullanılmaktadır.

Veri tabanına kayıt edilen adresler için standart olarak kullanılan 69 farklı URI (Uniform Resource Identifier) tipinden http ve https kullanılmıştır [34]. Veri tabanındaki ziyaret edilmesi planlanan tüm adresler için, üç bölüme ayırma işlemi uygulanmıştır. Bu bölümler adresin alan adını ifade eden kök kısmı, adresin yol kısmı ve sorgu kısmıdır. Bölümlere ayrılmış bu adreslerin sayısal olarak ifade edilmesi gerekmektedir. Bu nedenle hash fonksiyonu kullanılmakta ve karakterler sayısal olarak ifade edilmektedir. Bu sayede adresler üç boyutlu uzayda ifade edilebilir hale getirilmiştir.



Şekil 4.2. URL Bölümleme

Şekil 4.2'de URL bölümleme işlemi görülmektedir. Adresin kök kısmı için "*http://www.oracle.com*", yol kısmı için *"/www/index.php?"* ve sorgu kısmı için "id=10" seçilmiştir. Bu sayede adresler üç parçaya ayrılmıştır. Ayrılan bu parçalara kök kısmı için x, yol kısmı için y ve sorgu kısmı için z ifadeleri atanmıştır.

x = Adresin Kök Kısmı

y = Adresin Yol Kısmı

z = Adresin Sorgu Kısmı

Elde edilen x, y ve z değerlerinin sayısal değerlere dönüştürülmesi için; sistem mimarisinde de ifade edilmiş olan veritabanının Hash fonksiyonu kullanılmaktadır. Bu sayede veri tabanındaki tüm adres parçaları sayısal değerlere dönüştürülmüştür. Sayısal değerlere dönüştürülmüş adres parçaları ise X, Y ve Z olarak ifade edilmiştir.

X = Kök Sayısal Değer

Y = Yol Sayısal Değer

Z = Sorgu Sayısal Değer

Tablo 4.1 Adreslerin Bölünmesi ve Sayısal Değerlere Dönüştürülmesi

Adres Parçaları	**Karakterler**	**Dönüştürülen Sayısal Değerler**
X – (Kök)	x0=http://www.oracle.com	X0=3904851578
Y – (Yol)	y0=/www/index.php?	Y0=1547933509
Z – (Sorgu)	z0=id=10	Z0=33786873

Tablo 4.1’ in son sütununda görülen değerler, veritabanında kayıtlı olan ziyaret edilecek adreslere Hash fonksiyonunun uygulanması neticesinde elde edilen değerlerdir.

Veritabanında kayıtlı olan ziyaret edilmemiş adreslerin toplam sayısı hesaplanmaktadır. Hesaplanan bu değer K ile ifade edilmekte olup Hash fonksiyonu ile edilmiş olan sayısal değerlerle beraber önerilen yöntem ile beraber kullanılacaktır.

K = Toplam Kayıt Sayısı

URL atamada öncelikli kullanılacak olan popüler adreslerin belirlenmesi için önerilen modelde popüler adresler üst sıralara çıkarılmaya çalışılmaktadır. Bununla beraber popüler olmayan diğer adreslerinde arama robotu tarafından ziyaret edilebilmesi amacı ile modele bir zaman puanı da dâhil edilmekte, bu sayede tüm adreslerin ziyaret edilmesi amaçlanmaktadır. Zaman puanının oluşturulabilmesi için ziyaret edilen adresin veritabanına kayıt tarihi kullanılmış ve modelde T_y olarak ifade edilmektedir.

T_y = Kayıtın Yaratılma Tarihi

X, Y ve Z değerlerinin toplam kayıtlar içerisindeki ağırlıklarının bulunması için; her biri kendi içinde toplam kayıt sayısına bölünmüş bu sayede X, Y ve Z değerlerinin K içerisindeki ağırlıkları bulunmuştur.

Bulunan ağırlıklar X_p, Y_p ve Z_p olarak Denklem 4.1, Denklem 4.2 ve Denklem 4.3’de ifade edilmektedir.

$$X_p = (X_g \times 100) / K \qquad \text{Denklem 4.1}$$

$$Y_p = (Y_g \times 100) / K \qquad \text{Denklem 4.2}$$

$$Z_p = (Z_g \times 100) / K$$ Denklem 4.3

Adreslere ait zaman puanını hesaplamak için; adresin veritabanına kayıt tarihi, veritabanı içerisindeki en büyük kayıt tarihinden çıkartılıp yüzle çarpılmış ve en büyük ile en küçük tarih arasındaki farka bölünmüştür. Bu sayede veri tabanına daha önce yazılmış kaydın daha yüksek bir puan alması sağlanmış olup zaman puanı da T_p olarak adlandırılmıştır. Zaman puanı Denklem 4.4 ile gösterilmektedir.

$$T_p = ((Max(T_Y) - T_Y) \times 100) / (Max(T_Y) - Min(T_Y))$$ Denklem 4.4

T_p = Zaman Puanı

Elde edilen X_p, Y_p, Z_p ve T_p değerleri önerilen yöntem içerisinde ağırlıklandırılmaktadır. Bu ağırlık değerleri P_x, P_y, P_z ve P_t ile ifade edilmektedir. Ağırlıkların belirlenmesinde elde edilmiş değerler ile birçok denemeler yapılmış ve daha az popüler olan sayfalarında arama motoru tarafından ziyaretinin sağlanması amacıyla zaman puanı için kullanılan ağırlık değeri arttırılmıştır.

P_x = X_p'nin P içerisindeki Ağırlığı

P_y = Y_p'nin P içerisindeki Ağırlığı

P_z = Z_p'nin P içerisindeki Ağırlığı

P_t = T_p'nin P içerisindeki Ağırlığı

P_x, P_y, P_z ve P_t' ye ait ağırlıklar aşağıda ifade edilmektedir.

P_x = %40
P_y = %25
P_z = %15
P_t = %20

Bir adresin popülerlik puanını hesaplamak için aşağıdaki Denklem 4.7 kullanılmaktadır.

P = Popülerlik Puanı

$$P = \frac{\left(X_g \times 100\right)}{K} \times P_x + \frac{\left(Y_g \times 100\right)}{K} \times P_y + \frac{\left(Z_g \times 100\right)}{K} \times P_z + \frac{((Max(T_Y) - T_Y) \times 100)}{Max\left(T_y\right) - Min\left(T_y\right)} \times P_t$$

$$P = \frac{100}{K}\left(X_g \times P_x + Y_g \times P_y + Z_g \times P_z\right) + \frac{((Max(T_Y) - T_Y) \times 100)}{Max\left(T_y\right) - Min\left(T_y\right)} \times P_t$$

$$P = 100 \times \left(\frac{1}{K}\left(X_g \times P_x + Y_g \times P_y + Z_g \times P_z\right) + \frac{(Max(T_Y) - T_Y)}{Max\left(T_y\right) - Min\left(T_y\right)} \times P_t \right) \quad \text{Denklem 4.7}$$

Veritabanındaki tüm adresler için P popülerlik değeri hesaplanmakta ve en yüksek puanı alan adres en öncelikli olacak şekilde sıralama yapılmaktadır.

4.3 Uygulama Ortamı

Önerilen yöntemin uygulamasını gerçekleştirmek için aşağıdaki bileşenler kullanılmaktadır.

- Veri tabanı
- Arama Robotu

Arama robotları temel olarak aynı görevleri yerine getirseler de aralarında farklılıklar bulunabilmektedir. Önerilen URL atama yöntemini test edebilmek için yeni bir arama robotu geliştirilmiştir. Geliştirilen arama robotu paralel çalışmaya uygun olarak tasarlanmıştır.

4.3.1 Kullanılan yazılım ortamı

Önerilen yöntemde veritabanı olarak Oracle XE 11g kullanılmaktadır. Veritabanına ait olan bazı fonksiyonlarda yer almaktadır. Veritabanının mantıksal tasarımında Embarcadero ER Studio 6.6 programı kullanılmıştır. Yazılım dili olarak Java 1.7 kullanılmıştır. Java programlama dilinin açık kaynak kodlu olması, içerdiği geniş kütüphaneler ve platform bağımsız çalışabilmesi bu seçimde etkili olmuştur. Java geliştirme ortamı olarak da NetBeans IDE 7.1 kullanılmaktadır. İşletim sistemi olarak Windows 7 Enterprise kullanılmaktadır. Elde edilen verilerin grafiksel gösterimlerinin oluşturulması için SPSS 16 programı kullanılmıştır.

Arama robotu adresleri ziyaret edip, bu sayfalara ait bilgiler veritabanına kaydedilmeye başlandıkça uygulama sunucusunun kaynaklarını daha etkin kullanabilmek için veri tabanı tasarımında gereksiz alanların kullanılmamasına dikkat edilmiş ve bu sayede veritabanının getireceği depolama alanı sorununun da ortadan kaldırılmasına çalışılmıştır.

Microsoft Windows 7 Enterprise

Windows 7 Enterprise kurum içinde kullanıcıların değişen ihtiyaçlarını karşılamak için tasarlanmış iş bilgisayarları ve bilişim teknolojisi uzmanları için gelişmiş Windows işletim sistemlerimden biridir.

Oracle 11g Express

Oracle 11g XE bir veritabanı yönetim sistemidir. Yeni özelliklerle geliştiricilere, veritabanı idarecilerine ve kullanıcılara verilerin depolanması, işlenmesi ve çekilmesi konularında kapsamlı bir denetim sunmaktadır. XE sürümü ile eğitim maksatlı uygulama geliştiricilere ücretsiz kullanım hakkı sunmuştur [35].

NetBeans 7.1

NetBeans 7.1 geliştirme ortamı ile Java uygulamaları geliştirilebilinmektedir. Gerek kod yazımı gerek ise görsel olarak sürekle bırak ile tasarım yapmayı oldukça kolay kılmakta ve uygulama geliştirici için Java kütüphanelerini kullanmayı desteklemektedir.

Embarcadero ER Studio 6.6

Embarcadero ER Studio 6.6, veritabanı modelleme için gerekli işlevleri sağlayan bir yazılımdır.

Embarcadero ER Studio 6.6 programı ile yeni bir mantıksal veritabanı tasarımı yapılabilmekte ya da var olan fiziksel veritabanından geriye mühendislik yöntemi ile programın için mantıksal veritabanı modeli olarak aktarılabilinmektedir. Model üstünde yapılan değişiklikler aynı zamanda veritabanına uygulanabilinmektedir. Modeller arasında karşılaştırma yapabilmekte, istenen modeller birleştirilebilmektedir [38].

Java

Java, uzman bir topluluk tarafından sınanmış, incelenmiş, geliştirilmiş ve onaylanmıştır. Bugün, 9 milyonu aşkın yazılım geliştiricinin katkılarıyla dünyanın en yaygın ve en etkin programlama dilidir. Esneklik, verimlilik, taşınabilirlik özellikleri ve sunduğu olanaklarla yazılım geliştiriciler için vazgeçilmez bir araç olmuştur [36].

- Platform bağımsız uygulama geliştirme imkânı sağlamaktadır.
- Nesne tabanlı yapısı sayesinde diğer uygulama kodlarından faydalanma imkânı olmaktadır.
- Geniş kütüphane desteği sayesinde hazır metotlardan faydalanma imkânı sağlamaktadır.

SPSS 16

SPSS 16 istatistiksel analize yönelik bir bilgisayar programıdır. Dışardan veri girişine imkân verdiği gibi kendi editöründen de veri girişi yapmak mümkündür. İçerisinde istatistiksel analizler yapmaya imkân veren hazır formüller bulunmaktadır. Hesaplama sonuçları analiz edilebilinmekte ve istenirse grafik olarak sonuç üretmektedir.

4.3.2 Donanım ortamı

Arama motorlarının ihtiyaç duyduğu sistem gereksinimleri yüksektir. Kullanılan işlemci gücü ve bant genişliği gereksinimleri, indekslenen site sayısı arttıkça büyümektedir. Artan bu gereksinimlerin karşılanabilinmesi için arama motorları çok sayıda sunucu kullanmaktadır.

Arama robotları sayfaları ziyaret etmek için internet bağlantısına ihtiyaç duymaktadır. Bu yüzden 4 Mbit'lik internet bağlantısı sağlanmıştır. Arama robotlarının tüm internet bant genişliğini kullanması sağlanmış bu sayede daha verimli çalışması planlanmıştır.

Uygulamanın çalışacağı bilgisayara ait sistem özellikleri aşağıdadır,

- Intel® Core™2 Quad Processor Q8300 (4M Cache, 2.50 GHz, 1333 MHz FSB) işlemci
- 4 Gb DDR 2 800 MHz Bellek
- 60 Gb SSD Harddisk
- 4 Mbit İnternet Bağlantısı

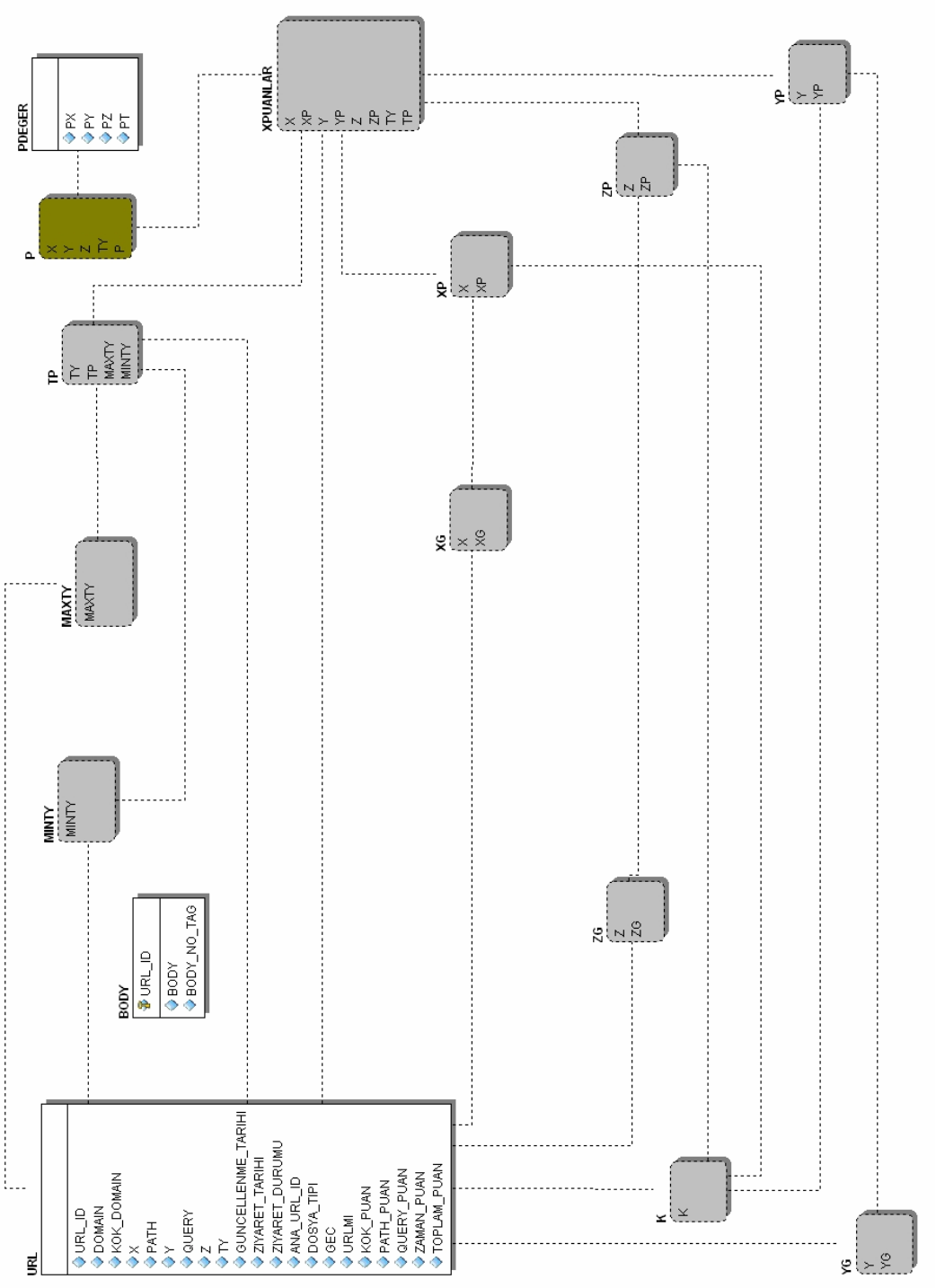
4.3.3 Veritabanı tasarımı

Veritabanı tasarımında mantıksal modellerin kullanımı için Embarcadero ER Studio 6.6 programı kullanılmıştır. Veritabanının mantıksal tasarımı gerçekleştirilirken ER (Entity Relationship- Varlık İlişki) olarak adlandırılan veritabanı modelinden faydalanılmaktadır. Mantıksal veritabanı tasarlandıktan sonra Oracle 11g Xe veritabanı yönetim sistemine aktarılmıştır.

Veritabanında body, pdeger ve URL olmak üzere 3 tablo; k, maxty, minty, p, tp, xg, xp, xpuanlar, yg, yp, zg, zp olmak üzere 12 bakış (view) bulunmaktadır. Şekil 4.2'de veri tabanı ve tablolar arasındaki ilişki gösterilmektedir.

Veri tabanı tasarımında bakışlardan faydalanılmıştır. Bakışlar bir ya da birkaç tablodan veri alan sorgulardan oluşmaktadır. Tablo gibi gözükmekte; tablolar gibi satır ve sütunlara sahiptirler. Tablolardan farklı olarak veri saklayamamakta ve asıl veri kaynağı olan tablolardaki değişimlerden etkilenmektedirler.

Uygulama geliştirme sürecinde veritabanı fonksiyonlarından en fazla şekilde faydalanılmaya çalışılmış, bu sayede arama robotu ve veritabanı arasındaki iletişim süresi kısaltılmaya çalışılmıştır.



Şekil 4.2. Veri Tabanı ve Tablolar Arasındaki İlişikler

4.3.4 Yazılım geliştirme süreci

URL atama işlemi için geliştirilen yeni modeli uygulayabilmesi amacı ile arama robotu yazılımı geliştirilmesi gerekmektedir. Java dilinde geliştirilen uygulama için Javanın kendi kütüphanelerinden de faydalanılmıştır.

Uygulamaya aşağıdaki kabuller ile başlanmıştır;

- Başlangıç Sayfası olarak www.oracle.com seçilmiştir.
- Kayıt edilen ilk 600.000 adres için bu yöntem çalıştırılmıştır.
- URI tiplerinden http ve https seçilmiştir.
- Ziyaret edilen sayfalar arama motorunun indeksi için kelime bazında ayrıştırılmamıştır.
- En yüksek popülerlik puanına sahip 10 adres seçilmiştir.
- İçerik tipi olarak “text/html” belirlenmiştir. Diğer içerik tipleri dikkate alınmamıştır.
- Yol veya sorgusu olmayan adresler için ilgili alanlara “-“ değeri atanmıştır.

Uygulamada ilk olarak adres bilgilerinin tutulduğu değişkenin tanımlanması gerekmektedir. Java nesnel tabanlı bir yazılım geliştirme ortamı olduğu için adres değişkeni, bir sınıf nesnesi olarak tanımlanmıştır. Adres sınıfı tanımlanırken Javanın kendi kütüphanelerinden Java.net.URL sınıfından katılım yolu ile yeni bir adres sınıfı türetilmiştir. Bu sayede Java.net.URL sınıfının sahip olduğu tüm değişkenler ve metotlar yeni tanımlanan sınıf için kullanılabilir duruma gelmiştir. Java.net.URL sınıfının içinde olmayan uygulamaya özel değişken ve metotlar ayrıca tanımlanmıştır. Tanımlanan bu yeni adres sınıfı uygulama için esas teşkil eden sınıftır. Uygulamanın ihtiyacı olan diğer sınıflarda ayrıca tanımlanmıştır. Şekil 4.3’ de adres sınıfının yapısı gözükmektedir.

```
import java.net.URL;
import veritabani.islemler.*;

public class Adres {

    public URL url;
    private String domain;
    private Integer ziyaret_durumu;
    private Integer url_id;
    private Integer ana_url_id;
    private String icerikTipi;
    private Integer urlmi;
```

Şekil 4.3. Adres Sınıfının Tanımlanması

Elde edilen bilgilerin saklanabilmesi için veritabanında saklanması gerekmektedir. Veritabanına bağlantı sağlandıktan sonra işlem sonuna kadar kesilmemektedir. Bu sayede veritabanı ile bağlantı kurma işleminde oluşan yüksek işlemci ve bant genişliği yükü azaltılmak istenmiştir. Uygulamanın veritabanı bağlantısını yapabilmek için Şekil 4.4' deki bağlantı yapısı kullanılmıştır.

```
Class.forName("oracle.jdbc.driver.OracleDriver");
        con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:cra
wlerdb", "crawler2", "123456");
```

Şekil 4.4. Veri Tabanı Bağlantı Cümleciği

Veritabanı bağlantısı sağlandıktan sonra uygulama ziyaret etmek için yeni bir adres talep etmektedir. Veri tabanından gelen yeni adres bilgisini kullanarak sayfanın içeriği kontrol etmektedir. Eğer sayfa hatalı veya ulaşılamaz durumda ise veritabanından gönderilen adresin hatalı olduğu bilgisini yollamaktadır. Şekil 4.5' de adresin hatalı olması durumunda veri tabanına yapılan güncelleme gözükmektedir.

```
sql = "update CRAWLER2.url set ziyaret_tarihi=sysdate ,
ziyaret_durumu='2' ,urlmi=0 where url_id=" + adres.getUrl_id();
```

Şekil 4.5. Adresin Hatalı Olması Durumu

Eğer ziyaret edilen adres hatalı değilse içerik tipinin kontrol edilmesi gerekmektedir. Şekil 4.6' da İçerik Tipinin Belirlenmesi işlemi gösterilmektedir. Veritabanı tarafından sağlanan adres resim, film gibi medya dosyaları olabilmektedir. Bu tür sayfaların ayrıştırılmasını önlemek için sayfa ayrıştırma işlemine geçilmeden sayfanın içerik tipi belirlenmektedir.

```
URLConnection hpCon = adres.url.openConnection();
   if(hpCon.getContentType()==null)
   {
   adres.setIcerikTipi("");
   }
   else
   {
   adres.setIcerikTipi(hpCon.getContentType());
   }
```

Şekil 4.6. İçerik Tipinin Kaydedilmesi

İçerik tipi ön tanımlı koşullardaki ile uyumlu ise sayfanın içeriği, uygulamanın çalıştığı bilgisayara indirilmektedir. İçerik tipi belirlendikten sonra adres sınıfına Şekil 4.7' de gözüktüğü gibi kaydedilmektedir.

```
adres.setUrl(new URL(adres.getDomain()));
        adres.setUrlmi(1);
        adres=icerikTipiBelirleme(adres);
```

Şekil 4.7. İçerik Tipinin Belirlenmesi

Sayfanın içeriğinin elde edilmesi işleminde, Javanın Java.io.InputStreamReader sınıfı kullanılmıştır.

```
String html = "";
      Body body=new Body();
      try {
         try (BufferedReader in = new BufferedReader(
                  new InputStreamReader(adres.url.openStream())))
{
            String inputLine;
            while ((inputLine = in.readLine()) != null) {
               html += inputLine;
            }
         }
         body.setUrl_id(adres.getUrl_id());
         body.setBody(html);
         body.setBody_no_tag(parserBody.tagTemizle(html));
```

Şekil 4.8. Sayfa İçeriğinin Alınması

Şekil 4.8' de sayfa içeriğinin alınması işlemi gösterilmektedir. Sayfaya ait içerik BufferedReader nesnesi ile html değişkenine kaydedilmektedir.

Döngü içerisinde sayfanın içeriği okunmakta, döngü ise ancak sayfa sonu geldiğinde sonlanmaktadır. Döngünün bitimi ile sayfaya ait içerik elde edilmiş olduktan sonra içerik olduğu gibi ve içerisindeki etiketler temizlenmiş olarak iki farklı şekilde kaydedilmektedir.

Etiketlerin temizlenmesi esnasında "<a href>" etiketleri oldukları gibi bırakılmaktadır. Şekil 4.9' da html etiketlerinin temizlenmesi gözükmektedir. Sayfaya ait içeriğin tümündeki etiketler kaldırılmakta ve sadece sayfanın içerdiği metin bilgisine ulaşılmaktadır.

```
String url = "";
        String url2 = "";
        String tag[] = {"!--",  "big", "body", "b", "br", "center", "dd",
                        "dl", "dt", "embed", "em", "font", "form", "h1",
                        "h2", "h3", "h4", "h5", "h6", "head", "hr",
                        "html", "img", "input", "i", "link", "li",
                        "marquee", "menu", "meta", "ol", "option", "p",
                        "small", "strike", "strong","table", "td", "th",
                        "title", "tr", "tt", "ul", "u"};
        for (int i = 0; i < tag.length; i++) {
           while (body.indexOf("<" + tag[i]) > -1) {
              url = body.substring(0, body.indexOf("<" + tag[i]));
              body = body.substring(body.indexOf("<" + tag[i]) +
tag[i].length(), body.length());
              url2 = body.substring(body.indexOf("/" + tag[i] + ">") +
tag[i].length() + 2, body.length());
              body = url + " " + url2;
```

Şekil 4.9. İçeriğin Etiketlerden Temizlenmesi

Sayfanın içeriği elde edildikten sonra arama robotunun daha sonra ziyaret edebileceği yeni adreslerin bulunması için sayfanın içeriğinde ayrıştırma işlemi yapılması gerekmektedir.

Şekil 4.10'da yeni adreslerin bulunması için ayrıştırma işleminin detayları gösterilmektedir. İlk olarak sayfaya ait tüm içerik büyük harfe dönüştürülmektedir. Sayfadaki "HREF" etiketleri aranmaktadır. "HREF" etiketi bulunduktan sonra etikete ait olan adresin bulunabilmesi için adres alanı aranmaktadır.

Bulunan adrese ait "http://" başlığı yoksa eklenmektedir. Bu sayede sayfalara ait standart URI tipleri belirlenerek sayfanın kontrolü sırasında çıkabilecek olan hatalardan uzaklaşmak istenmektedir.

```
String url = "";
      Integer i = 0;
      body = body.replaceAll("\'", "\"");
      while (body.toUpperCase().indexOf("HREF=") > -1) {
         Adres tempAdres = new Adres();
         tempAdres.setAna_url_id(adres.getUrl_id());
         body =
body.substring(body.toUpperCase().indexOf("HREF=") + 5,
               body.length());
         if(body.indexOf("")==0){
            body=body.substring(1,body.length());
           url=body.substring(0, body.toUpperCase().indexOf(""));
         }

         else url=body.substring(0, body.toUpperCase().indexOf('>'));
         if (url.indexOf('/') == 0) {
            url=adres.url.getHost() + url;
         }
         if(url.length()>12)
         if(!url.toUpperCase().substring(0, 4).equals("HTTP"))
            url="http://"+url;
         body = body.substring(body.toUpperCase().indexOf(""),
body.length());
         if (url.length() > 12 && !url.isEmpty()) {
            try {
               tempAdres.setDomain(url);
               tempAdres= parserAdres.urlParser(tempAdres);
               if(tempAdres.getUrlmi()>0)
               {

                veritabani.islemler.urlYaz(tempAdres, con);
         i++;
               }
               else veritabani.islemler.hataliUrlYaz(tempAdres, con);
               url="";
```

Şekil 4.10. Yeni Adreslerin Bulunması

Yeni adreslerin bulunması arama robotları için vazgeçilmez görevlerden biridir. Bunun için sayfalarda yeni adres ayrıştırma işleminin yerinde çalışması gerekmektedir. Arama robotu için tek yeni adres bulma yöntemi bu değildir. Site

yöneticileri tarafından arama motorlarına yardımcı olmak için kullanılan Sitemap.xml dosyası da bulunmaktadır.

Sitemap.xml dosyası içerisinde o siteye ait tüm adreslerin listesi bulunmaktadır. Bu sayede arama robotu tüm sayfaları ayrıştırmak yerine yeni adres listesini Sitemap.xml dosyasından çekebilmektedir. Güncel arama motorları Sitemap.xml dosyasının kullanımını önermektedir. Fakat Sitemap.xml dosyasının kullanımında arama motorları için dezavantaj olabilecek durumlarda oluşmaktadır.

Eğer site yöneticisi, sitesindeki dosyaları Sitemap.xml dosyasına koymayı unutursa yada koymak istemezse, sayfa arama robotu tarafından ziyaret edilemeyecektir. Bu yüzden uygulamamızda hem sayfa ayrıştırma yöntemi ile hem de Sitemap.xml dosyasından yeni adres bulunması sağlanmıştır. Şekil 4.11'de Sitemap.xml dosyasının ayrıştırma işlemi gösterilmiştir.

```
URL oracle = new URL(url);
        URLConnection hpCon = oracle.openConnection();
        hpCon.setConnectTimeout(10000);
        hpCon.setReadTimeout(10000);
        if (hpCon != null) {
           if (hpCon.getContentType().indexOf("xml") > -1) {
              BufferedReader in = new BufferedReader(
                    new InputStreamReader(oracle.openStream()));
              String inputLine;
              while ((inputLine = in.readLine()) != null) {
                 html += inputLine;
                 System.out.println(html);
              }
              in.close();
```

Şekil 4.11. Sitemap.xml Dosyasının Ayrıştırılması

Bulunan yeni adreslerin veritabanına eklenmesi gerekmektedir. Yeni adreslerin ayrıştırma işlemleri sırasında veritabanına kayıt işlemleri de yapılmaktadır. Şekil 4.12'de veritabanına kayıt cümleciği gözükmektedir.

```
sql = "INSERT INTO CRAWLER2.URL (    DOMAIN,
KOK_DOMAIN,KOK_DOMAIN_HASH, PATH,
PATH_HASH,QUERY, "
                    + "QUERY_HASH,ANA_URL_ID,ICERIK_TIPI) "
                    + "VALUES ( '" + adres.getDomain() + "', '" + a + "', "
                    + a.hashCode() + ", '" + b + "', "
                    + b.hashCode() + ", '" + c + "',"
                    + c.hashCode() + " , " + adres.getAna_url_id() + ",'"
                    + adres.getIcerikTipi() + "')";
```

Şekil 4.12. Yeni Adresin Kayıt Edilmesi

Uygulamanın amacı, URL atama yönteminin gösterimi olsa da daha verimli ve hızlı çalışabilmesi için Java'ya ait tread sınıfı kullanılmıştır. Bu sınıf sayesinde uygulama aynı anda birden fazla kez çalışma imkânına sahip olmaktadır. Paralel çalışabilen iş parçacıkları sayesinde eğer bir sayfada işlem uzun sürse de, ayrıştırıcı diğer işlem parçacığına ait sayfa içeriğini ayrıştırabilir durumda olmaktadır. Bu da uygulamanın daha verimli çalışmasını sağlamaktadır.

```
public Crawler2() {
      t = new Thread(this, "Data Thread");
  }
```

Şekil 4.13. Çoklu İş Parçacıklarının Kullanımı

Uygulamanın yeni adres alma kısmında geliştirilen URL atama yöntemi veritabanının içinde gerçeklenmiştir. Veritabanının içerisinde gerçeklenmesinin

amacı URL atama işlemi için istenen bilgi miktarının büyüklüğünden dolayı veri transferi sırasında yaşanacak zaman kaybını önlemektir. Yöntemin adımları bakışlar ile gerçekleştirilmiştir. Bakışlar kendilerinde bilgi tutmayıp bağlı oldukları tablolardan bilgiyi çektiklerinden dolayı her bir URL atama işlemi için tüm hesaplamaları yeniden yapmak gerekmektedir.

Her URL ataması için bu işlemleri yapmak sunucu bilgisayar üzerine büyük bir yük getireceği gibi aynı zamanda arama robotlarının bekleme sürelerini de uzatmaktadır. Bu sorunun çözümü için ise veri tabanın sunduğu hizmetlerden biri olan sabitlenmiş bakış (materialized view) fonksiyonundan yararlanılmıştır. Sabitlenmiş bakış, bakışların veri tabanında tablo olarak saklanması ve belirlenmiş aralıklarla güncellenebilmesidir. Tablo olarak kaydedildiğinden dolayı verilere erişim hızlı olmaktadır. Ancak tekrar güncelleninceye, kadar eski bilgilere göre veri saklamaktadır. Uygulamamızda popülerlik değerlerinin hesaplanması, sabitlenmiş bakış olarak kullanılmış olup, iki saatte bir kere olmak üzere güncellenmeye tabi tutulmuştur.

Uygulama kullanıcı arabirimine sahiptir. Sistem bilgisi sekmesinde sunucu bilgisayarın durumu hakkında bilgi verilmektedir. Sunucunun durumu “Aktif” veya “Pasif” olarak değişmektedir. Aynı ekranda arama motorunun kayıt ettiği toplam adres sayısı gözükmektedir. Gözüken bu bilgilerin güncel durumlarını öğrenmek için sağ alta bulunan güncelle düğmesi tıklanmaktadır. Uygulamanın kullanıcı arabirimine ait ekran görüntüleri Şekil 4.14’ de gösterilmiştir.

Şekil 4.14. Sistem Bilgisi Ekranı

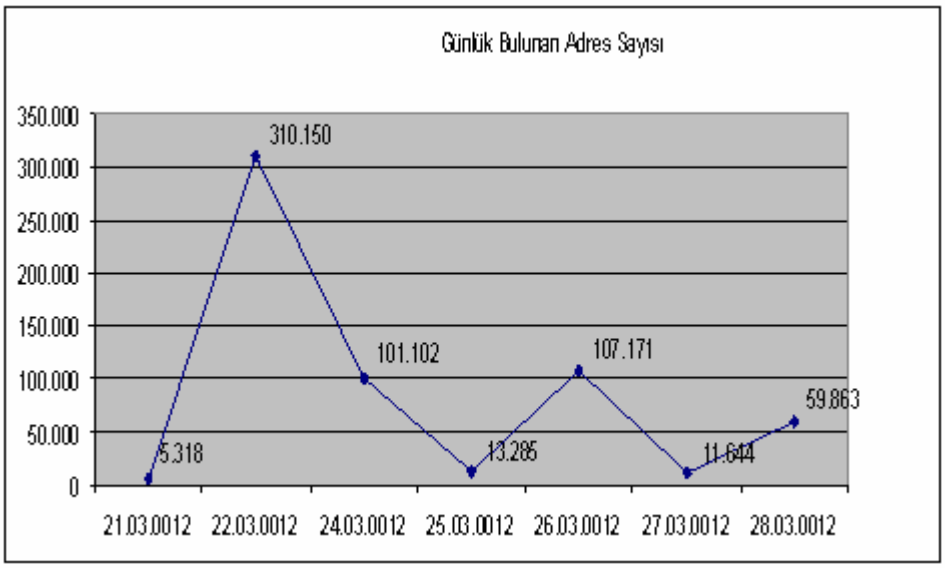
Şekil 4.15. İşlem Kaydı Ekranı

Şekil 4.15'de İşlem Kaydı Ekranı gözükmektedir. Bu ekranda kullanıcıya yapılan işlemler hakkında bilgi verilmiştir. Yeni bir arama robotu çalıştırıldığında, arama

robotunun ziyaret ettiği sayfadan topladığı yeni adres sayısı ve uygulamaya ait hata mesajları işlem kaydı ekranında gözükmektedir.

4.4 Uygulama Sonuçları

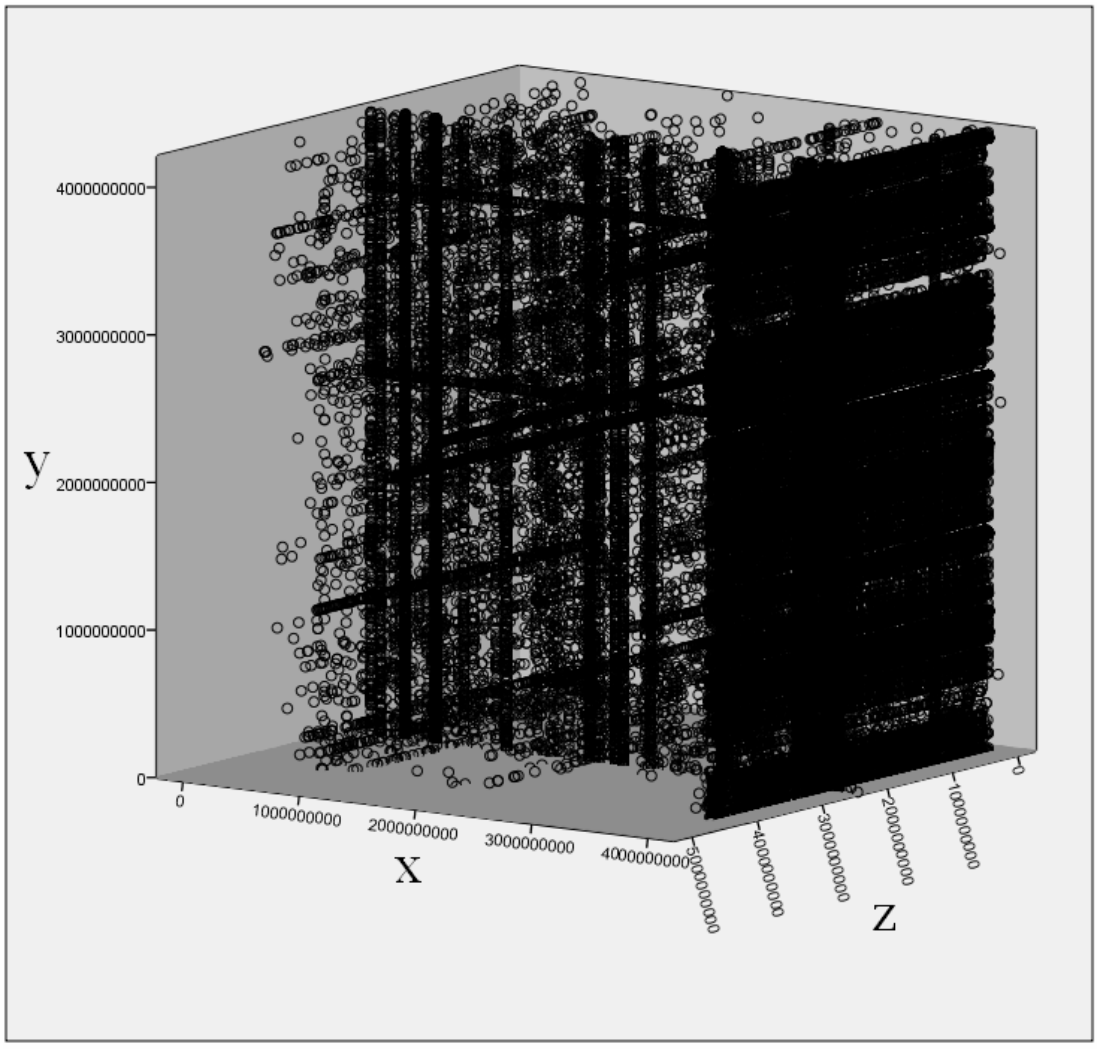
Geliştirilen arama robotu adres toplama görevi için çalıştırılmaya başlanmıştır. Arama robotunun adresleri kaydetme işlemine sistem kaynakları ölçüsünde devam edilmiştir. Öngörülen adres sayısına erişilince çalışması durdurulmuştur. Bir haftalık süreç içerisinde aralıklı olarak çalıştırılması ile 600.000 adet tekrar etmeyen adres elde edilmiştir. Şekil 4.16' da yeni adreslerin günlere göre kaydedilme grafikleri gözükmektedir.



Şekil 4.16. Arama Robotunun Günlük Yeni Adres Kayıt Sayıları

Veritabanına kayıt edilen adreslerin yöntemimize göre parçalanma işlemi yapılmıştır. Adresler kök, yol ve sorgu olarak üç parçaya ayrılmış, bulunan metinler sayısal değerlere çevrilmiştir. Çevrilen bu sayısal değerlerin x, y ve z olarak ifade edilip, üç boyutlu uzayda ifade edilmesi için koordinatlar şeklinde kullanılmıştır. Şekil 4.17'de

parçalanan bu adreslerin belirli yerlerde kümelenmeye başladığı gözlenmiştir. Kümelenen bu adres gruplarının diğer adreslerden daha yoğun olduğu ve öncelikli olarak ziyaret edilmesine karar verilmiştir.



Şekil 4.17. Adreslerin Üç Boyutlu Uzayda Gösterimi

Elde edilen adres listesi ile URL atama yönteminde kullanılmak üzere yeni model üzerinden hesaplamalar yapılmıştır. Yapılan hesaplamalar sonucunda elde edilen verilere göre oluşturulan popülerlik puanları adreslerin bulunduğu tabloda güncellenmiştir. Güncelleme sonrası adreslerin popülerlik puanlarına göre sıralaması Tablo 4.2’ de gösterilmiştir.

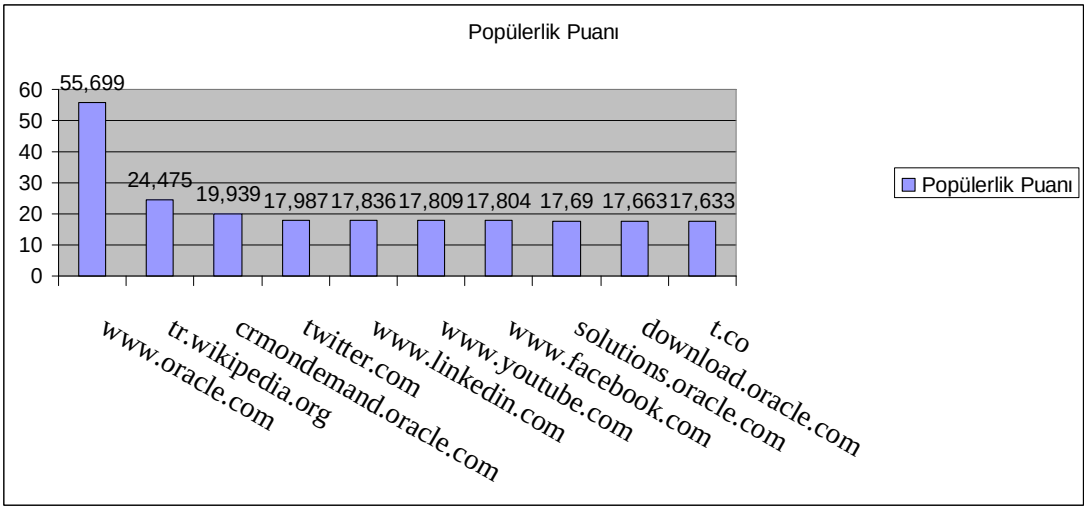
Tablo 4.2 Popülerlik Puanlarına Göre İlk On Adres

X	Y	Z	X_p	Y_p	Z_p	T_p	P
3974789626	4269535840	2472232949	88,29	11,278	29,357	65,798	55,699
3974789626	812950790	2472232949	88,29	9,496	29,357	65,825	55,259
3974789626	2946013137	2472232949	88,29	5,832	29,357	65,795	54,337
3974789626	949306565	2472232949	88,29	4,299	29,357	65,828	53,96
3974789626	956914949	2472232949	88,29	4,128	29,357	65,795	53,911
3974789626	2880589850	2472232949	88,29	3,796	29,357	65,806	53,83
3974789626	2551016791	2472232949	88,29	3,525	29,357	65,795	53,76
3974789626	3095761536	2472232949	88,29	3,039	29,357	65,806	53,641
3974789626	782090849	2472232949	88,29	0,559	29,357	65,823	53,024
3974789626	1899858770	2472232949	88,29	0,226	29,357	65,83	52,942

Tablo 4.3 X,Y ve Z değerlerinin Metinsel İfadeleri

Kök	Yol	Sorgu	X	Y	Z
www.oracle.com	/technetwork /indexes/ Documentation /index.html		2298577483	812950790	2472232949
www.oracle.com	/technetwork /indexes/ Downloads /index.html		2298577483	2551016791	2472232949
www.oracle.com	/technetwork/ Systems /index.html		2298577483	3095761536	2472232949
www.oracle.com	/technetwork/ java/index.html		2298577483	782090849	2472232949
www.oracle.com	/technetwork/ community/ oracle-ace /index.html		2298577483	1899858770	2472232949
www.oracle.com	/technetwork /index.html		2298577483	1076000419	2472232949
www.oracle.com	/partners/en/ knowledge-zone/ index.html		2298577483	2472232949	2472232949
www.oracle.com	/partners/index.html		2298577483	3738137711	2472232949
www.oracle.com	/us/corporate/ customers/ customersearch/ index.html		2298577483	2549979663	2472232949
www.oracle.com	/partners/en/ most-popular -resources/ Enablement -028916.htm		2298577483	1394345101	2472232949

Tablo 4.3'de Tablo 4.2'de verilen X, Y ve Z değerlerinin metinsel açıklamaları bulunmaktadır.



Şekil 4.18. Alan Adlarına Göre Popülerlik Puanları

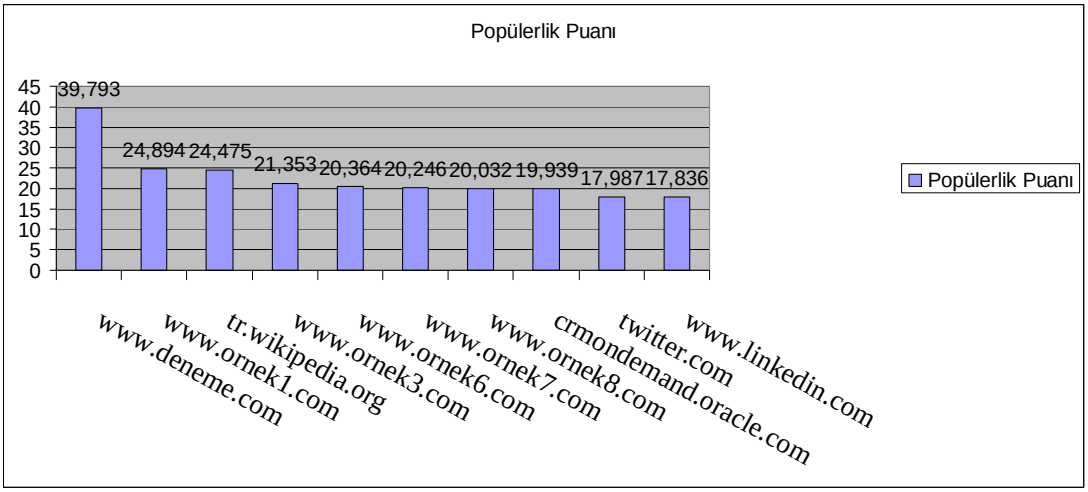
Popülerlik puanlarına göre sıralanan adresler, alan adlarına göre gruplandırılmıştır. Şekil 4.18'de görüldüğü gibi en yüksek puanı alan adres ismi www.oracle.com" olmaktadır. En yüksek puana sahip alan adının www.oracle.com olmasının başlangıç adresi olarak verilmesinden kaynaklandığı anlaşılmaktadır. Arama robotunun çalışmasına devam edilmesi durumunda diğer alan adlarının puanlarının yükseleceği beklenmektedir. Tablo 4.4' de popülerlik puanlarına göre alan adları gözükmektedir.

Tablo 4.4 Popülerlik Puanlarına Göre Alan Adları

Alan Adı	**Popülerlik Puanı**
www.oracle.com	55,699
tr.wikipedia.org	24,475
crmondemand.oracle.com	19,939
twitter.com	17,987
www.linkedin.com	17,836
www.youtube.com	17,809
www.facebook.com	17,804
solutions.oracle.com	17,69
download.oracle.com	17,663
t.co	17,633

Yöntemin çalışmasını gözlemlemek için yapay üretilmiş veriler üzerinden de popülerlik puanları hesaplanmıştır. Bu sayede olumsuz durumlarda yöntemin beklenen davranışları sergileyip sergilemediği kontrol edilmek istenmiştir. Üretilmiş verilerde en olumsuz durumun yaşanabilmesi için ilk 300.000 kayıt birbirinden bağımsız alan adları, yol ve sorgulardan oluşturulmuştur. Geri kalan 300.000 kayıt için ise alan adlarının yoğun olduğu bir grup yaratılmıştır. Bu sayede başlangıç adresi olarak verilen adresin puanlarının yüksek çıkması önlenmeye çalışılmıştır.

İlk kaydedilen ilk ziyaret edilir yöntemine göre URL ataması yapılması durumunda yoğunluğa sahip olan; yani ziyaret edilecek adres listesinde kök, yol, sorgu ve zaman olarak en çok tekrarlı gruba sahip adreslerin ziyaret edilmesi için baştaki 300.000 kaydın arama robotu tarafından ziyaret edilmesi gerekmektedir. Önerilen URL atama yöntemine göre ise yoğun olan grubun daha popüler adresler olduğu varsayıldığı için arama robotu tarafından ilk olarak ziyaret edilmeye başlanmıştır. Şekil 4.19'da üretilmiş veriler üzerinden hesaplanan en yüksek popülerlik puanına sahip ilk on alan adı gözükmektedir.



Şekil 4.19. Üretilmiş Veriler İçin Popülerlik Puanları

Şekil 4.19'de gözüktüğü gibi kayıt edilme sırasına göre kayıtların son grubunda olan "deneme.com" diğer adreslerden yoğun bir gruba dâhil olduğundan dolayı popülerlik

puanı yükselmiş ve URL atamasında Tablo 4.5' de gözüktüğü gibi arama robotunun ilk ziyaret edeceği adres olmuştur.

Tablo 4.5 Üretilmiş Veriler İçin Popülerlik Puanlarına Göre İlk On Adres

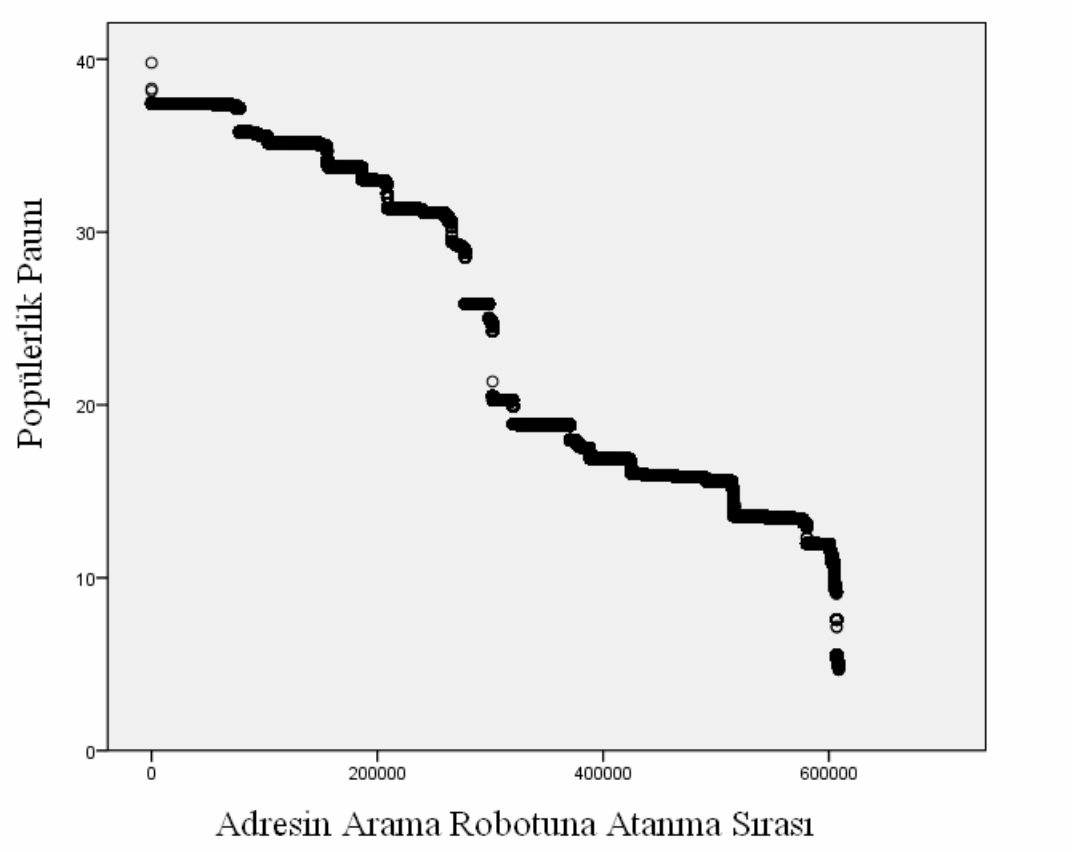
X	Y	Z	X_p	Y_p	Z_p	T_p	P
2298577483	812950790	2472232949	49,626	9,496	29,357	65,825	39,793
2298577483	2551016791	2472232949	49,626	3,525	29,357	65,795	38,294
2298577483	3095761536	2472232949	49,626	3,039	29,357	65,806	38,175
2298577483	782090849	2472232949	49,626	0,559	29,357	65,823	37,559
2298577483	1899858770	2472232949	49,626	0,226	29,357	65,83	37,477
2298577483	1076000419	2472232949	49,626	0,097	29,357	65,83	37,444
2298577483	2472232949	2472232949	49,626	0,079	29,357	65,831	37,44
2298577483	3738137711	2472232949	49,626	0,076	29,357	65,831	37,439
2298577483	2549979663	2472232949	49,626	0,059	29,357	65,83	37,435
2298577483	1394345101	2472232949	49,626	0,059	29,357	65,83	37,435

Tablo 4.6 X, Y ve Z değerlerinin Metinsel İfadeleri

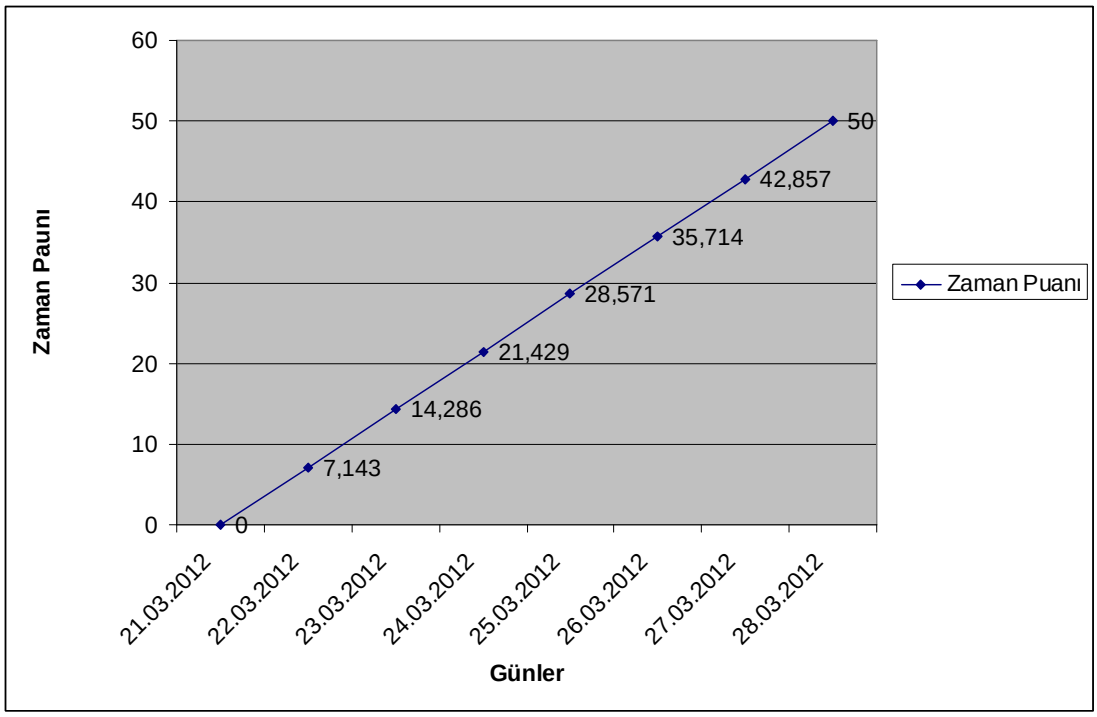
Kök	Yol	Sorgu	X	Y	Z
www.deneme.com	/technetwork /indexes /downloads /index.html		2298577483	812950790	2472232949
www.deneme.com	/partners/en /knowledge-zone/index.html		2298577483	2551016791	2472232949
www.deneme.com	/partners/index.html		2298577483	3095761536	2472232949
www.deneme.com	/us/corporate /customers /customersearch /index.html		2298577483	782090849	2472232949
www.deneme.com	/partners/en /most-popular-resources /enablement-028916.htm		2298577483	1899858770	2472232949
www.deneme.com	/us/products /productslist /index.html		2298577483	1076000419	2472232949
www.deneme.com			2298577483	2472232949	2472232949
www.deneme.com	/us		2298577483	3738137711	2472232949
www.deneme.com	/us/syndicatio n/subscribe /index.html		2298577483	2549979663	2472232949
www.deneme.com	/us/Technologies /cloud/index.html		2298577483	1394345101	2472232949

Tablo 4.6’de Tablo 4.5’de verilen X, Y ve Z değerlerinin metinsel açıklamaları bulunmaktadır.

Şekil 4.20’de üretilmiş adreslerin yöntemimize göre gelme sıraları gözükmektedir. Şekilden anlaşıldığı gibi popüler olan adresler ilk sıralarda URL atayıcı tarafından arama robotuna atanmaktadır.



Şekil 4.20. Üretilmiş Adreslerin Gelme Sırası



Şekil 4.21 Zaman Puanı Gösterimi

Şekil 4.21'de gözüktüğü gibi örnek olarak seçilen adresin, URL atayıcı tarafından arama robotuna atanmadığı her gün zaman puanı artmaktadır. Bu sayede yoğunluğa sahip olmasa da zaman puanı sayesinde adresin genel popülerlik arttığı için arama robotu tarafından ziyaret edilmesine imkân sağlanmıştır.

Geliştirilen yöntemin uygulanabilinmesi için her URL atama işlemi esnasında tüm adreslerin hesaplamaya dâhil olması ve popülerlik puanlarının hesaplanması gerekmektedir. Hesaplama işlemlerinin 3 dakikada tamamlandığı gözlemlenmiştir. Her URL atama isteği için aynı hesaplamaların yapıldığı düşünülürse, bu işlemlerin sunucu bilgisayara büyük bir yük getirdiği anlaşılmaktadır. Bu sorunu aşabilmek için hesaplama işlemleri 3 saatlik periyotlar halinde ayarlanmıştır.

Geliştirilen yöntem sayesinde diğerlerinden daha yoğun adres gruplarına sahip adresler öncelikli olarak ziyaret edilip, arama motorunun indeksine kaydedilmektedir. Bu sayede daha popüler olduğu varsayılan adresler diğerlerinden önce indekslenerek arama sonuçlarında çıkması sağlanmıştır.

Popülerlik puanı daha düşük olan adresler ise zaman puanları sayesinde, bekleme süreleri uzadıkça aldıkları puanlar arttırılmış; böylece onlarda arama robotu tarafından ziyaret edilebilinmiştir.

5.SONUÇ

Arama motorları bilgiyi arama ve ulaşmada şuan için en önemli araçlardır. Arama motorlarının web sayfalarını indeksleyip, sorgularda kullanabilmesi için arama robotlarını kullanmaktadır.

Arama robotları web sayfalarını ziyaret edip içeriklerini arama motorunun veri tabanına kaydetmektedirler. Ziyaret ettikleri sayfalardan da elde ettikleri yeni adresleri daha sonra ziyaret edebilmek için arama motorunun veri tabanına kaydetmektedirler.

Kaydedilen bu ziyaret edilmemiş adreslerin hangi sırada ziyaret edileceği arama motorları için önem arz etmektedir. Bu yüzden arama robotlarına yeni URL atamak için yöntemler geliştirilmiştir. Böylece arama robotunun daha popüler ya da önemli olan adresleri daha önce ziyaret etmesi hedeflenmiştir.

Bu tez çalışmasında arama robotlarına yeni URL atamak için kullanılan yöntemlere yeni bir yaklaşım önerilmektedir. Yöntemin çalışmasının gözlenmesi amacıyla bir arama robotu geliştirilmiştir. Geliştirilen arama robotunun esnek bir yapıya sahip olması, paralel işlemci mimarisini desteklemesi ve sistem kaynaklarını etkin kullanması hedeflenmiştir.

Geliştirilen yöntemin uygulaması gerçekleştirilmiştir. Arama motorunun ziyaret edilecek adres listesi kullanılarak popülerlik puanları hesaplanmıştır. Hesaplanan puanlara göre URL atama işlemi gerçekleştirilmiştir.

Popülerlik puanlarına göre URL ataması yapıldığında daha yoğun olduğu kabul edilen adreslerin arama robotu tarafından daha önce ziyaret edildiği tespit edilmiştir.

Popüler olmayan diğer adreslerinde zaman puanlarının artması ile birlikte arama robotları tarafından ziyaret edileceği gözlenmiştir.

Önerilen URL atama yöntemine, adreslere ait diğer parametrelerinde eklenmesinin popüler adreslerin bulunmasına katkı sağlayacağı değerlendirilmektedir.

Bu tez kapsamında önerilen URL atama yöntemi ile arama motorlarının daha yoğun adresleri daha önce indeksleyebildiği görülmektedir.

KAYNAKLAR

[1] Bolt, Beranek & Newman Inc.,A History of the ARPANET: The First Decade, Arlington, 1981.

[2] http://news.netcraft.com/archives/2012/01/03/january-2012-web-server-survey.html, (02.03.2012)

[3] Deutsch P., Archie—A Darwinian Development Process.. IEEE Internet computing, 2000.

[4] http://ksi.cpsc.ucalgary.ca/archives/WWW-ALK/www-talk-1993q2.messages/706.html, (02.03.2012)

[5] http://stuff.mit.edu/afs/sipb/user/mkgray/tmp/coolwwwmail, (02.03.2012)

[6] Pinkerton B., Finding what people want: Experiences with the WebCrawler, In Proceedings of the First World Wide Web Conference, Geneva, Switzerland, 1994.

[7] Trex E., "Jerry and David's Guide to the World Wide Web becomes Yahoo!". Blogs.static.mentalfloss.com, 2010.

[8] Brandt R., "Starting Up. How Google got its groove". Stanford magazine, 2009.

[9] http://ir.baidu.com/phoenix.zhtml?c=188488&p=irol-homeprofile, (08.03.2012)

[10] Microsoft. Retrieved, "Microsoft’s New Search at Bing.com Helps People Make Better Decisions: Decision Engine goes beyond search to help customers deal with information overload", 2009.

[11] http://myip.ms/browse/web_bots/ Known_Web_Bots_Web_Bots_2012_Web_Spider_List.html, (29.05.2012)

[12] Ghemawat S., Gobioff H., Leung S., The Google File System, SOSP’03, Bolton Landing, New York, USA, 2003.

[13] http://www.ietf.org/rfc/rfc3986.txt, (17.03.2012)

[14] http://www.press.umich.edu/jep/07-01/bergman.html, (25.03.2012)

[15] Wang X., Wang L., Wei G., Zhang D., Yang Y., Hıdden Web Crawlıng For Sql Injectıon Detectıon, Broadband Network and Multimedia Technology (IC-BNMT), 3rd IEEE International Conference, 2010

[16] Cho J., Garcia-Molina H., Parallel Crawlers , In Proceedings of the 11 th

International Conference on World Wide Web, USA, 2002

[17] Boldi P., Codenotti B., Santini M., Vigna S., Ubicrawler: A scalable fully distributed web crawler, The Eighth Australian World Wide Web Conference, 2002.

[18] http://www.robotstxt.org/robotstxt.html, (07.04.2012)

[19] http://www.sitemaps.org/protocol.html, (07.04.2012)

[20] Sun Y., Councill I., Giles L., The Ethicality of Web Crawlers, IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2010.

[21] Cho J., Garcia-Molina H., The Evolution of the web and implications for an incremental crawler, 8thInt.WorldWide Conference (WWW8), 1999.

[22] Chen Y., Tsai F., Chan K., “Blog search and mining in the business domain,”, ACM, 2007.

[23] https://www.google.com.tr/preferences?hl=tr, (14.04.2012)

[24] http://www.bing.com/settings.aspx?sh=2&FORM=WIWA, (14.04.2012)

[25] Lin S., Li Y., Li Q., Information Mining System Design and Implementation Based on Web Crawler, IEEE ,2008.

[26] Huitema P., Fizzano P., A Crawler for Local Search, Fourth International Conference on Digital Society, 2010.

[27] http://msdn.microsoft.com/en-us/library/ms717470.aspx, (24.04.2012)

[28] http://googledesktop.blogspot.com, (24.04.2012)

[29] Sigurðsson, K., "Incremental crawling with Heritrix". Proceedings of the 5th International Web Archiving Workshop (IWAW’05), 2005.

[30] http://www.dataparksearch.org, (02.05.2012)

[31] http://www.aspseek.org, (02.05.2012)

[32] http://www.httrack.com/page/6/en/index.html, (02.05.2012)

[33] http://nutch.apache.org, (02.05.2012)

[34] http://www.iana.org/assignments/uri-schemes.html, Official IANA-registered schemes, (14.05.2012)

[35] http://www.oracle.com/technetwork/licenses/database-11g-express-license-459621.html, (14.05.2012)

[36] http://www.java.com/tr/about, (15.05.2012)

[37] Cho J., Garcia-Molina H., Page L, Efficient crawling through URL ordering, Proceedings of 7th World Wide Web Conference, Brisbane, Australia, 1998.

[38] http://btgrubu.com/index.php?type=urunler&value=embarcadero, (15.05.2012)

ÖZGEÇMİŞ

1979 yılında İstanbul'da doğdu. İlköğrenimini Şair Nedim İlkokulunda, orta öğrenimini Etiler Otelcilik ve Turizm Meslek Lisesi'nde tamamlamıştır.

1998 yılında Maltepe Üniversitesi Mühendislik Mimarlık Fakültesi Bilgisayar Mühendisliği Bölümünü kazandı ve 2003 yılında mezun oldu. 2004 yılından itibaren Deniz Kuvvetleri Komutanlığında görev yapmaktadır.

2004 yılında, Maltepe Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Yüksek Lisans programında yüksek lisans öğrenimi yapmaya hak kazandı. Bir süre ara verdiği eğitim hayatına 2011 senesinde tekrar başlamıştır.

Evli ve bir kız çocuğu babası olan Ahmet Erdem KARACA, İngilizce bilmektedir.