T.C
FIRAT ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

PHP & MySQL KULLANARAK SQUİD PROXY LOG ANALİZİ

BİTİRME ÖDEVİ

DANIŞMAN
Yrd. Doç. Dr. Hasan H. BALIK

HAZIRLAYAN
Erhan YELİ

ELAZIĞ, 2003

T.C

FIRAT ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

PHP & MySQL KULLANARAK SQUİD PROXY LOG ANALİZİ

BİTİRME ÖDEVİ

Bu tez, ... tarihinde aşağıda belirtilen jüri tarafından oybirliği /oyçokluğu ile başarılı / başarısız olarak değerlendirilmiştir.

(İmza)	(İmza)	(İmza)
Danışman:	Üye:	Üye:

İÇİNDEKİLER

ŞEKİL LİSTESİ

ÖZET

Bitirme Ödevi

PHP & MySQL KULLANARAK SQUİD PROXY LOG ANALİZİ

Erhan YELİ

FIRAT ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

28 Sayfa
2003
Danışman : Yrd. Doç. Dr. Hasan H. BALIK

Bu çalışmada, Squid Proxy Server'in kurulumu, yapısı, çalışması incelenmiştir ve yapılan ödevin amacı olarak ta Squid Proxy'nin access.log dosyası okunarak log analizi yapılmıştır.

Ayrıca Squid Proxy'nin log analizini yapan diğer programlar incelenmiştir.

Log analizi yapılırken access.log dosyasının yapısı incelenmiş, PHP ile parse edilip MySQL veritabanına atıldıktan sonra, çeşitli sorgular sonucunda istenen veriler elde edilmiştir. Bu veriler PHP kullanılarak web ortamına yansıtılmıştır.

Squid Proxy log analizi yapılırken iki durum dikkate alınarak yapılmıştır.

Birincisi, geçmiş zamana ait log analizi. Bir kaç gün öncenin yada bir süre öncenin access.log dosyası alınır. Bu dosya PHP & MySQL ile işlenerek çeşitli raporlar alınır. Elde

edilen bu raporlar geçmiş zamana ait raporlardır. Raporlarda elde edilen sonuçlar; kim, saat kaçta, hangi siteye girmiş, en çok dosya indiren kullanıcılar, en çok web bağlantısı açan kullanıcılar, en çok girilen siteler ve bu sitelere kimlerin girdiği, genel istatistikler, cache istatistikleri....vs bir çok bilgi elde edilir.

İkincisi, realtime log analizi. Yani gerçek zamanlı olarak kimin hangi siteye girdiğinin incelenmesi. Bu işlem sırasında access.log dosyası etkin rol oynamaz. Linux, Squid Proxy, PHP & MySQL kullanılarak sonuçlar web'de gösterilir.

Linux RedHat 8 işletim sistemi altında PHP & MySQL ikilisi kullanılarak bu proje gerçekleştirilmiştir. Veritabanı kontrolü için web arayüzlü PHPMyAdmin programından yararlanılmıştır.

Anahtar Kelimeler : Log analizi, Squid Proxy log analizi, PHP & MySQL ile log analizi, PHP & MySQL ile Squid Proxy log analizi, Squid Proxy'nin realtime log analizi, Squid Proxy'nin PHP & MySQL ile realtime log analizi.

TEŞEKKÜR

Bitirme ödevi hazırlarken gerekli olanakları sağlayan, maddi ve manevi desteğini esirgemeyen değerli hocam Sayın Yrd. Doç. Dr. Hasan H. BALIK ‘a,

Squid Proxy konusunda her türlü yardımı yapan Sayın Bil. Müh. Gürkan KARABATAK’a,

Bu günlere kadar gelmemde emeği olan bütün hocalarıma,

Tüm hayatım boyunca bana her konuda destek olan aileme, en samimi duygularımla teşekkür ederim.

1. GİRİŞ

1.1. Ödevin Amacı

Ödevi yapma amacım Linux altında çalışan Squid Proxy'nin nasıl çalıştığını öğrenmek; Squid Proxy log dosyasını analiz edip, bu log dosyasını PHP & MySQL ile işleyerek kulanıcılar ile ilgili raporlar çıkarmaktır. Örneğin en çok girilen siteler, en çok dosya indiren kullanıcılır, cache istatistikleri, toplam indirilen dosya boyutu, cache kulanım yüzdeleri…

Ayrıca Squid Proxy Server'ı kullanarak gerçek zamanlı olarak kullanıcıların girdiği siteler, yazılan siteye giren kulanıcılar, yazılan alt ağdaki bilgisayarların girdikleri web siteleri, yazılan kelime ile ilgili web sitelerine giren kullanıcılar ve girdikleri web siteleri… gibi bilgileri elde etmek amaçlanmıştır.

1.2. Ödevin İçeriği

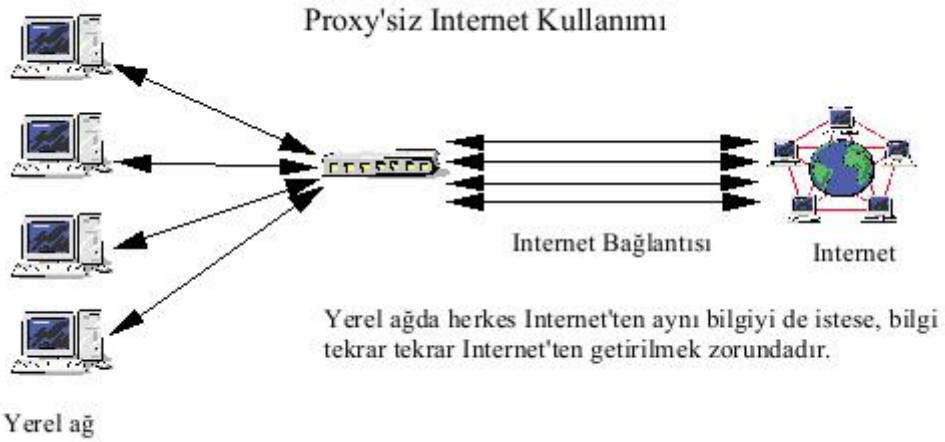
Squid Proxy Server'ın ne işe yaradığı ve Linux altında kurulumu anlatılmıştır. Daha sonra log dosyası PHP ile pars edilip MySQL veritabanına aktarılmıştır. MySQL veritabanında sorgular sonucunda geçmiş zamana ait veriler elde edildi. Örneğin bir gün önceki kayıtlar. Bu kayıtlarda en çok ziyarek edilen siteler, o gün girilen site sayısı, kimlerin hangi sitelere girdiği, en çok dosya indiren kullanıcılar, cache en çok meşkul eden kullanıcılar…vs bir çok bilgi elde edildi.

Linux, Squid Proxy, PHP ve MySQL entegrasyonu ile realtime kimlerin hangi sitelere girdiğini, verilen web sitesine online olarak giren kullanıcıların kimler olduğunu, verilen kelimenin geçtiği web adresi ve bu adreslere kimlerin girdiğini, bulup PHP ile webde yayınlayabiliyoruz.

2. SQUİD PROXY SERVER

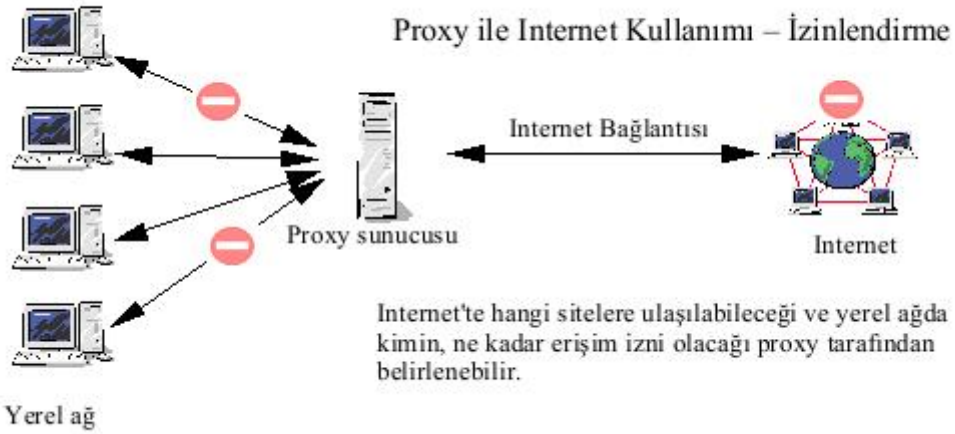
2.1. Proxy Nedir?

Bir ağda bulunan istemciler adına, internet'ten nesneleri (web sayfası, resimler, dosya v.b.) alıp, isteği yapan istemciye ulaştıran sistemdir. Bir proxy sunucusu, tek bir hattı paylaşarak, birçok kişiye internet bağlantısı sağlamaktadır. İyi bir proxy aynı zamanda isteklerin bir kopyasını da saklamaktadır, ki başka birisi bu bilgilere ulaşmak istediğinde, yavaş olan internet'en getirmek yerine yerel kopyayı sunsun. Böylece, bilgilere olan erişim zamanı kısalmakta ve hat kullanımı azalmaktadır.



Şekil 1 Proxy'siz internet kullanımı

Şekil 2 Proxy ile internet kullanımı



Şekil 3 Proxy ile kullanıcı izinlendirme

2.2. Caching Nedir?

Bu nesneler istemciye ulaştırılırken, bir kopyasının da, daha sonraki benzer istekleri karşılamak üzere disk üzerinde tutulmasıdır.

2.3. Squid Proxy Nedir?

Squid, Harvest programının devamı olarak NLANR tarafından Digital Unix bir makine üzerinde geliştirilmiştir. Özellikle Unix makineler üzerinde çalışmak için dizayn edilmiş olsa da pek fazla edilmese de, WindowsNT üzerinde de çalışabilen yüksek performanslı, web istemcilerinin isteklerine yanıt veren bir vekil (proxy) programıdır. Squid diğer geleneksel proxy ürünlerine benzemez. Bütün istekleri tek, bloklandırılmamış bir I/O işlemi ile yürütür. HTTP istemcilerinin taleplerini karşılayan, yüksek hızlı ve caching yapabilen bir proxy sunucudur. Squid sadece HTTP proxy olarak çalışır ama bunun yanında SSL ve HTTP üzeri FTP, tranparent caching, cache hiyerarşilerini, HTTP sunucu yansılama (accelerator), SNMP protokollerini destekler. Standart proxy, transparent proxy, reverse proxy gibi kullanım çeşitleri vardır. Bunların oluşum ve kullanım şekli router yada firewall'daki access list'lerin düzenlenmesiyle sağlanır.

Squid internet bilgilerini cache tutmaya yarayan bir yazılımdır. Siz bir siteye bağlanmak istediğiniz zaman, bağlanmak istediğiniz sitenin sunucusu sitedeki squid haber verir, daha sonra squid uzaktaki sunucuya bağlandıktan sonra, istenilen sayfayı indirir. Sayfanın bir kopyasınıda kendi bünyesine alır. İkinci sefer aynı siteye bağlanmak istediğiniz zaman siteyi diskten size getirir. Böylece daha kısa zamanda istediğiniz bilgiye erişirsiniz. ICP(Internet Cache Protocol) protokolü sayesinde squid diğer sistemlerin cache'leri arasında hiyararşik bir düzen kurar

(Birbirine izin veren sistemler için geçerlidir. Mesela ULAKNET'in sayesinde üniversiteler arasında hızlı iletişim sağlaması). Böylece kendisinde olmayan bilgiler iletişimde bulunduğu sistemler de varsa kendi aynı bilgiyi kendi cache'ine yazmayacaktır. Bu da büyük oranda disk kullanımını azaltacaktır. Siz bir web sitesi için istekte bulunduğunuzda squid ilk önce kendi cache'ine bakar, bulamazsa hiyerarşi içinde bulunduğu diğer sistemlerin cache'lerini kontrol eder. Orada da bulamazsa web sitesinin bulunduğu sunucuya bağlanır ve istediğiniz sayfayı indirdikten sonra bir kopyasını da kendi cache'ine kopyalar. Bu sayede istenilen bilgiye daha kısa sürede ulaşabiliriz.

Squid HTTP, FTP, GOPHER, SSL protokollerini destekler. Tabii ki squid'te herşey saklanmaz. örneğin çalıştırılabilen cgi programları, haber sayfaları gibi sürekli olarak yenilenen siteler, kredi kartı numaraları cache tutulmaz.

Squid AIX, Digital Unix, FreeBSD, HP-UX, Irix, Linux, NetBSD, Nextstep, SCO, Solaris gibi işletim sistemlerinde çalışır.

2.4. **Web Caching Ne Sağlar?**

Caching, dünya web ağını (www) dolaşırken, kullanıcıların daha verimli bir şekilde popüler olan Internet objelerine erişiminin yapılmasını sağlar. Böylece popüler çağırılan objeler, internet erişimi yapılamadan cache'lerden sunulur. Dolayısıyla web caching internet bant genişliğimizi efektif kullanmamamızı sağlar. İkinci olarak, popüler objeler çağırıldığında, bekleme süremiz düşer, son olacakta web sunucularımız üzerinde yük hafifletilmiş olur.

2.5. Donanım İhtiyacı

Donanım ihtiyacı belirlenirken belli ihtiyaç parametrelerine sahip olunmalıdır. Örneğin dakikadaki istek sayısının en üst değeri ne olabilir? Bu sayı, client'lar tarafından download edilebilecek objelerin sayısını gösterir. Eğer bunu belirlemek zor ise herhangi bir makineye yükleyip bazı istatistikler edinmek kolay yol olacaktır. Burada şu söylenebilir P133, 64 MB RAM, 2 GB disk ile internet çıkışının 512K' dan düşük olduğu durumlarda iyi sonuçlar alınabilir. Dakika'da 50-70 arasındaki isteği çok rahat kaldırabilir.

2.6. Hard Disk

Random seek time ne kadar küçük olursa, o kadar iyi olur. Diskin kafasının bir track'ten başka bir track'e giderken geçen zaman, seek time'i verir. Farklı birkaç disk kullanılacaksa aynı tipte olması tavsiye edilir yada cache_dir ler aynı disklerde tutulmalıdır. Squid den yapılan istek eğer obje cache'de yoksa orijin sunucudan alınacak ve ayni anda hem istek yapan makineye

aktarılacak hemde cache'e kopyalanacaktır (tabi obje cachelenebilir bir obje ise). Çok açıktır ki cacheden yapılabilecek istek sayısı özellikle bu parametre ile sınırlıdır.

Yapısı gereği SCSI disklerin tercih edilmesi gerekiyor ama küçük networkler için bu da çok önemli bir parametre olmaktan çıkıyor.

2.7. RAM

Küçük pointer yapılarda her StoreEntry için (cach'lenen her objenin memoryde tutulan kısmı) 56 byte + 16 byte da MD5 chechsum için memory de alan harcanır. Böylece toplam 72 byte alan kullanılan her metadata için kullanılır. 1.000.000 objeye sahip olan bir cache sadece 72 MB memory, metadata için kullanılır.

Squid RAM'de objelerin bir tablosunu tutar. Objenin file store'da olup olmadığını kontrol etmek için kullanır bu tabloyu. Böylece tabloya hızlı erişim yapılmış olur. Son derece açıktır ki, istenen objenin nerede olduğu, en hızlı böyle okunabilir.

- Disk buffer (okuma ve yazma işlemleri için)
- Hot objeler
- Network I/O bufferları
- IP CACHE içerikleri
- Her istemin durum bilgisi

Bütün bunlar Squid tarafından bellekte tutulan ve korunan nesnelerdir.

2.8. CPU

Squid sadece açılış sırasında biraz CPU'yu kullanır. Bu aşamada yavaş yanıt verebilir ama birkaç dakika sonra hız alacaktır. Multiprocessor performansı beklendiği gibi arttırmaz.

2.9. İşletim Sistemi Seçimi

Squid, in Harvest projesi devamında NLANR tarafından dijital UNIX bir makine üzerinde geliştirildiğini söylemiştik dolayısıyla Unix makinelerde daha performanslı çalışacağı kesindir. Solaris için açık üstünlüğü görülür. Ama önemli olan elimizdeki kaynakları en iyi şekilde kullanmayı bilmektir. *BSD, Linux vs. hangisinin olduğu çok önemli değil. Hangisini daha iyi yönetebiliyorsak onu tercih edebiliriz.

2.10. Temel Sistem Düzeni

~/bin altında _ Binary dosyalar

~/etc altında _ Config. Dosyaları (özellikle squid.conf)

~/cache altında _ Cache dosyaları

~/logs altında _ Log dosyaları bulunur

2.11. User ve Group ID

Squid diğer birçok UNIX daemonları gibi normalde nobody, nogroup user/group yetkilerini kullanarak çalışır. Tavsiye edilen ise "squid" adında bir user ve group yaratıp onun kullanılmasıdır. Yaratılan user'ın home dizini olarak /usr/local/squid i set etmeyi unutmamalıyız. Tabii bu durumda bir de /squid/etc/squid.conf daki

- cache_effective_user squid
- cache_effective_group squid

olarak değiştirilmelidir.

Burada dikkat edilmesi gereken konu, UNIX lerde portlarda 1024 altını kullanmak root hakkıyla yapılabileceği için squid de kullanılacak portlardan biri 1024 olacaksa squid, root kullanıcısı ile çalıştırılmalıdır.

2.12. Squid Derlenmesi ve Kurulumu

[root@hayal squid]# groupadd squid
[root@hayal squid]# useradd –d /usr/local/squid –g squid squid
[root@hayal squid]# tar –zxvf squid-2.4.STABLE3-src.tar.gz
[root@hayal squid]# cd squid-2.4.STABLE3
[root@hayal squid]# ./configure –prefix=/usr/local/squid
[root@hayal squid]# make
[root@hayal squid]# make install

Squid.conf dosyasını açmak için
[root@hayal squid]# pico /usr/local/squid/etc/squid.conf

/usr/local/squid/etc/squid.conf dosyası içinde bulunan aşağıdaki satırların başında bulunan '#' ler kaldırılır.

- cache_dir ufs /usr/local/squid/cache 100 16 256

Yukarıda bu squid'in 100MB'lık bir disk alanı kullanacağı ve bu /usr/local/squid/cache dizinin altında 16 dizin ve bu 16 dizinin her birinin altında 256 tane dizin oluşturulacağını ifade eder. Bu şekilde bir yapı olmasının sebebi Squid'in bilgilere daha çabuk ulaşmasıdır. Eğer hepsi birkaç dizinde olsaydı, bir bilgi için tüm dizini arayacaktı. Bu ise çok fazla zaman kaybına yol açar. Ön tanımlı değerleri değiştirmek için ifadenin başındaki # işaretini kaldırdıktan sonra değişiklik yapmanız gerekir.

- cache_effective_user squid
- cache_effective_group squid
- http_port 3128
- http_access allow all (#http_access deny all)

[root@hayal squid]# chown squid:squid /usr/local/squid/logs

[root@hayal squid]# chmod 770 /usr/local/squid/logs

Directory İzinleri ve Cache Dizini

[root@hayal squid]# mkdir /usr/local/squid/cache/

[root@hayal squid]# chown squid:squid /usr/local/squid/cache/

[root@hayal squid]# chmod 770 /usr/local/squid/cache/

[root@hayal squid]# /usr/local/squid/bin/squid -z

2002/01/09 19:15:34| Creating Swap Directories

Çalıştırmak için

[root@hayal squid]# /usr/local/squid/bin/squid –N –d 1 –D

Squid hakkında ayrıntıları bu program yardımıyla alabilirsiniz. Format == **client mgr:** dir.

[squid@hayal squid]$ client mgr: |more

```
 cbdata              Callback Data Registry Contents public
mem                  Memory Utilization      public
events               Event Queue     public
config               Current Squid Configuration     hidden
comm_incoming          comm_incoming() stats   public
ipcache              IP Cache Stats and Contents     public
fqdncache              FQDN Cache Stats and Contents   public
idns                 Internal DNS Statistics public
http_headers           HTTP Header Statistics  public
menu                 This Cachemanager Menu  public
shutdown               Shut Down the Squid Process     hidden
info                 General Runtime Information     public
```

Client programıyla çalışmasını deniyoruz.

[root@hayal squid]# /usr/local/squid/bin/client http://www.squid-cache.org

```
<HTML>
<HEAD>
<TITLE>Squid Web Proxy Cache</TITLE>
</HEAD>
<BODY BGCOLOR="#ffffff">
<TABLE border="0" cellpadding="8" cellspacing="1" width="100%">
<TR>   <TD BGCOLOR="#CCAEFF" valign="top">
     <dl compact>
     <dt><FONT COLOR="#990000"><BIG><em>Documentation</em></BIG></FONT></dt><small>
     <br>&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;<A HREF="Doc/FAQ/FAQ.html">FAQ</A>
     <br>&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;<A HREF="http://squid.visolve.com/">Configuration Guide</a>
```

Squid 'in cache raporunun alınması.

[squid@hayal squid]$ tail -f /usr/local/squid/logs/cache.log

```
2001/02/08 19:24:14| Set Current Directory to /usr/local/squid/cache
2001/02/08 19:24:14| Loaded Icons.
2001/02/08 19:24:14| Accepting HTTP connections at 0.0.0.0, port 3128, FD 17.
2001/02/08 19:24:14| WCCP Disabled.
2001/02/08 19:24:14| Ready to serve requests. -> İşlem tamam demektir.
```

2.13. Log Dosyaları

/usr/local/squid/logs dizinin altında üç çeşit log dosyası vardır. Bunlar:

cache.log: Squid'in başlatılmasından kapatılmasına kadarki bütün süreçteki durumların kaydının tutulduğu yerdir. Squid düzgün açılmamışsa buradan sebebini öğrenip, çözüm geliştirebilirsiniz. Yada durup dururken birden kapanmışsa nedenini ancak buradan öğrenebilirsiniz. Ancak eğer squid -s parametresi ile başlatılmışsa bu loglar sistem log dosyasına da yazılacaktır.

access.log: Bu dosyada, gelen giden bütün HTTP trafiğinin sonuçlarını görebilirsiniz. Yapılan isteklerin durumu burada tutulur. Native ve Common diye isimlendirilen iki tip formatta bulunur. Native format squid için özeldir. Common tip adından da anlaşılacağı gibi standart web sunucular log formatındadır. Bir çok analiz programı bu dosyayı kullanarak çeşitli istatistik verilerine ulaşmanızı sağlarlar. En çok ziyaret edilen site, kim nereye ne kadar bağlantı kurmuş, ne kadar büyüklükte dosya indirilmiş vb... Genellikle bu dosyada göreceğiniz TCP_REFRESH_HIT, REFRESH_MISS, IMS_HIT gibi kavramlar biraz karıştırılır. Bunların üzerinde kısaca durmak istiyorum.

- **TCP_MISS** : Cache'de bulunamayan isteklerdir.
- **TCP_HIT** : Cache'de bulunan ve cache'den cevaplanan istekler.
- **TCP_REFREST_HIT** : İstenen obje cache'dedir ama cache objenin tamamen bayat olduğuna inandığı için orijin server'a gider ve orijin server'dan gidip bu objenin değişmediği bilgisini alıp, objeyi cache'den sunar. Bu istek REFRESH_HIT ile loglanır. Squid.conf dosyasında refresh_pattern ayarı küçültülürse bu yanıt tipinin arttığını göreceksiniz.
- **TCP_REF_FAIL_HIT** : Nadir olur. Çünkü obje cache'dedir ama bayat olduğu için orijin sunucuya sorulur ama sorguya yanıt alınamadığında bu yanıt mesajı görülür.
- **TCP_REFRESH_MISS** : Obje cache'de ama bayat, orijin sunucudan yapılan sorgulama sonucunda objenin içeriğinin değiştiği yanıtı alınınca bu yanıt mesajı alınıyor.
- **TCP_IMS_HIT** : İstenen objenin cache'de olup, tazelenmeye ihtiyacının olmadığı orijin sunucudan yapılan küçük bir sorgulama sonucu anlaşıldığında bu yanıt alınır.

store.log: Temel olarak bir debug ve transaction log dosyasıdır. İstenen objelerin durum bilgisini cache'e kaydedilip edilmediği, disk üzerinde nereye kaydedildiği, objenin cinsi (html/tex/image) gibi bilgileri içerir.

2.14. Cache Yerleştirme Politikaları

Cache obje yer değişim politikaları kısaca, cache'den hangi objenin kaldırılacağına karar verir. Kullanım amacı ise, daha iyi kaynak kullanımıdır. (Disk, bellek ve ağ bant genişliği gibi) Bellek *(heap*) ve bağlı liste *(linked list)* tabanlı olmak üzere iki algoritma üzerinde geliştirilmişlerdir. Bağlı liste tabanlı çalışan squid için ilk geliştirilmiş olan LRU-L (Least Recently Used -link) dediğimiz algoritmadır. Diğerlerinin hepsi heap tabanlıdır.

2.14.1. LRU-L

LRU-L (Least Recent Used-link), adından da anlaşılacağı gibi bu algoritma, son erişim zamanına bağlı olarak sıralanan bir bağlı liste kullanarak oluşturulmuştur. Bir obje yakın zamanda referans edildiğinde çiftli bağlı listenin başına taşınır. Bu bağlı listeden objelerin çıkartılması sabit bir zaman içerisinde bu liste kuyruğuna erişime bağlıdır.

2.14.2. LFUDA

LFUDA (Least Frequently Used Dynamic Aging), LRU-L gibi bağlı liste tabanlı değil, bellek (heap) temellidir. LFUDA dan önce LFU vardı. LFU çalışma mantığı şu şekilde idi: Her bir cache objesi için bir referans değeri atanır ve bu değer atanan her obje için bir artırılır. Bir

obje cache den kaldırılacağı zaman ona en düşük seviyedeki referans değeri atanır. Bazen LFU policy kurulumu için öncelikli bir kuyruk (heap) kullanılırdı. Fakat LFU'nun kötü bir zaafı vardı. Cache kirlenmesinden çok kötü etkilenirdi. Şayet düzgün popüler olan obje unpopülar olursa, o obje uzun süre cache'de popüler bir obje gibi kalır. Bu yüzden Yeni gelecek olan popular objelere bir süre sonra cache'de yer kalmazdı. İşte bu noktada objeye *Dynamic Aging* ile başka bir referans değeri daha atanarak, objeye bir ömür değeri atandı. Bu ömür değeri dolan cache'den atılıyor.

2.14.3. GDSF

GDSF (Greedy Dual Size Frequency), bu policy yöntemi, cache'de daha popüler ve küçük objeleri tutmak ve böylece obje hit hızını artırmak için düşünülmüştür. Bu yöntem, her objeye, objenin yaşı ve büyüklüğünü, objenin referans değerine bölerek elde ettiği anahtar değeri atar. Yaş faktörünün eklenmesi ile daha önce cache'leşen objelerin etkisini sınırlamış oluruz. Burada, LFUDA da dynamic aging yaptığı işin benzerini yapmış olur. GDSF ile bu özelliği sayesinde daha iyi bir hit oranı elde ederken LFUDA ile ise daha yüksek bir *byte hit* oranına sahip olabilirsiniz.

2.15. Cache Verimliliğinin Ölçümü

Cache verimliliğinizin ölçümünde iki değer çok önemlidir. Bunlar, *Cache Hit Oranı* ile *Byte Cache Hit Oranıdır*.

Cache Hit Oranı = Cachelenen İstek Sayısı / Toplam Yapılan İstek Sayısı

Byte Hit Oranı = Cachenen objelerin byte miktarı / Toplam indirilen objelerin byte miktarı

Peki bu değerleri nereden edinebiliriz**. client mgr:** yada **cachemgr.cgi** sayesinde edinebiliriz.

sample_time = 984769255.324442 (Fri, 16 Mar 2001 19:00:55 GMT)
client_http.requests = 83147 -> Toplam istek sayısı
client_http.hits = 30647 -> Toplam hit sayısı
client_http.errors = 14
client_http.kbytes_in = 34781
client_http.kbytes_out = 456439 -> Toplam byte sayısı
client_http.hit_kbytes_out = 124340 -> Toplam hit byte sayısı
Hit Oranı = 30647 / 83147 = % 37
Byte Hit Oranı = 124340 / 456439 = % 27

2.16. Access.log Dosyasının İncelenmesi

Yaptığım program çalışma şekli olarak iki kısımdan oluşur. Birincisi, geçmişteki logların incelenmesi. İkincisi realtime sitelerin gözlenmesi. Birincisinin çalışırken referans aldığı dosya access.log'dur. Squid proxy'e gelen bütün web istekleri burada belirli bir formatta kaydedilir. Bu dosyayı okuyup işleyerek geçmiş için log analizi yapılır.

[squid@hayal squid]$ tail -f /usr/local/squid/logs/access.log

1054151235.980 1000 10.8.2.164 TCP_REFRESH_HIT/200 45077 GET http://www.komikaze.net/komikfilmler/a.avi - DIRECT/212.98.228.13 video/x-msvideo

1054151235.980 1000 10.8.2.37 TCP_MISS/200 1443 GET http://www.harran.edu.tr/banner/geri.gif - DIRECT/193.140.254.8 image/gif

1054151235.980 1000 10.8.2.37 TCP_MISS/200 9560 GET http://www.harran.edu.tr/banner/logo.jpg - DIRECT/193.140.254.8 image/jpeg

1054151235.980 1000 10.8.2.37 TCP_MISS/200 26614 GET http://www.harran.edu.tr/banner/duyuru.jpg - DIRECT/193.140.254.8 image/jpeg

1054151235.980 1000 10.8.2.37 TCP_MISS/200 20332 GET http://www.harran.edu.tr/gap3.gif - DIRECT/193.140.254.8 image/gif

1054151235.980 47000 10.8.2.248 TCP_MISS/200 138340 POST http://search.springer.de/search97cgi/s97_cgi - DIRECT/195.71.111.125 text/html

1054151235.980 0 10.8.2.164 TCP_HIT/200 49524 GET http://www.komikaze.net/komikfilmler/a.avi - NONE/- video/x-msvideo

Soldan itibaren access.log alanlarını inceleyelim.

- **İlk alan** cache'e istekte bulunulan zamandır. UNİX zaman birimi olan time stamp biçimindedir. Örneğin:1054151235.980
- **İkinci alan** bağlantıda indirilen dosya boyutudur. Byte olarak. İstek cache'den karşılanırsa değeri sıfırdır.
- **Üçüncü alan** istekte bulunan kullanıcının ip numarasıdır.
- **Dördüncü alan** web isteğinin cache'ten mi yoksa web sunucusundan mı olduğunu gösterir. Yanında "/" ile ayrılmış olan sayılar http RFC numarasıdır. Örneğin; TCP_HIT, TCP_MISS,….
- **Beşinci alan** gelen paketin cache'te harcadığı süreyi gösterir.
- **Altıncı alan** gelen paketin metodunu gösterir. GET, POST… metodu.
- **Yedinci alan** ise web adresinin ayrıntısını içerir. Bu ayrıntı gelen paketlerin isimlerini ve ayrıntısını gösterir.

3. PROGRAMIN YAPISI

3.1. Geçmiş Zamana Ait Log Analizi

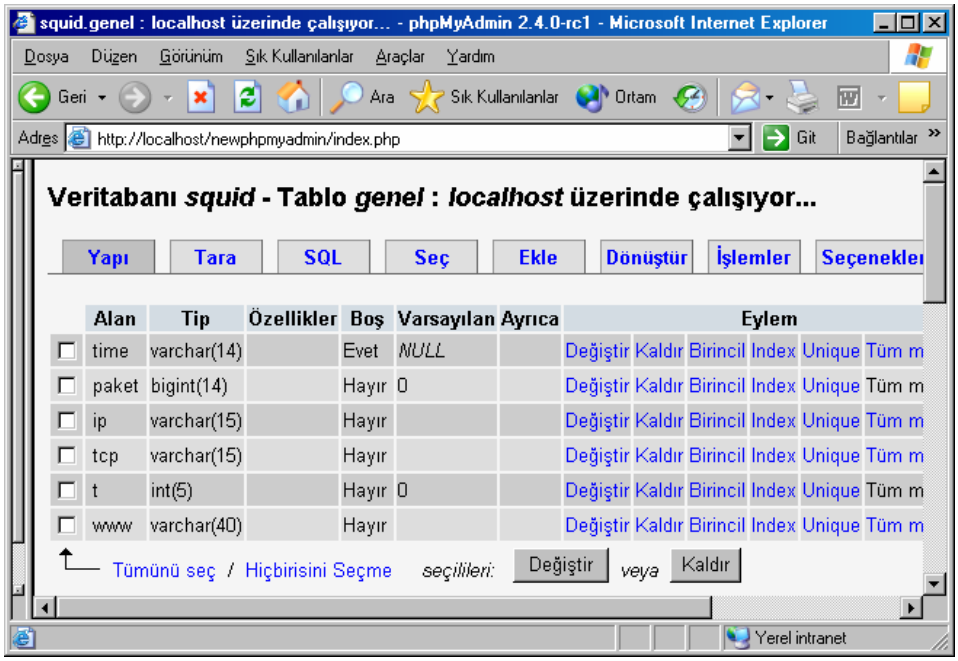
- Access.log'dan okuma ve pars işlemi
- Pars edilen alanları MySQL veritabanında tabloya atma
- MySQL'e sorgu gönderip sorgu sonucunu webde alma.

Bu modülde access.log kullanılır. Access.log yapısı yukarıda belirttiğim gibidir. Önce access.log üzerinde işimize yarayacak alanları belirleyip onları ayıklamamaız gerekir. Bu işleme pars işlemi denir. Pars işlemini PHP ile yapıp gerekli alanları MySQL veritabanına atıyoruz.

Data sonra bize gerekli sorguları PHP ile MySQL'e gönderip sorgu sonuçlarını PHP ile alıp webde yayınlıyoruz.

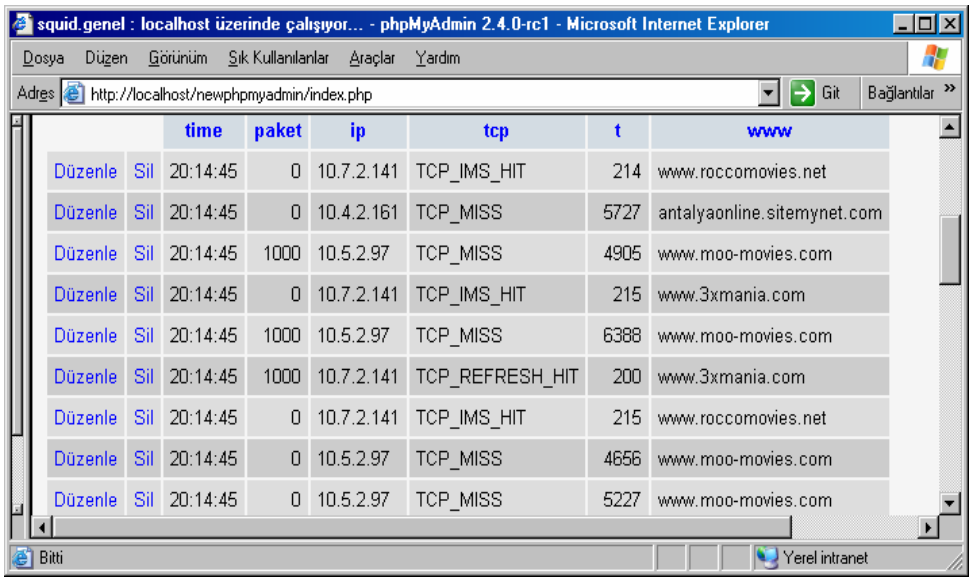
Şekil 4 Geçmiş zamana ait log analizi

Oluşturulan veritabanındaki tablo yapısı PHPMyAdmin görüntüsüyle aşağıdaki şekildeki gibidir.



Şekil 5 PHPMyAdmin ile oluşturulmuş "genel" tablosu

Pars işlemi sonrası veritabanı



Şekil 6 Parse edilip MySQL'e gönderilmiş alanların PHPMyAdmin ile görüntüsü

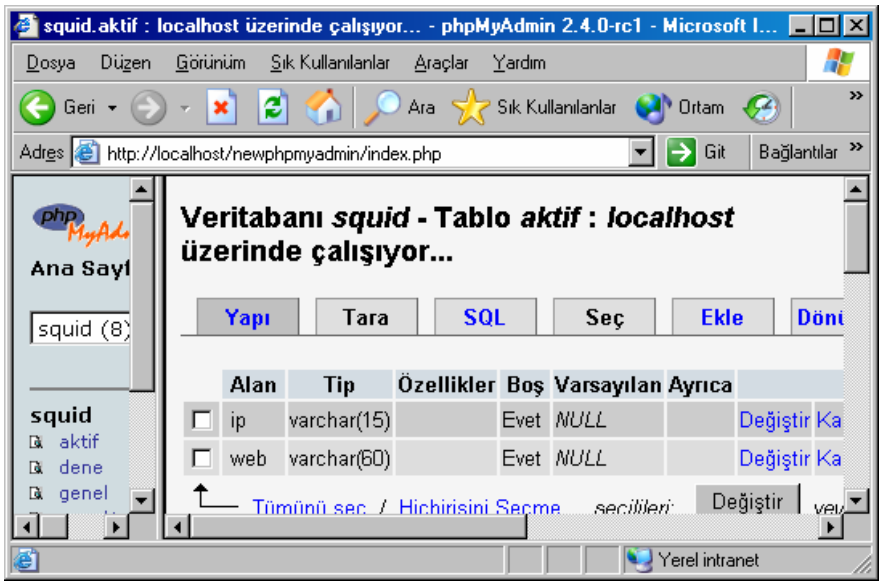
3.2. Online Kontrol (Realtime Site – Kullanıcı Kontrolü)

Bu modül gerçek zamanlı site - kullanıcı kontrolünü gerçekleştirir. Linux'te Squid Proxy komutu sonucu oluşturulan dosyadan ip ve web sitesinin PHP ile pars

edilip, MySQL veritabanında işlenip realtime olarak sorgu sonuçlarının PHP ile webde görüntülenmesinden meydana gelir.

Yani "client mgr:active_requests > $aktifdosya" komutu ile aluşturulan dosyada oluşan veriler realtime PHP ile işlenip veritabanında sorgu sonucu yine realtime webde yayınlanır.

Oluşturulan tablo aşağıdaki şekildedir.

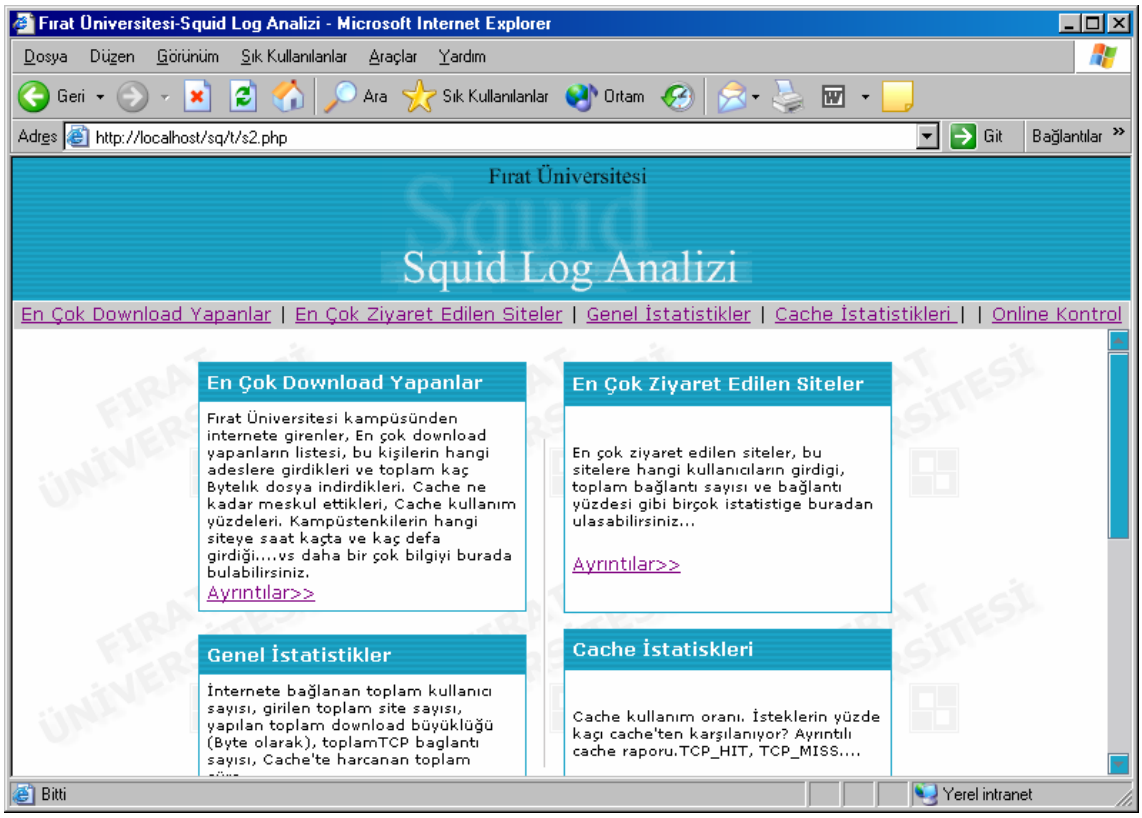


Şekil 7 PHPMyAdmin ile oluşturulmuş "aktif" tablosu

4. SQUİD PROXY LOG ANALİZ PROGRAMININ İNCELENMESİ

Programın modulleri aşağıdadır.

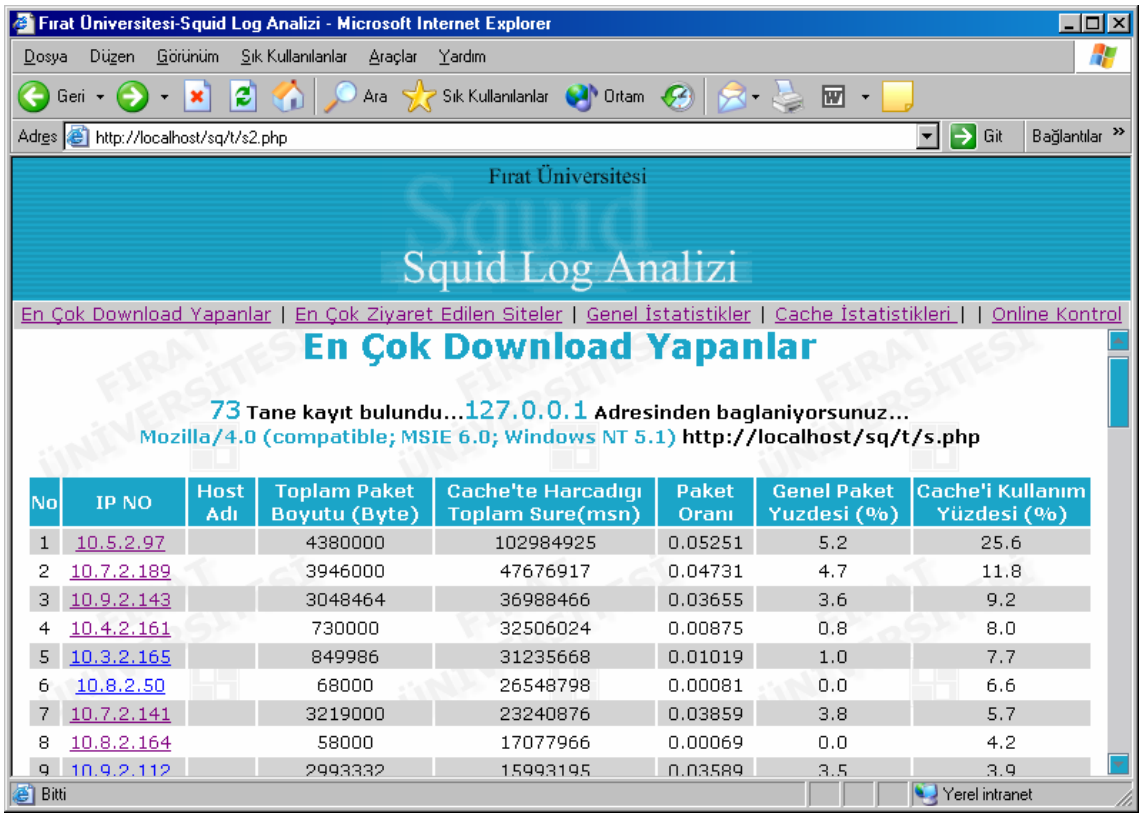
- En çok download yapanlar.
- En çok ziyaret edilen siteler.
- Genel istatistikler.
- Cache istatistikleri.
- Online kontrol.



Şekil 8 Squid log analizi programının giriş sayfası

4.1. En çok Download Yapanlar

Pars işleminden sonra veritabanındaki veriler sorgulanarak aşağıdaki sonuç elde edildi. En çok download yapanlar bulunurken, her bilgisayarın indirdiği veriler toplandı ve bu sonuç bulundu. Cache'te harcadığı zamanlar toplanarak cache'te ne kadar süre harcadığı bulunur. Bulunanlar genel toplama oranlandığı zaman yüzdeler elde edilir.

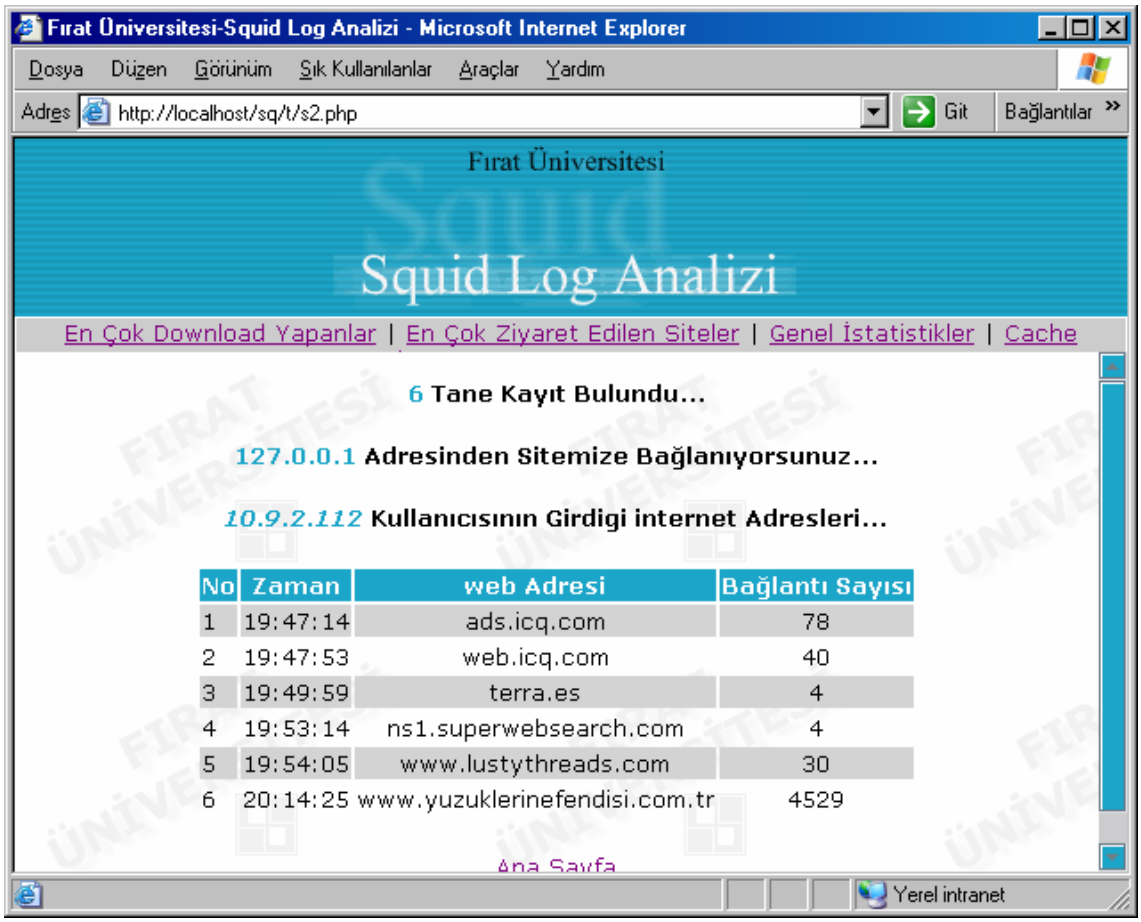


No	IP NO	Host Adı	Toplam Paket Boyutu (Byte)	Cache'te Harcadığı Toplam Sure(msn)	Paket Oranı	Genel Paket Yuzdesi (%)	Cache'i Kullanım Yüzdesi (%)
1	10.5.2.97		4380000	102984925	0.05251	5.2	25.6
2	10.7.2.189		3946000	47676917	0.04731	4.7	11.8
3	10.9.2.143		3048464	36988466	0.03655	3.6	9.2
4	10.4.2.161		730000	32506024	0.00875	0.8	8.0
5	10.3.2.165		849986	31235668	0.01019	1.0	7.7
6	10.8.2.50		68000	26548798	0.00081	0.0	6.6
7	10.7.2.141		3219000	23240876	0.03859	3.8	5.7
8	10.8.2.164		58000	17077966	0.00069	0.0	4.2
9	10.9.2.112		2993332	15993195	0.03589	3.5	3.9

Şekil 9 En çok Download Yapanlar

4.1.1. Girilen siteler

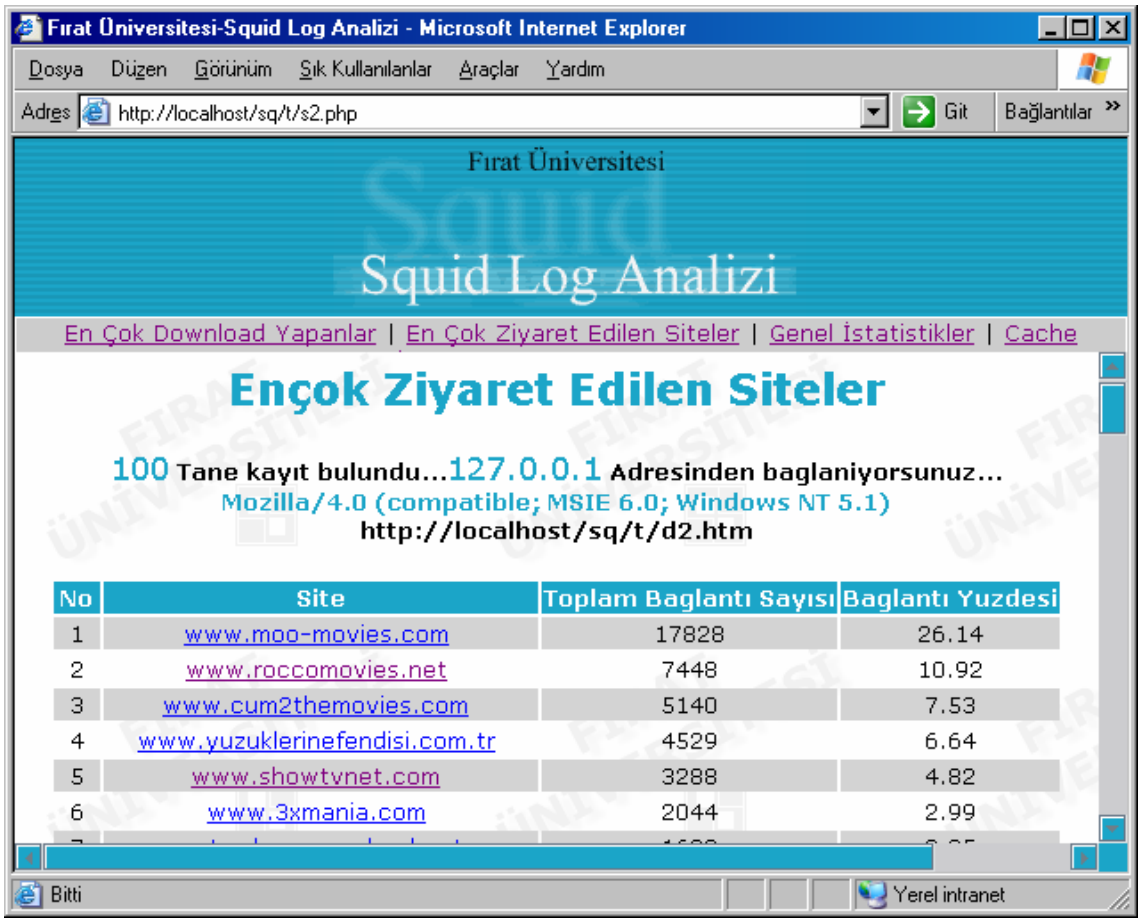
Her ip'nin linkine tıklandığı zaman girdiği siteleri gösterir. Yukarıdaki ençok download yapanlar sayfasında 10.9.2.112 ip numaralı kullanıcının linkine girdiğimiz zaman onun girmiş olduğu siteleri ve siteye kaç bağlantı açtığını bize gösterir. Görünümü aşağıdaki şekildeki gibidir.



Şekil 10 10.9.2.112 IP numaralı kullanıcısının girdiği web siteleri

4.2. En Çok ziyarek Edilen Siteler

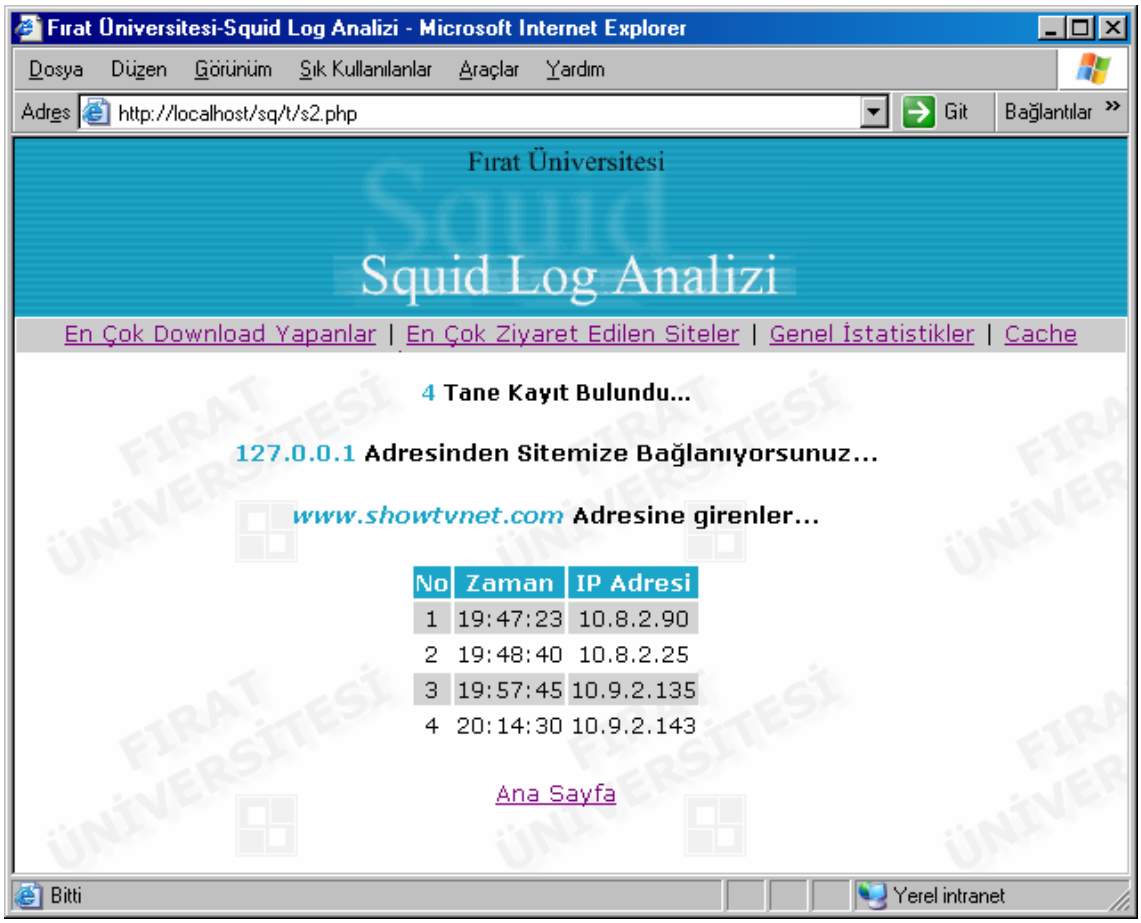
Bulunurken siteye açılan bağlantı sayısı baz alınmıştır. Sorgular sonucunda en çok bağlantı açılan site en çok ziyret ediliyor olarak belirlenmiştir.



Şekil 11 En çok ziyaret edilen web siteleri

4.2.1 Sitelere Kimlerin Girdiği

Sitenin üzerine tıklandığı zaman gelen linkte siteye kimlerin girdiği gözükür. Örnekte www.showtv.net adresine giren kullanıcılar yer almaktadır.



Şekil 12 www.showtv.net adresine giren kullanıcılar

4.3. Genel İstatistikler

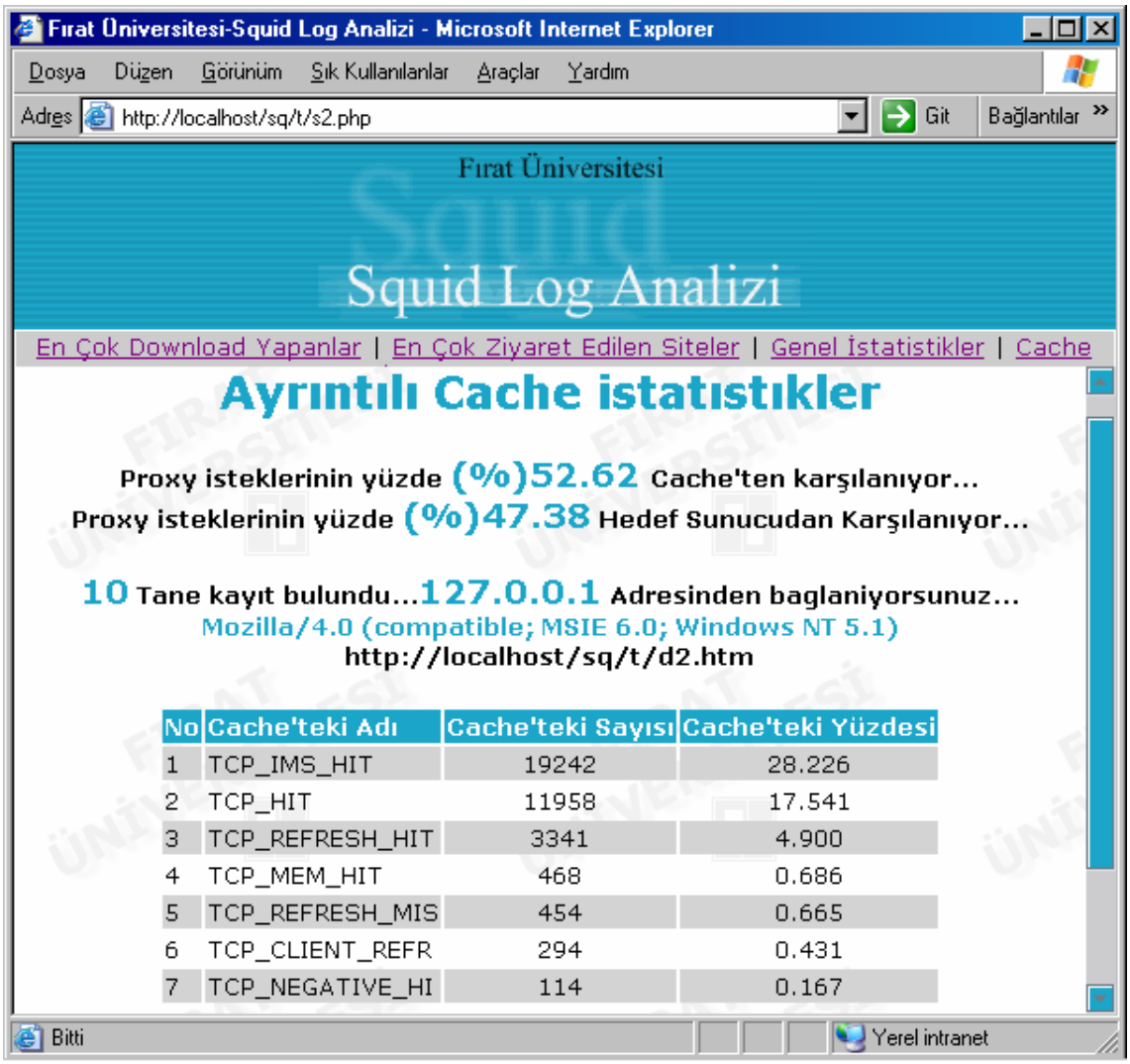
Genel toplamları verir. Sorgular sonucunda oluşan toplamlar burada yer alır.

- Toplam host sayısı
- İndirilen toplam dosya büyüklüğü(Bytes)
- Toplam TCP bağlantı sayısı
- Girilen toplam site sayısı
- Toplam cache süresi(msn)

Şekil 13 Genel istatistikler

4.4. Ayrıntılı Cache İstatistikleri

Cache'teki TCP_MISS, TCP_HIT….vs için ayrıntılı rapor oluşturulur. İsteklerin yüzde kaçı cache'ten karşılanıyor bulunur.



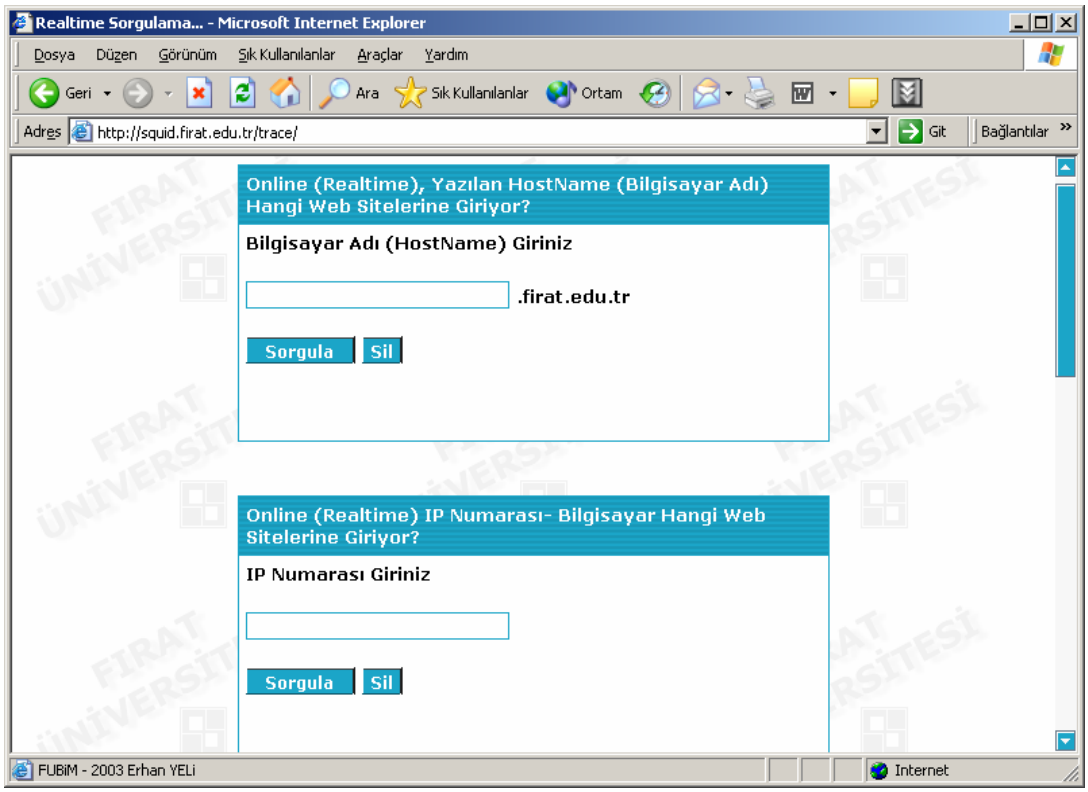
No	Cache'teki Adı	Cache'teki Sayısı	Cache'teki Yüzdesi
1	TCP_IMS_HIT	19242	28.226
2	TCP_HIT	11958	17.541
3	TCP_REFRESH_HIT	3341	4.900
4	TCP_MEM_HIT	468	0.686
5	TCP_REFRESH_MIS	454	0.665
6	TCP_CLIENT_REFR	294	0.431
7	TCP_NEGATIVE_HI	114	0.167

Şekil 14 Ayrıntılı cache istatistikleri

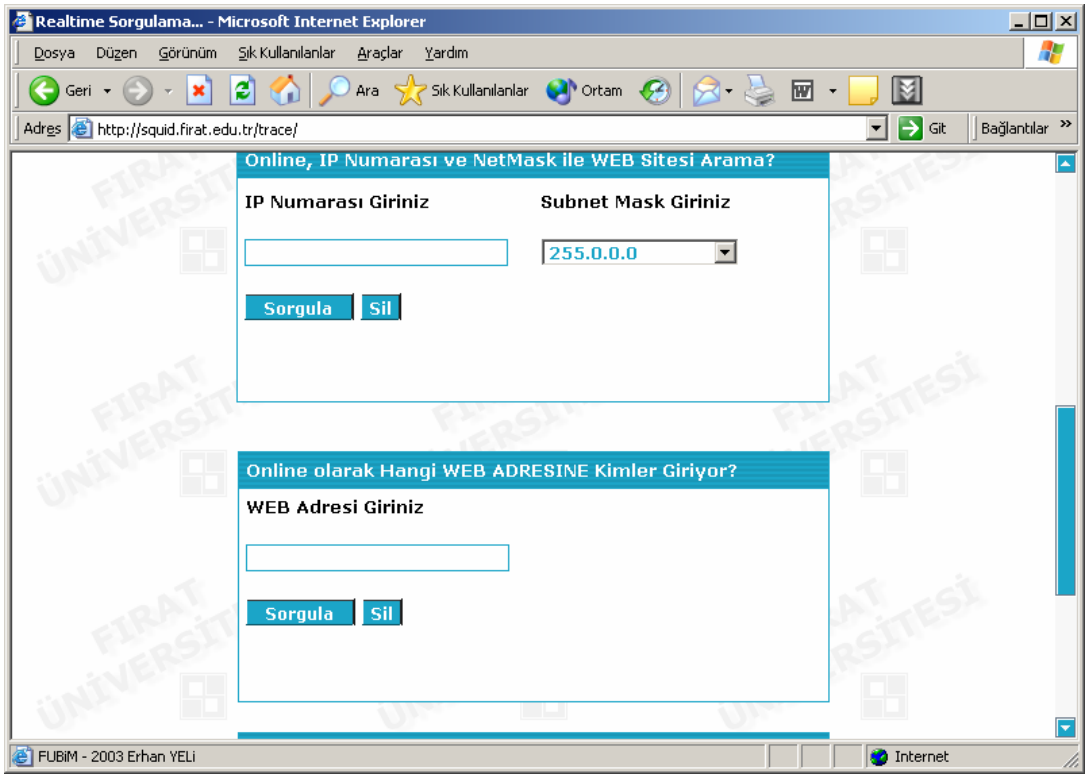
4.5 Online Konrol

Bu modülde realtime kimlerin hangi sitelere girdikleri bulunur. Aramalar aşağıdaki şekildedir.

- Bilgisayar adına göre arama.
- İp numarasına göre arama.
- Alt ağ maskesine göre arama.
- Web adresine göre arama.
- Girilen kelimeye göre arama

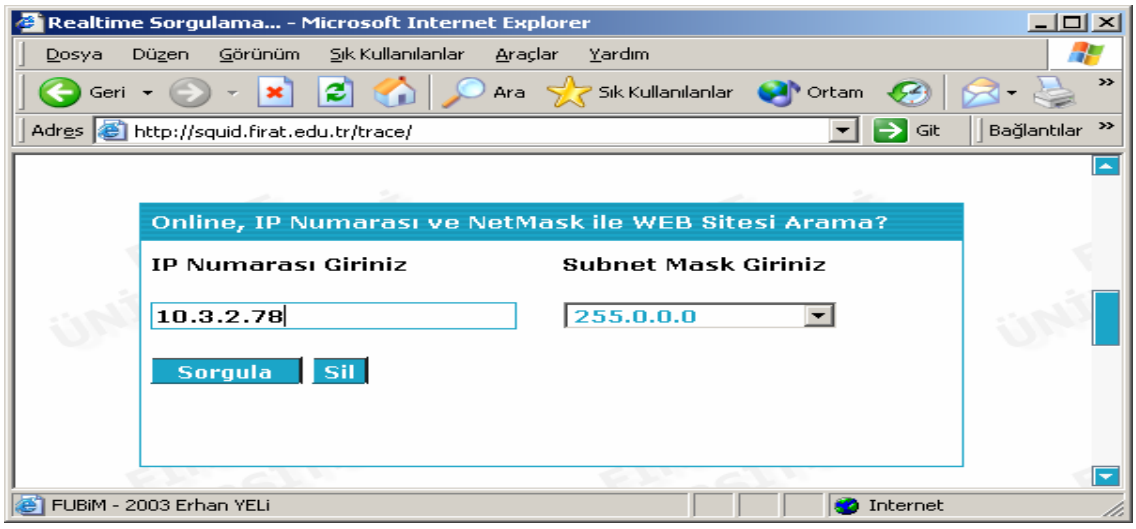


Şekil 15 Online kontrol -1



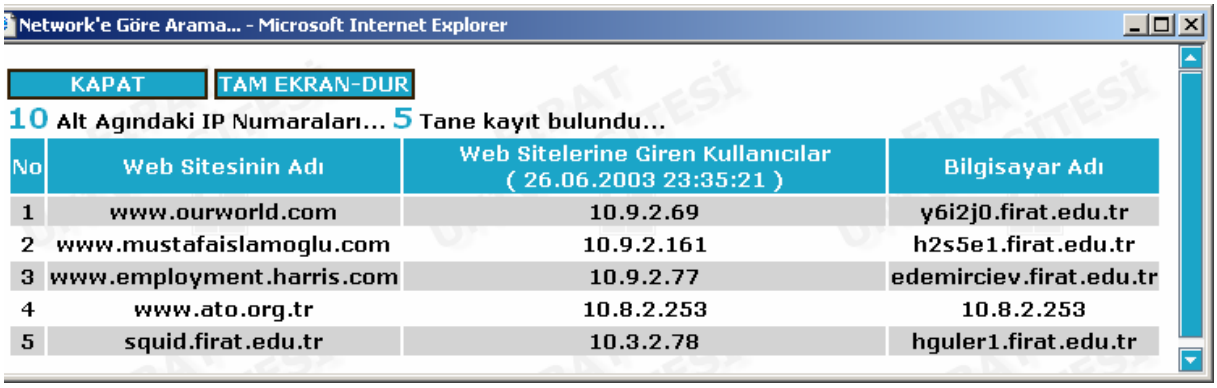
Şekil 16 Online kontrol-2

4.5.1 Alt Ağa Göre Arama



Şekil 17 Online kontrol (Alt ağa göre arama)

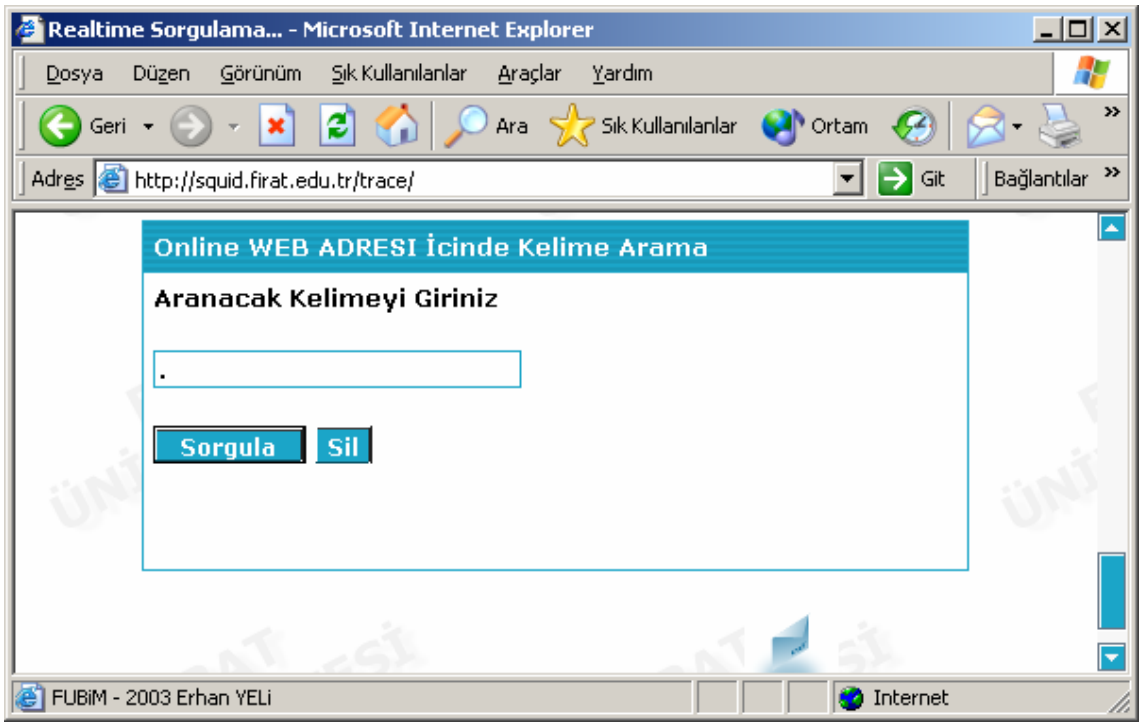
Sorgu sonucunda



Şekil 18 Alt ağa göre arama sonucu

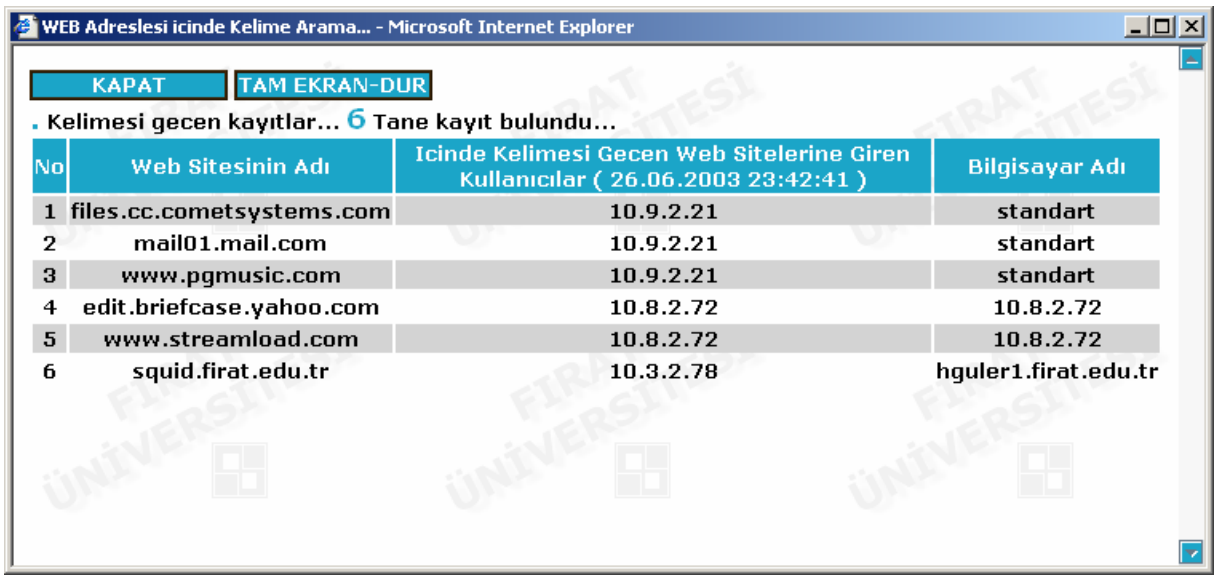
4.5.2 Girilen Kelimeye Göre Arama

Girdiğimiz kelimeye göre arama yapabiliyoruz. Örneğin her web adresinde bulunan "." geçen web adreslerini buluyoruz.



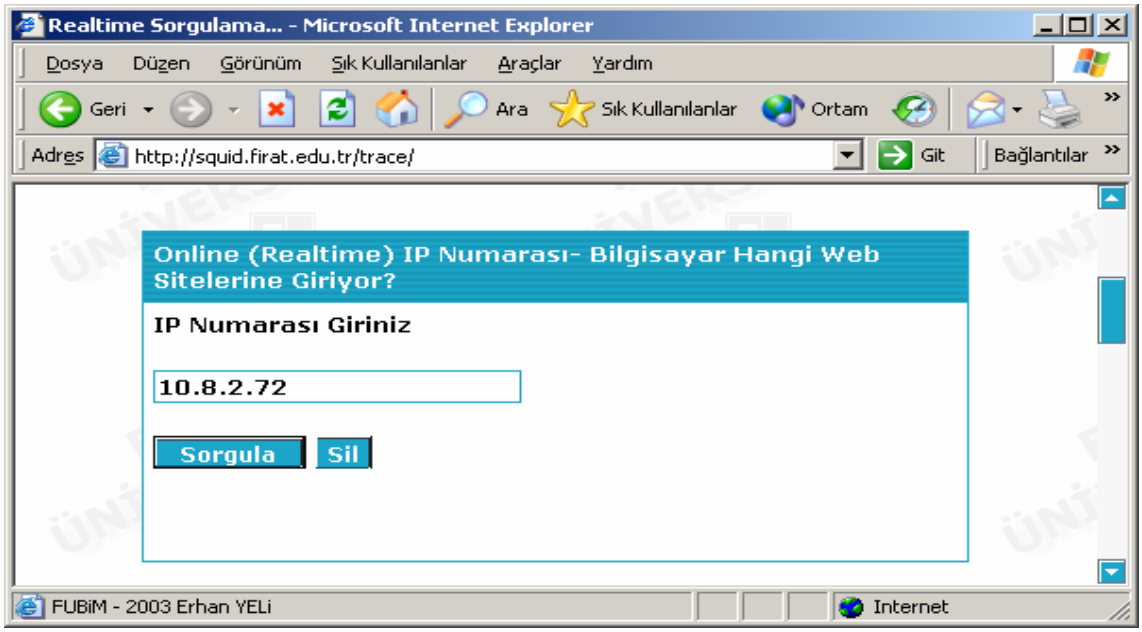
Şekil 19 Girilen kelimeye göre arama

Sorgu sonucunda



Şekil 20 Girilen kelimeye göre arama sonucu

4.5.3 Girilen IP Numarasına Göre Arama



Şekil 21 Girilen IP numarasına göre arama

Sorgu sonucunda

Şekil 22 Girilen IP numarasına göre arama sonucu

5. SONUÇ

Squid Proxy Server'in neden, nerelerde, nasıl kullanıldığını öğrendim. Yerel ağlar için çok performanslı bir Proxy yazılımı. GNU lisanslı olduğu için ücretsiz olarak bulunabiliyor.

İş hayatında yerel bir ağ kurduğumda kullanıcıların internete çıkışını squid Proxy server ile yapmaya çalışacağım. Linux işletim sisteminin ne kadar kullanışlı ve faydalı olduğunu daha iyi anladım.

Bitirme projemde amaçladığım gibi access.log dosyasını PHP & MySQL ile işleyerek kulanıcılar ile ilgili raporlar çıkardım. En çok girilen siteler, en çok dosya indiren kullanıcılır, cache istatistikleri, genel istatistikler, toplam indirilen dosya boyutu, cache kulanım yüzdeleri… gibi verileri elde ettim ve PHP ile internette yayımlayabildim.

Ayrıca Squid Proxy'de realtime web sitesi ve kullanıcı kontrolü yaptım. IP numarası veya bilgisayar adı verilen kullanıcının realtime hangi sitelere girdiği, verilen sitelere realtime kimlerin girdiği, aradığımız kelime ile ilgili sitelere giren kullanıcıları host adları, ip'leri, girdikleri siteler, alt ağ maskesine göre sorgulama yaparak o ağdaki kullanıcıların girdikleri web sitelerini PHP ile webde yayınladım.

KAYNAKLAR

1. www.linux.org.tr
2. www.php.org.tr
3. www.linux-sevenler.de
4. www.belgeler.org
5. www.turk-php.com
6. www.gelecek.com.tr
7. www.zend.com
8. www.mysql.com
9. www.enderunix.org
10. www.w3.com
11. www.squid-cache.org
12. www.fazlamesai.net
13. www.programlama.com
14. www.sorucevap.com
15. www.wrox.com

KİTAPLAR

1. Özgür ÇAYCI, PHP ve MySQL
2. Kayra OTANER, PHP ve MySQL ile Web Yazılımı Geliştirme
3. Görkem ÇETİN, Bilgisayar Ağları ve Linux Ağ Yönetimi
4. T. H. CORMEN, Instruction To Algorithms