

T.C.

GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

140056

DAĞITIK WEB ÖNBELLEK MİMARİLERİ

140056

Kemal YILMAZ

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ

ANABİLİM DALI

TEZ DANIŞMANI

Yrd.Doç.Dr.F. Erdoğan SEVİLGİN

GEBZE

2003



Kemal YILMAZ'ın tez çalışması, G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 16-06-2003 tarih ve .2003-18 sayılı kararıyla oluşturulan jüri tarafından Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS Tezi olarak kabul edilmiştir.

JÜRI

ÜYE : Yrd. Doç. Dr. F. Erdoğan SEVİLGİN
(Tez Danışmanı)

ÜYE : Yrd. Doç. Dr. İbrahim SOĞUKPINAR

ÜYE : Yrd. Doç. Dr. İlyas KANDEMİR

ONAY

G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
..25.07.2003..... tarih ve ..2003/24..... sayılı kararı.

İmza/Mühür

ÖZET

Günümüzde Internet üzerinde bulunan ve ulaşılan bilgi miktarı çok büyüktür ve her geçen gün dramatik olarak artmaktadır. Bu bilgilere ulaşma hızını artırmak için bilinen metotlardan biri gerekli dosyaların bir çok kopyasını dünyanın değişik yerlerinde tutmaktır. Diğer bir metot ise web bilgilerinin daha sonra kullanmak için geçici olarak saklanması sağlayan web önbellek sunuculardır. Web önbellek sunucular tek başına çalıştıkları gibi sunucuların kendi bilgi ve kaynaklarını başka sunucular ile paylaşabilmelerini sağlayan dağıtık web önbellek sunucuları olarak da çalışabilmektedirler. Dağıtık önbellek sunucuların amaçları hizmet süresini, kaynak sunucu üzerinde yapılan iş miktarını, kaynak sunucuya bağımlılığı ve toplam ağ trafiğini azaltmaktır.

Bu tezde çeşitli dağıtık önbellek sunucu sistemlerinin özellikleri ve sistemlerin performans artırımı sağlamak için kullandıkları metotlar incelenmiştir. Her bir sistem istek yönlendirme, yük dengeleme, tutarlılık, ölçeklenebilirlik ve performans kriterlerine göre karşılaştırılmıştır. Ayrıca tezin ikinci bölümünde bu kriterlere dayanarak özgün bir dağıtık web önbellek sunucu tasarımı ve gerçekleştirilmesi yapılmıştır.

SUMMARY

Today, the information accessed and present on the net is found in large amounts and this is increasing dramatically day by day. In order to increase the speed of access to this information, one of the known methods is keeping various copies of the required files in different places on the earth. Another method is the web cache servers which stores web information temporarily in order to use it later. As web cache servers function alone, they can also function as distributed web cache servers which enables sharing its own information and resources with other servers. The aim of distributed cache servers is to reduce the service time, the work done on the resource server, the dependency on the resource server and overall network traffic.

Through this thesis, various methods of the distributed cache server systems for increasing systems performance and properties is evaluated. Each system is compared on the request forwarding, load balancing, stability, scalability and performance criterias. Also in the second part of the thesis, a custom distributed web cache server design and implementation is realized based on these criterias.

TEŞEKKÜR

Tez çalışmam sırasında beni yönlendiren, bana her türlü desteği sağlayan kıymetli hocam Yrd. Doç. Dr. F. Erdoğan Sevilgen'e;

Tez çalışmamı destekleyen ve gerekli izinleri sağlayan TÜBİTAK-UEAKE'ye;

Manevi desteklerini hep yanımda hissettiğim aileme;

Teşekkür ediyorum.



İÇİNDEKİLER DİZİNİ

	<u>Sayfa</u>
ÖZET	İV
SUMMARY	V
TEŞEKKÜR	VI
İÇİNDEKİLER DİZİNİ	VII
SİMGELER VE KISALTMALAR DİZİNİ	X
ŞEKİLLER DİZİNİ	XII
1. GİRİŞ	1
1.1 Dağıtık Önbellek Sistemlerine Genel Bakış	2
1.1.1 Coğrafi Push Önbellekleme	2
1.1.2 Hiyerarşik Önbellekleme	2
1.1.3 Uyarlanabilir	2
1.1.4 Özet	3
1.1.5 CİSİ	3
1.1.6 İpucu-Hiyerarşi	4
2. KRİTERLER	5
2.1 Kriterlere Genel Bakış	5
2.2 Tekil Sunucu	6
2.2.1 İstek Yönlendirme	6
2.2.2 Yük Dengeleme	7
2.2.3 Tutarlılık	7
2.2.4 Ölçeklenebilirlik	8
2.2.5 Performans	8
2.3 Sorgu-Tabanlı	9
2.3.1 İstek Yönlendirme	9
2.3.2 Yük Dengeleme	11
2.3.3 Tutarlılık	12
2.3.4 Ölçeklenebilirlik	12
2.3.5 Performans	13
2.4 Bilgi-Tabanlı	14

2.4.1 İstek Yönlendirme	14
2.4.2 Yük Dengeleme	14
2.4.3 Tutarlılık	16
2.4.4 Ölçeklenebilirlik	17
2.4.5 Performans	18
3. TARTIŞMA KONULARI	19
3.1 Web İçeriği	19
3.2 Eşitlik	19
3.3 Tutarlı Özetleme (Consistent Hashing)	20
4. ANALİZ VE ÖNERİLER	21
4.1 İstek Cevap Süresi	21
4.1.1 Tekil Sunucu	21
4.1.2 Sorgu-tabanlı	22
4.1.3 Bilgi-tabanlı	22
4.2 Sunucu Yükü	23
4.2.1 Tekil Sunucu	23
4.2.2 Sorgu-tabanlı	23
4.2.3 Bilgi-tabanlı	24
4.3 Ağ Trafığı	24
4.3.1 Tekil Sunucu	24
4.3.2 Sorgu-tabanlı	25
4.3.3 Bilgi-tabanlı	25
5. TASARIM	27
5.1 Mimari	27
5.2 Önbellek Yerleştirme	28
5.3 Önbellek Sunucu Çözümleme	29
5.4 Önbellek Bilgi Değişim Protokolü	31
5.4.1 Q sorgusu	32
5.4.2 P sorgusu	32
5.4.3 Y cevabı	32
5.4.4 N cevabı	33

5.5 Önbellek Yer deęiřtirme	33
6. SONUÇ	34
6.1 Performans	34
6.2 Analizler	34
6.2.1 Önbellek Yer deęiřtirme	34
6.2.2 Önbellek Yerleřtirme	35
6.2.3 Hata Toleransı	35
6.2.4 Parametrelerin Etkisi	36
KAYNAKLAR	37
ÖZGEÇMİř	39



SİMGELER VE KISALTMALAR

BGP	Border Gateway Protocol (Sınır Geçit Protokolü)
CGMP	Cooperative Group Management Protocol (Ortak Grup Yönetim Protokolü)
CISI	Cooperative Information Sharing on the Internet (Internet üzerinde Ortak Bilgi Paylaşımı)
CPU	Central Processing Unit (Merkezi İşlem Ünitesi)
CRP	Cache Resolution Protocol (Önbellek Çözümleme Protokolü)
DEC	Digital Equipment Company
FAS	Federation of American Scientists (Amerikan Bilimadamları Federasyonu)
FTP	File Transport Protocol (Dosya Transfer Protokolü)
GB	Gigabayt
HTML	HyperText Markup Language
HTTP	HyperText Transport Protocol
ICP	Internet Cache Protocol (Internet Önbellek Protokolü)

LRU	Least Recently Used (En Seyrek Kullanılan)
NCSA	National Center for Supercomputer Application (Süper Bilgisayar Uygulamaları Ulusal Merkezi)
TTL	Time-to-Live (Yaşam Süresi)
UCLA	Univeristy of California at Los Angeles
URL	Uniform Resource Locator (Düzenli Kaynak Belirteci)
YAA	Yerel Alan Ağı

ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
2.1. Uyarlanabilir Web Önbellek Sistemi	10
5.1. Dağıtık Web Önbellek Sisteminin Genel Mimarisi	27
5.2. Dağıtık Web Önbellek Sunucu Çözümleme	30
5.3. Q sorgusu	32
5.4. P sorgusu	32
5.5. Y cevabı	32
5.6. N cevabı	33



1. GİRİŞ

İnternet günümüzde dünya üzerinde bilgi paylaşmak için kullanılan öncelikli kaynaklardan biri olmuştur. Bu da onun üzerinde bulunan ve ulaşılan bilgi miktarının çok büyük miktarlara ulaşmasına neden olmuştur. İnternetin kıtalararası hızının artması için yapılan altyapı maliyetlerin çok yüksek olması ve rekabet ortamından internete bağlantı fiyatlarının düşmesi, internet hızını artırmak için yapılan yatırımların internetteki bilginin artış hızının gerisinde kalmasına neden olmuştur. Bu yüzden bilgilere altyapı yatırımı yapmadan değişik metotlardan ulaşma konusunda çalışmalar yapılmaktadır. Bunlardan biri olan web önbellek sunucular internet performansını artırmak için yıllardır kullanılmaktadır. Bu sunucular kullanıcılar tarafından istenilen popüler bilgilerin her istendiğinde kaynak sunucudan getirilmek yerine kullanıcılara daha yakın yerlerde tutulmasını sağlarlar. Web önbellek sunucuların kullanılmasının amaçları bekleme süresinin, sunucu yoğunluğunun ve ağ trafiğinin azaltılmasıdır.

İnternetin doğasından kaynaklanan dağınıklığı, önbellek sunucuların ağdaki diğer kaynaklardan bağımsız, tek başına çalışması yerine, bir önbellek sunucunun diğer önbellek sunucuların kaynaklarından faydalanması ihtiyacını doğuruyor. Dağıtık önbellek sunucu sistemleri, bir çok önbellek sunucunun coğrafik ve donanımsal şartları göz önünde bulundurarak bilgi ve kaynaklarını birbirleriyle paylaşmalarını sağlayan sistemlerdir. Bu önbellek sunucularının depolama özellikleri arasında koordinasyon sağlama ve depolanmış bilgilerin birbirleriyle paylaşılmasını sağlama sayesinde her isteğin hedef sunucudan getirilmesi engellenmiş olur. Dağıtık önbellek sistemleri performans ve ölçeklenebilirlik gibi konularda tekil çalışan sunuculardan üstünlükleri vardır.

1.1 Dağıtık Önbellek Sistemlerine Genel Bakış

1.1.1 Coğrafi Push Önbellekleme

Coğrafi Push Önbellekleme sisteminde [12] kullanıcıların isteklerine cevap vermesi gereken birincil (master) sunucu vardır. Birincil sunucu istenilen bilginin diğer sunucularda var olup olmadığını kontrol edilmesinden sorumludur. Bu işlemi yapabilmek için birincil sunucunun bütün isteklerin adreslerini ve isteklerin yoğunluğunu tutması gerekmektedir. Eğer bir istek belli bir zaman için cevaplanmazsa, birincil sunucu istenilen sayfayı doğrudan kaynak sunucudan getirir. Birincil sunucu kaynak sunucudan getirdiği bilgiyi depolamak isteyecek bütün önbellek sunucuların bilgisini tutar. Yenilenen bilgiyi bütün bu sunuculara gönderir (push) ve gönderirken bu sunuculardaki boş alan miktarı ve yük gibi kriterlere bakar.

1.1.2 Hiyerarşik Önbellekleme

Hiyerarşik önbellek sistemler[6], önbellek sunucuların hiyerarşi ağacı şeklinde tasarlanması ile oluşturulmuştur. Bir önbellek sunucudan herhangi bir istekte bulunulduğunda, öncelikle kendi deposunda var olup olmadığını kontrol eder. Eğer kendisinde yoksa aynı seviyede bulunan diğer önbellek sunucular ve hiyerarşide bir üstündeki sunucuya isteği aktarır. Önbellek sunucu bilgiye sahip olduğunu bildiren ilk sunucudan bilgiyi alır. Eğer hiç bir önbellek sunucu bu bilgiye sahip değilse veya belli bir zaman içinde cevap gelmemişse önbellek sunucu bu bilgiyi doğrudan kaynak sunucudan alıp depolar.

1.1.3 Uyarlanabilir

Uyarlanabilir [9] önbellek sistemlerinde, iç içe geçen önbellek sunucu grupları

vardır. Bu önbellek grupları ağda meydana gelen değişikliklerde otomatik olarak kendilerini konfigüre edebilirler. Dinamik olarak grupları oluşturmaları uyarlanabilir web önbellek sisteminin hiyerarşik sistemden ayıran en önemli faktörlerden biridir. Bu gruplar bilinen multicast adreslerine oylama mesajı göndererek oluşturulurlar. Bu oylama sinyalinin gücüne göre her bir önbellek sunucu o grubun içinde olup olmasına gerektiğine karar verir. Periyodik olarak her grup kendi bilgilerini komşu gruplarla paylaşarak hangi önbellek sunucunun hangi bilgiye sahip olduğunu öğrenir. Bir önbellek sunucu kendinden istenen bilgiyi bulamadığında kendinde olan en son bilgilere bakarak kendi grubundaki hangi önbellek sunucusunun bu bilgilere sahip olabileceğini bulur.

1.1.4 Özet

Özet önbellek sistemlerinde [2], her bir önbellek sunucuda diğer önbellek sunuculardaki bilgilerin özet tablolarının bulunduğu bir dizin tutulur. Özet önbellek sisteminde, sunucular gelen istekleri kendilerinde bulunan özet tablolarına göre yönlendirirler. Performansı artırmak için bu özet tabloları sıkıştırılmış veri yapılarında tutulur ve belli aralıklarla güncellenir.

1.1.5 CISI

CISI (Cooperative Information Sharing on the Internet) sisteminde [5], önbellek sunucular aralarındaki yakınlık bilgisi kullanarak bir iletişim çizgesi oluşturur ve bu sayede önbellek sunucular birbirlerine en kısa yoldan bağlanır. Her bir sunucu komşularının bilgilerini içeren bir dizin tutar. Bu sistemde komşu demek iki sunucu arasının eşik uzaklıktan daha yakın olması demektir. Eğer bir istemci sunucudan istekte bulunursa, bu sunucu bütün sistemlerde olduğu gibi önce kendi bilgilerini kontrol eder. Eğer istenilen bilgiyi bulamazsa daha sonra komşularının bilgilerini içeren dizinden arar. Eğer obje orada bulunursa, sunucu bilgiyi dizinde ismi bulunan önbellek sunucudan ister. Dizinde de bulamazsa doğrudan bilgiyi kaynağından ister. Her iki durumda da

önbellek sunucu bu bilgiyi kendi deposuna da ekleyebilir.

1.1.6 İpucu-Hiyerarşi

İpucu-Hiyerarşi sisteminde [7], her bir önbellek sunucu bilgilerin bulunduğu yerleri gösteren ipuçlarını içeren bir tablo tutar. İstemciler isteklerini hiyerarşinin en altında bulunan önbellek sunuculara gönderirler. Eğer istenilen bilgi bu önbellek sunucuda ise, sunucu bilgiyi hemen kullanıcıya döner. Eğer yoksa, önbellek sunucu kendi ipucu tablosuna bakarak bu bilginin yerini tespit etmeye çalışır. Eğer ipucu tablosunda bulabilirse isteği doğrudan listedeki önbellek sunucuna yönlendirir. Bilgi ipucu listesinde yoksa önbellek sunucu bilgiyi doğrudan kaynağından getirir.

2. KRİTERLER

Yukarıda anlatılan dağıtık önbellek sistemlerini karşılaştırmak için öncelikle bu sistemler tekil-sunucu, sorgu-tabanlı sistemler, bilgi-tabanlı sistemler olmak üzere üç ana gruba ayrılacaktır. Coğrafik push önbellekleme sistemleri tekil-sunucu grubuna aittir. Hiyerarşi ve uyarlanabilir dağıtık önbellek sistemleri ise sorgu-tabanlı önbellekleme sistemleri grubundadır. Bilgi-tabanlı sistemler ise özet, CISI ve ipucu-hiyerarşi dağıtık önbellek sistemleridir. Bu bölümde üç sistem için istek yönlendirme, yoğunluk dengeleme, tutarlılık, ölçeklenebilirlik, ve performans kriterleri incelenecektir. Bu beş kriter haricinde bir çok faktör dağıtık web önbellek sistemlerini etkilemesine rağmen bu beş kriter sistemler arasındaki ana farkları göstermek için yeterlidir. Bunların haricindeki diğer faktörler ise 3. bölümde tartışılacaktır.

2.1 Kriterlere Genel Bakış

Önbellek sistemlerinde istek yönlendirme protokolü istenilen bilginin depolanan önbellek sunucudan istemciye yönlendirilmesi yoludur. Bütün sistemler önbellek sunucuda aranan bilginin bulunmaması durumunda bilgilere ulaşmak için bir metot kullanırlar. Eğer bu önbellek sunucu dağıtık olmayan bir önbellek sunucu ise, bilgiyi doğrudan kaynak sunucudan getirir. Fakat dağıtık önbellek sunucular, bu bilgiyi doğrudan kaynak sunucudan getirmek yerine yakınlarında bulunan diğer sunuculardan getirme avantajından yararlanmak isterler.

Dağıtık önbellek sistemlerinin amaçlarından biri kaynak sunuculardaki yük miktarını azaltmak olduğu gibi, bir diğer amacı da sistem içinde bulunan önbellek sunucular arasındaki yük miktarını dengelemektir. Yük dengelemek, ağ trafiğini azaltmak ve sistemdeki istekleri her yere eşit dağıtmayı içerir. Bir diğer etken ise sistemin çok fazla istenilen bilgileri getirmek için yapılan istekleri nasıl dengelediğidir.

Dağıtık önbellek sistemlerinde tutarlılığın iki yönü vardır; bilgilerin ve meta-datanın tutarlılığı. Dağıtık veya dağıtık olmayan bütün önbellek sunucular bilgilerin

tutarlılığından emin olmalıdır. Eğer önbellek sunucu deposunda bir bilgiye sahipse ve kaynak sunucu bu bilgiyi değiştirmişse, önbellek sunucunun bilgisini değiştirmesi gerektiğini bilmesi gerekir. Yoksa kendisindeki bilgiyi yenileyene kadar istemcilerle eski bilgiyi vermek zorunda kalır. Diğer tutarlık ise meta-data tutarlılığıdır. Bir çok sistem tablolar, dizinler veya başka kaynaklar ile sistemdeki sunucuların durumu hakkında bilgi tutar. Eğer bir sunucudaki bir bilgi veya bilginin durumu değişirse diğer sunucuların bundan haberi olması gerekir.

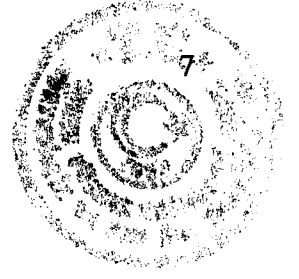
Dağıtık önbellek sunucu sistemlerinin göz önünde bulundurulması gereken en önemli kriterlerinden biri de internette meydana gelen değişikliklere uyumluluğu ve ölçeklenebilirliğidir. Ölçeklenebilirlik, sistemin büyümesinin ve bu büyümenin performansa olan etkisinin incelenmesi ile ilgili bir ölçüttür. Internette bu gelişim devam ettikçe sistemlerin ölçeklenebilirlik konusunda bir takım çözümleri olması gerekir.

Sonuncu olarak incelenecek kriter ise performanstır. Dağıtık web önbellek sistemlerinin performansı bekleme süresi, yük dengesi, ağ trafiği ve ölçeklenebilirlik gibi bir çok etkenin birleşmesinden oluşmaktadır.

2.2 Tekil Sunucu

2.2.1 İstek Yönlendirme

Tekil sunucu yönlendirme sisteminde, birincil sunucu bütün istekleri yönlendirir. Bu sistemin bilgileri doğrudan kaynak sunucudan getireceği anlamına gelmez. Fakat birincil sunucu gelen isteklerin uygun olan sunucuya gönderilmesinden sorumludur. Bu sistemde, birincil sunucu ağ yapısı ve önbellek sunucuların yapısını bilmektedir. Tekil sunucu sisteminde isteklere nasıl cevap verileceğine karar vermeye çalışır. Bunun için kullanılan teknik ise Blaze [3], tarafından ağ trafiğini azaltmak için geliştirildi. Bu metotta istemci sunucuya normal bir web sunucu gibi bağlanır, daha sonra bu sunucu kullanıcıyı en yakın sunucuya yönlendirir. Bu metot istemcinin birincil sunucudan en yakın sunucuyu sorduğu için “0-800” tekniği olarak da bilinir.



2.2.2 Yk Dengeleme

Tekil sunucu sistemleri, nbellek sunucularındaki yk ve aē tafiēini azaltarak yk dengelenmesi yapar. Bir bilgi iēin belirtilen eēik sresi aēılırsa sunucu bilgiyi baēka bir nbellek sunucuya kopyalamaya ēalıēır. Her bir nbellek sunucu son bir saat iēinde yapılan isteklerin sayısını yk dengeleme faktr olarak tutar ve bilgi isteēi tamamlandığında yk bilgisini artırır. Ayrıca nbellek sunucu her saat yeni yk bilgisini oluēturur. Eēer sunucu eēik deēerini geēerse, kendisine ait en popler dosyaları baēka bir sunucuya gnderir ve kendine ait yk bilgisini sıfırlar. Bu metot sayesinde popler sunucular kendilerine gelen yk miktarını daha az popler sunuculara aktarmıē olurlar.

Fazla yklenmiē olan nbellek sunucu, elindeki bir bilginin koyulacaēı nbellek sunucuyu bulmak iēin elindeki kayıttaki bulunan nbellek sunucular listesinden seēer. Bu sisteme gre bilgilerinin kopya sayısı poplerlik oranlarıyla doēru orantılıdır. Bilgilerin ne zaman ve neredeki nbellek sunucuda depolanacaēına karar veren sunucular, ayrıca aē tafiēini azaltmak iēin bilgileri kendilerine en yakın sunuculardan getirir. Bunun iēin nbellek sunucular aē mimari bilgisini ve bilgilerin ulaēım geēmiēlerini kullanılırlar. Sonuē olarak, yk dengelemek iēin aynı anda popler bir bilginin birden fazla kopyası tutulur.

2.2.3 Tutarlılık

Tekil sunucu nbellek sistemleri bilgi tutarlılığını saēlarlar. Eēer birincil sunucudan bir istekte bulunulursa birincil sunucu nce kaynak sunucudan elinde olan bilginin gncel olup olmadıēını sorgular, eēer bilgi gncel deēil ise kaynak sunucudan yeni olan bilgiyi alır. Ayrıca bu sistemler bilgilerin gncel olup olmadıēına karar vermek iēin belli bir eēik deēeri tutarlar, bilginin kaynak sunucudan en son kontrol edildiēi tarih deēeri bu eēik deēerini aēmıēsa bilgi geēersiz sayılır.

2.2.4 Ölçeklenebilirlik

Tekil sunucu sistemlerinin ölçeklenebilirliği iki ön kabule dayanır. Birincisi, bilgileri kopyalayan her sunucu aynı zamanda kendi üzere kopyalanan bütün bilgileri de kabul eder, diğeri ise sunucular eşit dağılmıştır. Bu sistemde sunucular iş yükü miktarlarını artıracığını düşünürlerse kendilerine gönderilen her bilgiyi kopyalamayı kabul etmeyebilirler. Aynı zamanda bilgilerin sistem üzerinde eşit dağıtılıp dağıtılmadığının kontrol edilmesi de fazladan yapılan bir iştir. Buna karşın bu sistemde bilgileri koyacak yeni bir sunucuyu sisteme eklemek kolaydır. Böylece birincil sunucular popüler olan bilgileri daha az popüler olan sunuculara kaydırabilirler.

Tekil sunucu sisteminde ölçeklenebilirliği etkileyen bir diğer sorun ise kayıt bilgileridir. Kayıt bilgisi, kullanılabilen bütün önbellek sunucularının veritabanıdır ve internet üzerinde diğer önbellek sunuculardan gelen istekleri kotarması gerekir. Kayıt bilgisinin ölçeklenebilir olması için milyonlarca kayıt tutabilmelisi ve güvenilir olması gerekmektedir, bunu sağlamak için dağıtık olarak tutulur. Buna karşılık, kayıt bilgisinin kesin bir tutarlığa sahip olması gerekmez. Çünkü kayıt bilgisi, bu listeden rasgele olarak sunucu seçer ve bunların hepsinin kullanılabilir olmasına gerek yoktur.

2.2.5 Performans

Tekil sunucu sistemlerinin hangi bilgileri nereye koyacaklarını seçebilme avantajları vardır. Bu sistemler az sayıda popüler dokümanı olan meşgul sunucular için uygundur [11].

Bu sistem ile geleneksel istemci-önbellekleme ile karşılaştırılmıştır. İstemci-önbellekleme, Gwertzman [11] tarafından, istemci herhangi bir dokümanı ister ve bir daha aynı dokümanı istemez şeklinde simüle etmiştir. Bu yaklaşım sayesinde simülasyonu basitleştirmek için önbellek sunucularını büyüklüğünün sonsuz olduğu kabul edilir. Gwertzman iki farklı kaynak kullanmıştır: NCSA kaynağı ve FAS kaynağı.

İstemci-önbellekleme ile tekil sunucu önbellekleme sistemi NCSA üzerinde karşılaştırırken, her iki metodun da eşit performansa sahip olduğunu ve her ikisinin kombinasyonunun kullanmasının ağ bant genişliği kullanımının azaltılmasını için en efektif yol olduğunu belirlenmiştir. FAS üzerinde denediğinde ise istemci-önbellekleme çok daha başarılı olduğunu görür, çünkü FAS yakın çevresinde çok popüler olan bir sunucudur, isteklerin çoğu yakın makinelerden gelmektedir. Sunucunun bir özelliği de isteklerinin bir çoğunu alan az sayıda dosya bulunmaması, bu yüzden çekirdek dosyaları dağıtamamaktadır. İstemci-önbellekleme ve tekil sunucu önbellekleme her iki sistemde de eşit bant genişliği almışlardır, fakat farklı bilgileri etkilemişlerdir. Bu yüzden bunların kombinasyonunun kullanılması en fazla verimliliği sağlamıştır. Bu simülasyonda bütün internet gezginlerinin istemci-önbellekleme özelliğine sahip olması gerektiğini göstermiştir. Ayrıca, tekil sunucu sistemlerinin, kaynak sunucunun yükünü ve ağ bant genişliği miktarını azalttığı da görülmüştür.

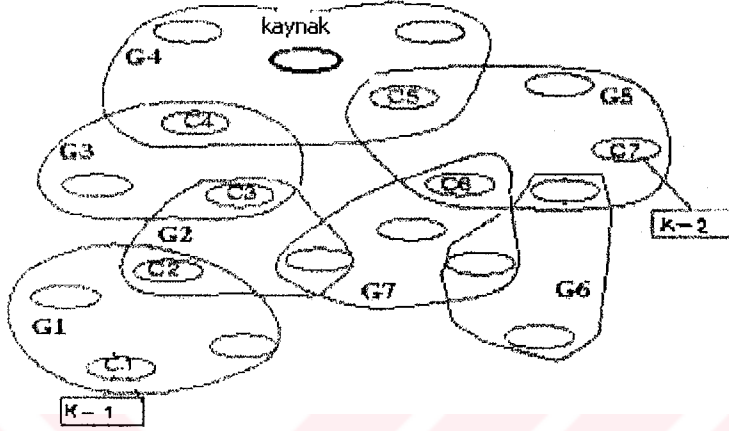
2.3 Sorgu-Tabanlı

2.3.1 İstek Yönlendirme

Sorgu tabanlı sistemlerde her bir sunucu kendi içinde aradığı bilgiyi bulamazsa, istek diğer önbellek sunuculara yönlendirilir. Bu metodu kullanan sistemlere örnek olarak hiyerarşik önbellek sistemi verilebilir. Hiyerarşik önbellek sisteminde istek yönlendirmenin amacı istek için geçen sürenin azaltılmasıdır. Eğer bir istek aranan sunucuda bulunamadıysa, o istek sunucunun çocuklarına, ebeveynine ve kaynak sunucuya yönlendirilir. Her bir sunucu “bulundu” veya “bulunmadı” bilgisini ve aranan bilgi kaynak sunucuda ise o da “bulundu” bilgisini döner. İlk “bulundu” bilgisini gönderen sunucudan bilgi istenir. Eğer bütün önbellek sunucularda bu bilgiye sahip değilse, bilgi kullanıcıya en yakın önbellek sunucu tarafından doğrudan kaynak sunucudan getirilir.

Sorgu-tabanlı önbellekleme grubuna giren bir diğer sistemde uyarlanabilir web önbellekleme sistemidir. Bu sistemde önbellek sunucular kendi aralarında gruplar

oluştururlar. Uyarlanabilir önbellek sisteminde, bütün gruplar kendi kendini organize ederler ve birbiri üzerinde geçen gruplar oluştururlar. Bu gruplar şartların değişmesinde kendilerini otomatik olarak uyarlarlar.



Şekil 2.1: Uyarlanabilir Web Önbellek Sistemi

Şekil 2.1, uyarlanabilir web önbellek sistemi ve istek yönlendirme algoritmasını göstermektedir. K-1 yeni bir sayfa ister ve bu istek C1 önbellek sunucusuna ulaşır, eğer C1 istenen sayfaya sahip değilse, isteği kendi grubu olan G1 grubuna multicast yapar. İstek G1 grubu içinde de bulunamazsa C2 dahil olduğu diğer grup olan G2'nin kaynak sunucuya daha yakın olduğunu anlar. Bunun üzerine C2 isteği G2 grubuna multicast yapar ve istek bu metot kullanılarak yönlendirilmeye devam eder. Bu arama isteği önbellek gruplar tarafından kaynak sunucuya yönlendirmesiyle veya bu bilgiye sahip olan bir önbellek sunucunun bulunması ile son bulur. İstek cevaplanacağı zaman aranan bilgiye sahip olan sunucu cevabı bütün gruba multicast yapar, böylece isteğin istemciye daha hızlı ulaşmasını sağlar. Eğer K-2 kullanıcısı K-1 kullanıcısından sonra aynı sayfayı isterse, C5 aranan bilgiye sahip olmalıdır. Kaynak sunucudan bilginin istenmesine gerek yoktur, çünkü istek daha önce G5 grubuna multicast yapılmıştır.

2.3.2 Yük Dengeleme

Sorgu-tabanlı önbellek sistemlerinde yük dengelemek için sorgu isteklerinin bütün sistem içinde nasıl dağıtıldığına bakılır. Hiyerarşik önbellek sisteminde, her bir önbellek sunucu başına yük miktarını azaltmak için en alt seviyedeki önbellek sunucular birbirleriyle yardımlaşırlar. Yoğun istekle karşılaşan önbellek sunucuların yükleri başka sunuculara dağıtılır. Bu sistemlerde yüksek performanslı sunucuların önbellek hiyerarşisinin yüksek seviyelerinde kullanılması önemlidir, çünkü bu sunucular çok fazla istek miktarıyla karşılaşmaktadır.

Uyarlanabilir web önbellek sistemleri yük dengeleme ile ilgilenen bir başka sorgu-tabanlı önbellek sistemidir. Bu sistemlerde hiyerarşik önbellek sistemlerine benzer metotlar kullanılmaktadır. Bu sistemler ebeveynlerinden, çocuklarından ve kaynak sunucudan sorgulamak yerine, sadece kendi gruplarına multicast yaparlar. Bu sistemde yük dengelemek için ağ yoğunluğu veya önbellek sunucusunun kendinde bulunan bilginin popülerliğinin artması temel alınır. Bu bilgiler önbellek grup içinde dağıtılır ve bu gruplar yeni duruma göre kendilerini yeniden konfigüre ederler.

Uyarlanabilir önbellek sunucuların en önemli konusu grupların kendi kendilerini konfigüre etmesidir. Uyarlanabilir önbellek sunucular bu işlemi yapmak için Önbellek Grup Yönetim Protokolünü (CGMP) [9] kullanırlar. CGMP her hangi bilgi bulunamadığında bilgiyi bulmak için harcanacak hop sayısını azaltmak için önbellek gruplarını ayarlar. CGMP bir önbellek sunucu bir gruba dahil olduğunda grubun konfigürasyonunu bilmesini sağlar ve bunun için oylama sistemini multicasting kullanarak gerçekleştirir. Önbellek sunucular grup üyelerine, gruba üye olup olamayacağını sorar ve grup buna cevap döner. Bir başka deyişle, yeni bir sunucu sisteme eklendiğinde grup üyelerine bir oylama mesajı gönderir ve diğer sunucular önbellek sunucu ile olan uzaklıklarını göz önünde bulundurarak bir cevap verirler. Önbellek sunucuların verdikleri cevaplar toplanır ve bu sonuç belli bir eşik miktarından fazla ise önbellek sunucu gruba dahil olur. Grupları dinamik olarak ayarlamak için grup üyeleri belli aralıklarla birbirlerine oylama mesajları gönderirler.

2.3.3 Tutarlılık

Sorgu-tabanlı önbellek sistemleri bilgi tutarlılığını göz önünde bulundurmışlardır. Bu sistemler bilgilerin başlıklarında belirtilen bilgilere göre çalışan yaşama-süresi (TTL) metodu sayesinde bilgi tutarlılığını sağlarlar. TTL üç metot üzerine çalışır; Birincisi bilgi geçerlilik süresinin sonu ile ilgili bir bilgiye sahipse, sistem buna uyar; İkincisi bilgi son güncelleme zamanı bilgisini içeriyorsa, sistem bilginin TTL'sini daha önceki değişikliği ile son değişikliği arasında geçen sürenin yarısına eşitler; Üçüncüsü ise bilgi herhangi bir zaman bilgisine sahip değilse, sistem bilginin adresinin tipine (HTTP, FTP, ..) göre varolan TTL değerini bilgiye atar. Sonuç olarak eğer bilginin TTL süresi dolarsa, önbellek sunucu bilgiyi geçersiz sayar.

2.3.4 Ölçeklenebilirlik

Sorgu-tabanlı sistemlerdeki ölçeklenebilirliği etkileyen en önemli etkenlerden biri multicastingdir. Hiyerarşik sistemlerde, multicasting kullanıcının bir bilgiyi çocuklarında, ebeveynlerinde ve kaynak sunucuda sorguladığında oluşur. Hiyerarşik sistemi gerçekleştirmek için geliştirilen Harvest önbellek[6] projesi haberleşmek için internet önbellek protokolünü (ICP)[16] kullanmaktadır. ICP protokolünün ölçeklenebilirlik ile problemi vardır, eğer önbellek sunucuların sayısı artarsa fazladan yapılan iletişim ve hesap miktarı bununla doğru orantılı olarak artmaktadır [2].

Uyarlanabilir web önbellek sistemleri ise ölçeklenebilirlik problemi için önbellek grupları içinde gizli bir hiyerarşi kurarak çözümlemeye çalışmışlardır. Birçok sistem bütün önbellek sunucuların sakladığı bilgi ile ilgilenirken, uyarlanabilir önbellek sistemi bütün sistemdeki sunucuların hangi sayfaların saklandığı konusunda ilgilenmez, sadece kendi grubundaki önbellek sunucularla ilgilenir. Uyarlanabilir önbellek sunucularının ölçeklenebilirlik problemi ise maksimum hop sayısı ile ilgilidir ve bu da önbellek ağının çapı ile doğru orantılıdır.

Uyarlanabilir önbellek sistemi isteklerin popülerliğini temel alır. Popüler olan bilgiler birçok grup içindeki bir çok önbellek sunucuda bulunurlar. Popüler olmayan bilgiler ise kaynak sunucuya daha yakın olan önbellek sunucularda saklanır. Bu tasarım önbellek sisteminin yoğun sunuculara karşı kendini uyarlamasını sağlar. Kendi kendini konfigüre edebilme özelliği uyarlanabilir önbellek sistemin ölçeklenebilirliği için en önemli yardımcıdır.

2.3.5 Performans

Sorgu-tabanlı sistemlerden olan hiyerarşik önbellek sistemlerin de performansı en çok etkileyen sebeplerden biri önbellek sunucu yüküdür. Hiyerarşide yukarıda olan sunucular altta bulunan sunucuların ihtiyacı olan bilgileri getirmek zorundadırlar. Eğer alttaki önbellek sunuculardaki başarı oranı yüzde elli ise, gelen isteklerin yarısı kaynak sunucu yerine bir üst seviyedeki sunucuya gönderiliyor demektir. Bu metotla, hiyerarşinin üstünde bulunan sunucular çok büyük yoğunluk ile karşılaşacağı için isteklerinde hız artışından daha çok hız yavaşlamasına neden olacaktır.

Hiyerarşik önbellek sistemlerinde iletişim problemlerini çözmek için en temel metot broadcastingdir. Hiyerarşik önbellek sisteminde, eğer önbellek sunucu isteği kendi içinde bulamazsa, bu mesajı bütün çocuklarına ve ebeveynine broadcast yapar, fakat çok sayıda çocuk varsa iletişim içinde problem oluşturur. İletişim problemi çocuk sayısını sınırlandırılarak çözümlenmeye çalışılmıştır, ama bu hem önbellek hiyerarşisinin büyümesine hem de çocuk sayısı az olduğu için “kuzen” önbellek sunucusunda bulunan bir bilginin kaynak sunucudan getirilmesini neden olur. Aynı zamanda eğer hiç bir hiyerarşi yapısı kullanılmazsa durum daha kötü bir hale gelir. Bütün önbellek sunucuların aynı sevide tutulduğunu düşünürsek, istekler bütün önbellek sunuculara gönderilecek demektir. Bunu düzenlemek için sistemin coğrafik büyüklüğü ve önbellek sunucu sayısı sınırlandırılarak sistemin ölçeklenebilirliği tehlikeye atılmış olur. Sonuç olarak bu sistemde performans problemi hala çözümlenememiştir, fakat yine de araştırmalar FTP dosyalarının hiyerarşik önbellekleme olarak kullanılarak getirilmesinin



geniş alan ağı trafiğini yarı yarıya azalttığı göstermiştir [6].

Uyarlanabilir web önbellek sistemleri ise sabit hiyerarşik sistem yerine önbellek gruplar içinde bulunan hiyerarşik yaklaşım ile performansı artırmaya çalışmaktadır. Önbellek grupları daha esnek yaparak yoğun sunucuların azaltılması sağlanmaya çalışılmaktadır. Uyarlanabilir web önbellekleme sisteminin performansını analiz etmek için UCLA tarafından bir simülasyon geliştirilmiştir. Parsec simülasyon dili kullanılarak UCLA paralel hesaplamalar laboratuvarı tarafından geliştirilen bu yazılım önbellek grup yönetim protokolü (CGMP) ve önbellek çözümleme protokolünün (CRP)[15] performansını ölçmeye çalışmaktadır.

2.4 Bilgi-Tabanlı

2.4.1 İstek Yönlendirme

Sistemde bulunan önbellek sunucular hakkında bilgiler tuttukları bilgi depoları vardır. Bu grubu ait olan özet önbellek sistemi, CISI ve ipucu-hiyerarşi sistemleri öncelikle kendilerinde bulunan diğer sunucular hakkındaki bilgiyi kullanarak bilgiyi doğrudan bulunmasını bekledikleri sunucudan getirirler. Eğer aradıkları bilgiyi kendi bilgi depolarında veya bilginin bulunması gereken önbellek sunucuda bulamazlarsa doğrudan kaynak sunucudan getirirler.

2.4.2 Yük Dengeleme

Bilgi tabanlı sistemler yük dengelemek için diğer sunucuların bilgilerini tuttukları tabloları kullanarak ağ trafiğini azaltmaya çalışırlar. Özet önbellek sunucular bu bilgiler Burton Bloom tarafından 1970'lerde bulunan Bloom Filter metodu kullanılarak

sıkıştırılmış yapılarda saklarlar. Bloom Filter yaklaşımı hakkında daha detaylı bilgi [2]'de bulunabilir.

Bilgi tabanlı bir diğer sistem ise CISI sistemidir. Bu sistemde belli uzaklıktaki önbellek sunucular belli bir zaman aralığında kendilerinde oluşan değişiklikleri diğer sunuculara bildirdikleri için, önbellek sunucuları sadece birbirlerinin bilgilerinin indeksini tutarlar. Bu işlem bilgilerin tutulduğu dizinlerin boyutlarını ve ağ trafiğini azaltır. Bu sistemdeki en önemli problem bütün sistemin kendi kendini senkronize edebilme özelliği ile karşı karşıya kalmasıdır. Bu sistemdeki bütün sunucuların aynı anda senkronize olmasına sebep olabilir ve bu da sistemin performansında azalmaya neden olur. Sistemdeki bir çok sunucu aynı anda birbirlerine uyarı mesajı gönderebilir. Bu belli zaman aralıklarında sistemin ağ trafiğini çok artması manasına gelir. Bu sorunu çözmek için senkronize olma zamanlarını değiştiren bir parametre kullanılabilir.

İpucu-hiyerarşi sistemleri, özet önbellek ve CISI sistemlerine benzer metotlar kullanılır. Bilgilerin daha yaygın olarak dağıtılması için önbellek sunucuların büyük sayıda ipucu tutmalarına izin verilir ve bunlara hızlı erişimleri sağlanır. İpucu tabloları hafızada duran dosya yapılarıdır, bu tablolarda bulunan kayıt büyüklükleri küçük olduğu için ipuçlarını paylaşırken ağ trafiğini azaltır. Ayrıca küçük kayıt büyüklüğü sabit olan fiziksel hafıza miktarı içinde daha çok ipucu tutabilmeye imkan sağlar.

Önbellek sunucular arasındaki trafiği azaltmak için ipucu-hiyerarşi sistemleri hiyerarşi ağaç budama yöntemini kullanırlar ve böylece yapılan değişiklikler sadece bu değişiklikten etkilenecek sunuculara gönderilir. Örnek vermek gerekirse, eğer bir önbellek sunucu çocuğunun elinde olan bilgiye sahip olduğunu biliyorsa, ebeveyninden gelen bilgiyi çocuğa göndermez. Bu sistemlerdeki potansiyel problem eğer bir bilgi sisteme ilk defa eklenirse oluşur. Bu bilginin bir çok kopyasını koymak diğer sunuculara bağlanmak gerekir ve bu da ağ trafiğini artırır.

2.4.3 Tutarlılık

Bilgi-tabanlı sistemler tutarlılık için meta-data tutarlılığını sağlamalıdır. Özet önbellek sistemlerinde son değişiklik tarihi veya büyüklük gibi parametreler bilgilerin var olup olmadığını gösterir. Meta-data bilgilerinin tutarlılığı sağlamak için, özet bilgilerinin belli aralıklarla değiştirmeleri gerekir. Belli zaman aralıklarıyla bilgilerin güncellenmesi özet tablolarında bulunan bilginin asıl olması gereken önbellek sunucuda bulunmamasına neden olur, fakat bu durum sistemin tutarlılığını etkilemez sadece başarı oranı etkiler.

CISI sisteminde de, önbellek sunucuda ve bilgilerde olan değişiklikler belli zaman aralığıyla diğer sunucularla paylaşılır. Her bir değişiklik olduğunda asıl önbellek sunucu, bilgi ID'si (URL), aksiyon (ekleme veya çıkarma), uzaklık (asıl önbellek sunucu ile arasındaki) ve zaman (asıl önbellek sunucunun saatine göre değişiklik zamanı) bilgilerini tutar. Önbellek sunucular bu değişikliklere dayanarak bütün sistem içindeki bilgileri paylaşırlar ve değişikliklerden haberdar olurlar.

İpucu–hiyerarşi metodu meta-data ve bilgi tutarlılığını sağlamak için başka bir metot kullanır. Önceki tutarlık metotları zayıf tutarlılık sağlarlar, bu sistemde ise daha güçlü tutarlılık sistemi sağlanmıştır. Meta-data tutarlılığını sağlamak için belli aralıkla değişiklikleri göndermek yerine, değişiklik olduğu zaman değişiklikler diğer sunuculara gönderilir. Eğer önbellek sunucu her değişiklikte bütün önbellek sunucuları değişiklikten haberdar ederse, bu ağda çok yüksek bir trafiğe neden olacağı için bütün sistem yerine yakın olan sunuculara değişiklikleri gönderir. Bu durumda da bir önbellek sunucu kendinde bilgiyi bulamazsa sadece kendine yakın olan önbellek sunucularda bulma şansına sahiptir.

Bu sistem ayrıca kendine özgü bilgi tutarlılık sistemine sahiptir. Bilgi tutarlılığını sağlamak için güncelleğinde-gönder (push-on-update) sistemi kullanılmaktadır. Eğer bir bilginin geçerliliği bitmişse, yeni bilginin kopyaları o bilgiyi daha önce tutan önbellek sunuculara gönderilir.

2.4.4 Ölçeklenebilirlik

Bilgi-tabanlı sistemlerde ölçeklenebilirliği etkileyen en önemli problemten biri bilgi listelerinin tutuluğu veri yapılarının büyüklüğüdür. Özet Önbellek sistemlerinde daha öncede belirtildiği gibi verileri sıkıştırmak için Bloom Fitler metod kullanılmaktadır. Bu sistemlerde hafıza gerekliliklerinin yanında ağ kullanım miktarı da ölçeklenebilirlik önündeki engellerden biridir. Ağ kullanımındaki artış özet tablolarının tutarlığı ile ilgilidir. Tutarlılık kısmında anlatıldığı gibi bu sistemde özet tabloları belli bir eşik değerine kadar tam doğru olmayabilirler.

CISI sistemini ölçeklenebilirliği, her bir önbellek sunucuda tutulan dizin sayısının yakında olan önbellek sunucular ile sınırlandırılması ile çözümlenmiştir. İletişim yolunu ölçeklenebilir yapmak için internette kullanılan Border Gateway Protokolüne (BGP) benzer bir protokol kullanılmaktadır. Bu mekanizma aynı zamanda sistemin kendini otomatik olarak değişikliklere adapte etmesini sağlar.

İpucu-hiyerarşi sistemi de, sistemin ölçeklenebilirliğini artırmak için sıkıştırılmış veri yapılarını kullanır. Özet sisteminden farklı olarak, eğer önbellek sunucu aradığı bilgiyi kendi içerisinde bulamazsa hiyerarşi sistemini kullanarak bu bilgiyi ebeveynlerinden sorar. Bu hiyerarşi bir ağaç şeklinde görülebilir, fakat gerçekte bu dinamik olarak alt dallarını etkileyen bir veri yapısıdır. Plaxton algoritmasını kullanarak bütün alt dallarını ve bunların kollarını ölçeklenebilir ve hata toleranslı bir şekilde ayarlar [7]. Buna ek olarak, hiyerarşik sistemin kullanılmasıyla ipucu-hiyerarşisi, hiyerarşi ağacı budama metodu ile sistemde etkilenen önbellek sunucu sayısı sınırlandırmaktadır. Bu metot hem ağ trafiğini azaltmaktadır, hem de sistemin ölçeklenebilirliğini artırmaktadır.

2.4.5 Performans

Bilgi-tabanlı sistemlerin performansı için her sistem ayrı incelenecektir. Özet önbellek sistemlerinin performans analizi Harvest hiyerarşik önbellek sistemi ile karşılaştırılarak yapılmıştır. Performans analizinde Bloom Filter yaklaşımının, sistemin ağdaki bant genişliği kullanımını azalttığı görülmüştür. Bloom Filter tabanlı özet tabloları, mesajların büyüklüğünü %25 ile %60 arasında azaltmıştır. Özetlerin büyüklüğü yaklaşık olarak 32 bayt mesaj başlığı ve 4 bayt içerikten oluşmaktadır. Bu sonuçta mesajların boyutunun ICP tabanlı hiyerarşik sistemlere göre %55 ile %64 arasında daha küçük olduğunu göstermiştir, aynı zamanda CPU kullanım oranını %30 ile %95 arasında azaltmıştır [2].

CISI sistemlerinde kullanılmak istenen bilgi listeleri ve bu listelerin sadece yakın olan sunuculara ait olması ağ trafiğini azaltmaktadır. Şu anlık önbellek sunucular arasında iletişim başarı oranı artırmaya hedefliyor, fakat kaynak sunucudaki bilginin istemcilere diğer önbellek sunucudan daha yakın olduğu durumlarda, önbellek sunucu kullanmak avantajlı değildir. Bu yüzden her zaman önbellek sunucular arasındaki iletişimi maksimize etmek kazançlı olmayabilir, ayrıca her sunucu komşu önbellek sunucuların bilgilerini içeren dizinleri kendisi yönetir. Sadece değişiklikleri birbirlerine haber vermeleri fazla iletişim gerektirmez. Bu yaklaşım için şu anda gerçekleştirilmiş bir simülasyon olmadığı için gerçek dünyada nasıl bir performans göstereceği belirsizdir [5].

İpucu-hiyerarşi sisteminin performansı ölçeklenebilirlik konusunda karşılaşılan konularla aynıdır. Fakat performans iki sebepten dolayı sınırlanabilir. Birincisi, ipucu önbellek veri yapısı küçük ve hızı normal bir veritabanından bile daha yavaştır. Diğeri ise önbellek sunucular, bilgilerin saklandığı sunucular hakkında süresi dolmuş ipuçları içerebilir. Fakat yapılan araştırmalar bu sistemin uygulamalarının sistemde 1.3 ile 2.3 kat arasında hız artışı sağlamıştır, bunun yanında ipucu tablosunun büyüklüğünün de performansı doğru orantılı olarak etkilediği görülmüştür [7].

3. TARTIŞMA KONULARI

Bu bölümde sistemlerin performansını etkileyen diğer konuları işlenecektir. Bu faktörler de dağıtık web önbellek sistemlerinin performanslarını karşılaştırırken göz önünde bulundurulması gereken web içeriği, eşitlik ve tutarlı özetleme gibi önemli kriterlerdir.

3.1. Web İçeriği

Web içeriği dağıtık ve dağıtık olmayan önbellek sistemlerinde sistemin performansını etkileyen en önemli etkenlerden biridir. Bilgi dinamik ve önbellek-olabilir değilse; önbellek sisteminin performansını artırılması konusunda yardımcı olamaz. Ne yazık ki, web teknolojileri bu konuda gelişmektedir ve bir çok firma dinamik ve kullanıcıların isteklerine göre üretilen web sayfaları yaratmaktadır. Bu da genellikle içeriğin “önbellek olmasın” veya çok kısa bir yaşam-süresi (TTL) ile işaretlenmesine neden olmaktadır. Gelecekte tasarlanacak önbellek sunucular bu konuda değişik çözümler üretmek zorunda kalacaklardır.

Günümüzde web içeriğinde artan diğer bir konu ise multimedya dosyalarının dağıtılmasıdır. Bu dosyalar standart bir resim veya yazı dosyasından çok daha büyüktür ve ağ altyapısındaki bant genişliğinin işgal etmektedir. Multimedya dosyalarının önbellek sunucular tarafından saklanması ağ trafiğini ve kaynak sunucu yükünü azaltacaktır.

3.2. Eşitlik

Dağıtık önbellek sistemleri, bütün önbellek sunucuların aynı grup tarafından yönetildiğini kabul ediyor, fakat eğer sunucular başkaları tarafından yönetilirse, bu eşitlik konusunda yeni problemlerin ortaya çıkmasına neden olur. Bir önbellek

sunucunun dağıtık önbellek sistemdeki diğer sunucuları, istekleri ile kullanması ne kadar uygundur ?

İki önbellek sunucunun beraber çalıştığını düşünelim ve birinci sunucunun ikinci sunucudan çok daha fazla kullanıcıya hizmet verdiğini kabul edelim. Eğer bu iki sunucu beraber çalışırsa, az sayıda kullanıcıya hizmet eden sunucu çok sayıda kullanıcıya hizmet eden sunucuya göre fazla miktarda isteğe cevap vermek zorunda kalacaktır. Dağıtık önbellek sistemleri bu konuyu görmezden gelmektedir, fakat önbellek sunucular daha yaygın hale geldikçe bu konuların daha çok düşünülmesi gerekecektir.

3.3 Tutarlı Özetleme (Consistent Hashing)

Tutarlı özetleme dağıtık önbellek sunucuların göz önünde bulundurulması gereken bir diğer önemli problemdir. Tutarlı özetleme belli bir internet adresini belli bir önbellek sunucu kümesine adreslemedir. Bu metodun avantajı dağıtık önbellek sisteminin avantajlarından yararlanmak için çok fazla iş yapmaya gerek kalmamasıdır. İstemci bir adresi istediğinde, istek doğrudan o adresi kendisinde tutan önbellek sunucuya yönlendirilir, böylece sunucularda herhangi bir değişiklik yapmaya gerek yoktur. Aslında sunucuların diğer sunucularla yardımlaşmalarından bile haberleri olmak zorunda değildir.

Tutarlı özetlemenin bazı dezavantajları da vardır. Birincisi, tutarlı özetlemede internet isteklerinin kaynak sunucudan önceden tasarlanmış önbellek sunucuya yeniden yönlendirilmesi gerekir, eğer istemci yazılımlarında bir değişiklik yapılmadıysa bu problem oluşturabilir. İkincisi ve daha önemlisi ise, tutarlı özetleme yoğun sunucular üzerindeki yükün azaltılması konusunda herhangi bir şey yapmamaktadır. İstekler ağ trafiğinden daha çok internet adreslerine göre yönlendirildiği için, bazı sunucular çok fazla sayıda istekle karşılaşabilir.

4. ANALİZ VE ÖNERİLER

Coğrafik push sistemleri, hiyerarşik, uyarlanabilir, özet, CISI ve ipucu-hiyerarşi olmak üzere altı web önbellek sistemi incelendi. Bu sistemler tekil sunucu, sorgu-tabanlı ve bilgi-tabanlı olarak üç gruba ayrıldı. Bu bölümde sistemlerin incelenen özelliklerinin karşılaştırılacak ve aralarındaki ilişkiler analiz edilecektir. Öncelikle hangi sistem istek cevap süresini en aza indirdiği, ikinci olarak hangi sistemin sunucu yükünü en fazla azalttığı ve üçüncü olarak hangi sistemin toplam ağ trafiğini en aza indirdiği incelenecektir.

4.1. İstek Cevap Süresi

Dağıtık önbellek sistemlerini karşılaştırmak için kullanılan en önemli kriterlerden biri sistemin istek cevap süresini ne kadar azalttığı yada isteğin gelmesi süresinin ne kadar olduğudur. Gecikme süresi istenilen bilginin getirilebilmesi için kaç hop sayısına ihtiyaç duyulduğu ile doğru orantılıdır ve hop sayısı sistemdeki yönlendirme mekanizması ile belirlenir. CISI ve ipucu-hiyerarşi sistemleri cevap verme süresini kısaltma üzerinde çalışmışlardır. Uyarlanabilir ve hiyerarşik sistemler ise daha çok başarı oranını artırma üzerinde odaklanmışlardır. Başarı oranını artıma ile cevap süresini azaltma arasında ters orantı vardır ve başarı oranı artırmak için çok sayıda önbellek sunucudan araştırma yapılması gerekmektedir, bu işlem de her bir istek için çok fazla hop sayısı demektir.

4.1.1. Tekil Sunucu

Coğrafik push web önbellek sistemi cevap süresini azaltmamaktadır. Her bir istemci doğrudan birincil sunucuya bağlanmaktadır. Birincil sunucu istemciyi istenmiş bilginin kopyasını tutan yerel sunucuya yönlendirmektedir. Yerel sunucu bilgiyi

göndermeden önce elindeki bilginin tutarlı olduğunu doğrulamak için kaynak sunucuya ulaşmaktadır. Bu işlemlerde her bir isteğin en az üç hop içerdiğini göstermektedir. Bu yaklaşım cevap süresini azaltmaktan çok sunucuların yük miktarını azaltmaya yönelmiştir.

4.1.2 Sorgu-tabanlı

Sorgu-tabanlı yaklaşımda cevap verme süresini azaltmak yerine başarı oranını artırmak üzerinde durulmuştur. Hiyerarşik sistemde bir önbellek sunucuda aranan bilginin bulunamaması durumunda ebeveyninde arama ihtiyacı doğuracağı için hop sayısını artırmaktadır. Bu işlem hiyerarşi yapısı büyüdükçe hop sayısının artmasına sebep olmaktadır, her yeni hop'ta yeni bilgiler ile karşılaşılacağı için başarı şansı artmaktadır. Hiyerarşik önbellek sistemlerinde başarı oranı artmasına rağmen, cevap verme süresi de artmaktadır. Uyarlanabilir önbellek sisteminde ise hop sayısı için en kötü senaryo ağın çapı kadardır [8].

4.1.3 Bilgi-tabanlı

Bilgi-tabanlı sistemler başarı oranının artırılması yerine erişim ve cevap süresinin azaltılması üzerinde odaklanmışlardır. Bu yaklaşım ayrıca sunucu yükünün ve ağ trafiğinin azaltılması avantajlarını da yanında getirmektedir. Bilgi-tabanlı sistemler istenilen bilginin bulunduğu önbellek sunucuya doğrudan ulaşmaya çalıştığı için sadece bir hop yaparlar. Önbellek sunucu ellerindeki listede bulunursa bilgi doğrudan o sunucudan getirir, eğer bulunamazsa yine doğrudan kaynak sunucudan getirir, böylece her iki durumda da sadece bir hop yapılıır.

4.2 Sunucu Yüğü

Sunucu yüğü sistemlerin istekleri nasıl işlediğı ve önbellek sunucuları nasıl organize ettiğı ile ilgilidir. Sunucu yüğü sistemlerin bilgi ve meta-data tutarlılıklarını korumak için kullandıkları mekanizmalara bağıdır.

4.2.1 Tekil Sunucu

Tekil sunucu mimarisinde yük dengesini sağlamak için bilginin hangi sunucuya ve ne zaman kopyalanacağına karar vermek için eşik değeri parametresini ve bilginin istek geçmişini kullanır. Coğrafiş push sistemi bu metot sayesinde bilgilerin coğrafiş popülaritelerine göre ağı üzerinde dağıtılmasını sağlar. Eşik değeri bir önbellek sunucunun belli zamanda tutabileceğı en fazla bilgi miktarıdır. Bu sayede bilgiler sistem üzerinde eşit olarak dağıtılır. Her istek için göndereceğı bilginin tutarlı olup olmadığını kontrol etmesi gerekir, fakat bu kontrol sunucuların yük miktarını artırır.

4.2.2 Sorgu-tabanlı

Sorgu-tabanlı sistemler sunucu yükünü azaltmak için kendine ait bir metot kullanır. Hiyerarşik önbellek sistemi, önbellek sunucuların hiyerarşi yapısını ayarlar ve istekleri hiyerarşinin değışik seviyelerine gönderir. Bu yolla sunucu başına düşen yük miktarı azaltır. Fakat, eğer hiyerarşi yapısında çok fazla çocuk varsa ve hiyerarşinin seviyesi çok yüksekse ise sunucu yükleri artmaktadır.

Uyarlanabilir önbellek sistemleri ise sunucu yükünü azaltmak için üst üste geçen dinamik grupları gizli hiyerarşi şeklinde kullanır. Bu metot ile de hiyerarşik önbellek sunucu mimarisinde olduğı gibi eğer hiyerarşinin seviyesi çok fazla olursa sunucuların toplam yük miktarı artmış olur. Bu hiyerarşi boyunca her önbellek sunucunun istek ile ilgili işlem yapması demektir.

4.2.3 Bilgi-tabanlı

Bilgi-tabanlı sistemlerde sunucu yükü, sunucuların ellerinde bulunan sıkıştırılmış veri yapılarında bilginin bulunduğu sunucuyu bulması olarak yorumlanabilir. Bu veri yapıları az tutarlı ve belli zaman aralıklarında güncellendiği için fazla sunucu yüküne sebep olmazlar. Özet önbellek sunucular ise, her bir sunucunun hafıza kullanım miktarını Bloom Filter metodu ile azaltmaya çalışırlar, CISI sistemi ise sistem yükünü azaltmak için sadece komşu önbellek sunucuların bilgilerini tutar. Bu bilgilerin sınırlandırılması, sunucuların meta-data bilgilerinin tutarlılığını kontrol etmek için yaptıkları iş miktarını azaltmaktadır. Bu metodun en büyük dezavantajı ise, önbellek sunucular bütün önbellek sistemi hakkında bilgiye sahip olmadıkları için sistemin başarı oranının düşmesidir. İpucu-hiyerarşi sisteminde de ipucu tablosu daha fazla bilgiyi tutmak için sıkıştırılmış bir yapıya sahiptir. Ayrıca bilgi güncelleme sırasında güncellenecek sunucuları ağaç budama yöntemiyle seçtiğinden dolayı sistemin yükü azalmaktadır.

4.3 Ağ Trafik

Ağ trafiği istek yönlendirme veya tutarlılık güncelleme ile oluşabilir. İstek yönlendirmede ağ trafiği bir tek isteğin bütün ağ üzerinde gezmesinden kaynaklanır aynı zamanda bilgi ve meta-data tutarlılığını sağlamak için yapılan güncellemeler de ağ trafiğini artırır. Dağıtık önbellek sistemlerindeki bilgi güncelleme sıklığı ise sistemlerin ne kadar tutarlı olmak istedikleri ile doğru orantılıdır.

4.3.1 Tekil Sunucu

Tekil sunucularda en büyük trafik problemi birincil sunucularda oluşmaktadır. Çünkü bu sunucular sisteme gelen bütün isteklerin yönlendirildiği merkezlerdir. Ayrıca her yerel sunucu bir bilgiyi istemciye göndermeden önce onun geçerliliğini kontrol

etmek için kaynak sunucuya bağlanması da ağ trafiğini artıran bir diğer sebeptir.

4.3.2 Sorgu-tabanlı

Sorgu-tabanlı sistemleride ağ trafiğine sebep olan en önemli nedenlerden biri multicastingtir. Hiyerarşik önbellek sisteminde sistem büyürse multicasting çok fazla ağ trafiğine sebep olacaktır. Eğer önbellek sunucu sayısı artarsa, iletişim ve işlem miktarı “*quadratic*” olarak artacaktır [2]. Hiyerarşik sistemde, üst seviyede olan önbellek sunucular alt seviyedeki sunuculardaki isteklere de cevap vermek zorunda oldukları için bu sunuculardaki ağ trafiği fazla olacaktır.

Uyarlanabilir web önbellek sisteminde de ağ trafiği önemli bir problemdir. Eğer önbellek sunucu aradığı bilgiyi bulamazsa bilgi istemci makinesi ile kaynak sunucu arasındaki bütün gruplara multicast yapılmalıdır. Büyük ağlarda bu yaklaşım ağ trafiği çok artıracaktır, fakat sistemin tasarımıda popüler bilgilere daha hızlı erişim olduğu için popüler bilgiler multicasting sayesinde bir çok grupta bulunmaktadır. Bu da popüler bilgilerin başarı oranını artırmaktadır, başarı oranı arttıkça isteklerin meydana getirdiği ağ trafiği azalmaktadır. Popüler olmayan bilgiler kaynak sunucuya yakın önbellek sunucularda saklandığı için ağ trafiğini en çok artıran bilgiler bunlardır. Uyarlanabilir önbellek sistemi yoğun sunucuların yoğunluğunu azaltmak için oylama metodunu kullanmaktadır. Oylama metodu kullanmakta ağ trafiğinin artmasına neden olmaktadır, çünkü grup üyeleri kendilerini konfigüre etmek için belli aralıklarla birbirlerini oylama mesajları gönderirler.

4.3.3 Bilgi-tabanlı

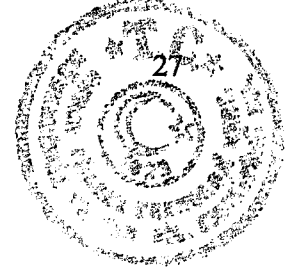
Bilgi-tabanlı sistemlerde istek yüklememe mekanizmasında bir tane hop kullanıldığı için ağ kullanımı azalmaktadır. Bu sistemler ayrıca bilgileri sıkıştırılmış veri yapılarında tutup bunları belli aralıklarla güncelledikleri için bu işlem ağ trafiğini

azaltmaktadır.

Daha öncede belirtildiği gibi özet önbellek sistemleri veri yapılarının büyüklüğünü azaltmak için Bloom Filter metodunu kullanmaktadırlar ve özet tablolarında zayıf tutarlılık seçilmiştir. Büyüyen sistemlerde bilgilerin güncellemesini belli aralıklarda yapmak bütün bilgilerin tutarlı olduğunu bilmekten daha avantajlıdır. Ayrıca Bloom Filter yaklaşımı sayesinde mesajların büyüklüğünü hiyerarşik sistemler kullanılan mesajlardan %55 - %64 daha azdır[2] ve CPU kullanım oranı %30- %95 daha düşüktür [2]. Yalnız buradaki tek ön kabul özet sisteminde mesajların sistem içindeki bütün sunuculara yönlendirilebilmektedir.

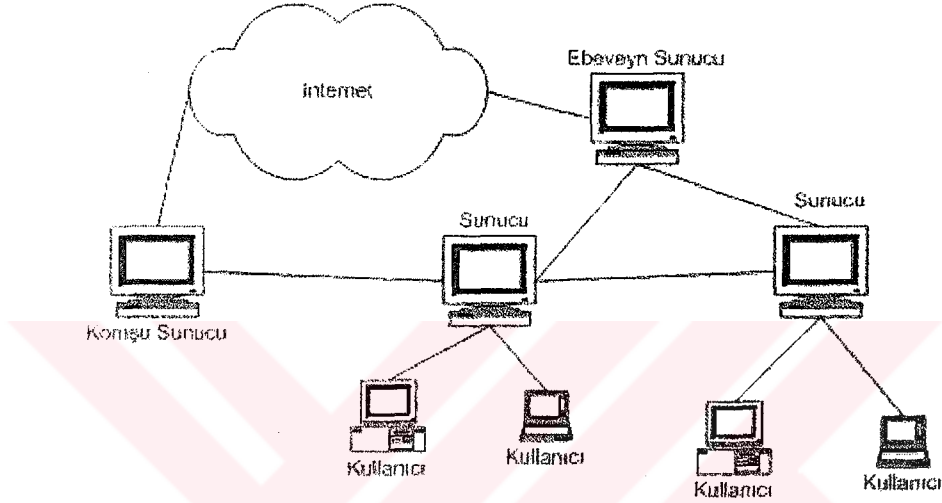
CISI sisteminde sadece komşu önbellek sunucular hakkında bilgi tutmak sunuculardaki bilgi sayısının az olmasını sağlamaktadır. BGP yönlendirme protokolü sayesinde önbellek sunucular arasında ölçeklenebilir bir iletişim çizgesi oluşturulmaktadır. CISI sisteminde bilgi sadece belirli uzaklıktaki önbellek sunucularda arandığı için diğer bilgi-tabanlı sistemlere göre daha az ağ trafiği oluşturmaktadır.

İpucu-hiyerarşi sistemi de ipucu tablolarını sıkıştırılmış veri yapılarında tutar. Ayrıca performansı artırmak için güncellemeleri yaparken hiyerarşi ağacı budama yöntemini kullanır. Özet önbellek sisteminde olduğu gibi bir istek bütün ağ üzerinde yönlendirilebilmelidir.



5. TASARIM

Bu bölümde daha önce anlatılan kriterlere göre gerçekleştirilen özgün dağıtık web önbellek sunucu sisteminin tasarımı anlatılacaktır.



Şekil 5.1: Dağıtık Web Önbellek Sisteminin Genel Mimarisi

5.1. Mimari

Sistem tasarımında önbellek sunucuların birbirlerine çok uzak olmadığı kabul edildi, böylece önbellek sunucularının bilgilerini hızlı bir şekilde paylaşabilmeleri sağlanmaktadır. Eğer sistem birbirine çok uzak sunuculardan oluşursa, önbellek sunucuların bilgi paylaşımı yavaş olacaktır ve önbellek sunucuların cevap süresi eşik değerinden fazla olacağından birbirlerinin bilgilerinden faydalanamayacaklardır. Bu durumda bize tekil çalışan sunuculardan daha kötü performans verecektir, çünkü bekleme süresi ve ağ trafiği artacaktır.

Sisteminde iki tip önbellek sunucu vardır:

- Web erişimi olan sunucular: Bu sunucular bilgileri doğrudan kaynak sunucudan getirebilirler.
- Web erişimi olmayan sunucular: Bu sunucular bilgiyi doğrudan kaynak sunucudan getiremezler, kendilerine gelen isteği bağlı oldukları önbellek sunucuya yönlendirmeleri gerekmektedir.

Sunucular arasında böyle bir ayırımın olmasının avantajını bir örnek ile açıklayalım. A ve B önbellek sunucular aynı YAA (Yerel Alan Ağı) üzerinde ve B makinesi aynı zaman YAA için ağ geçidi (gateway) olsun. A bilgiyi kaynak sunucudan getirmek istediğinde istek B makinesi üzerinden geçmek zorundadır. Eğer A makinesi gelen isteği B makinesine yönlendirirse, B makinesi de fazla iş yapmadan aynı bilgiye sahip olur ve sistemin başarı oranı artar. Sistem içinde sadece web erişimi olmayan sunucular kullanılırsa Harvest önbellek sistemine benzer [6].

5.2 Önbellek Yerleştirme

Her önbellek sunucu sadece istemcilerin istedikleri bilgileri saklarlar. Bu yüzden 'pushing' ve 'prefetching' sistemde bulunmamaktadır. Aynı zamanda sistemde kaynakları verimli kullanmak için çok büyük bilgiler saklanmamaktadır [17]. Bunun sebebini şöyle açıklayabiliriz: S büyüklüğünde ve F sıklığında erişilen O bilgisi olduğunu düşünelim. Aynı zamanda o1 ve o2 olan iki küçük bilgi olsun ve bu bilgilerin büyüklüğü s1,s2 ve sıklıkları f1,f2 olsun. Ayrıca $s1+s2=S$ olarak kabul edelim, eğer S eşik değerinden büyükse $f1>F$ ve $f2>F$ olma olasılığı çok büyüktür. Eğer O bilgisi depolanmışsa bunların yerine getirilecek olan küçük bilgilerin ihtiyacı olan ağ genişliği $s1*f1+s2*f2$ olur. Eğer O bilgisi depolanmamışsa O bilgisini getirmek için gerekli olan ağ genişliği $S*F$ dir. Eğer $f1>F$ ve $f2>F$ olarak kabul edersek;

$$S*F = (s1+s2)*F = s1*F+s2*F$$



$$f_1 > F \Rightarrow s_1 * f_1 > s_1 * F$$

$$f_2 > F \Rightarrow s_2 * f_2 > s_2 * F$$

(5.1.)

$$s_1 * f_1 + s_2 * f_2 > S * F$$

Buradan eşik değerinden büyük bilgilerin saklanmamasının daha uygun olduğu görülmektedir.

5.3 Önbellek Sunucu Çözümleme

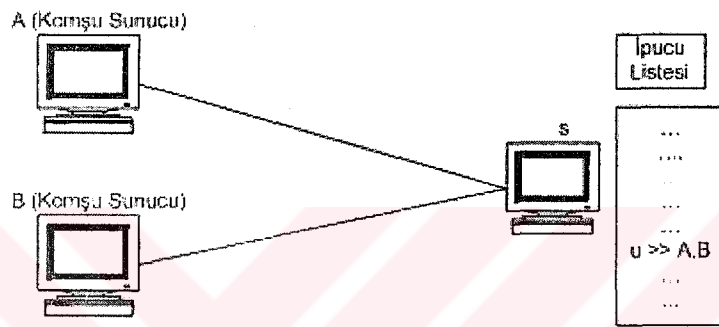
Önbellek sunucudaki bütün adresler için en azından bir ipucu vardır. İpuçları bilginin kaynak adresini (URL) ve bilginin bulunduğu önbellek sunucu adresini içeren bir veri yapısıdır. İpuçları istenilen bilginin hangi sunucuda olabileceğini söyler, fakat hangi sunucuda olmayacağını söylemez. Bilgi önbellek sunucuda başarılı bir şekilde saklandığı takdirde sadece o URL ile ilgili bir ipucu oluşturulur. Bu yüzden HTTP hata mesajları ve saklanması uygun olmayan bilgiler için ipucu oluşturulmaz.

Eğer bir önbellek sunucu kullanıcının istediği bilgiyi kendi deposunda bulamazsa, bu isteği daha sonra anlatılacak hız kriterine göre en hızlı komşularına yönlendirir. Eğer belli zaman içinde hiçbir komşusu bu bilgiyi ona dönmezse, önbellek sunucu bilgiyi sunucu tipine göre ebeveyn sunucudan veya kaynak sunucudan getirir.

Hiyerarşide yüksek seviyede olan sunucular çocuklarından gelen istekleri sadece kendi depolarda ararlar. Eğer böyle bir sunucu isteği kendi deposunda bulamazsa kendisinin bir üst seviyesindeki sunucudan veya kaynak sunucudan ister. Burada hiyerarşi seviyesi ne kadar büyürse bekleme süresi o kadar artacaktır. Ancak hop sayısında bir değişiklik olmayacaktır, çünkü zaten istemci bilgiyi doğrudan kaynak sunucudan istediği zaman da aynı sunuculardan geçmesi gerekir.

İsteği yönlendiren komşu sunucu sayısı o sunucunun derecesi olarak nitelendirilir. Eğer ağ içinde kullanılan ortalama derece artarsa başarı oranı da artacaktır, çünkü

önbellek sunucular toplamda daha büyük önbellek bilgisi arasından arama yapacaktır. Her bir önbellek sunucunun kapasitesini 2 GB olarak alırsak ve sunucuların derecesini 2'den 3'e çıkarırsak, önbellek sunucuların arama yaptıkları bilgi miktarını 2 GB artırmış oluruz. Aynı zamanda sunucular arasında paylaşılan ipucu sayısı da artar, bu da sunucuların aranılan URL adresinin bulunduğu sunucuyu bulma ihtimalini artırır. Fakat sunucuların derecesini artırmak aranılan her bilgi için gerekli olan ağ mesaj miktarını artırmaktadır ve bu da ağ kullanım miktarını artırır.



Şekil 5.2: Dağıtık Web Önbellek Sunucu Çözümleme

İstek yönlendirme kısmının çalışması şöyle anlatılabilir. Sunucu 's', URL adresi 'u' yu kendi deposunda bulamadığında u'yu elinde bulunan ipucu listesinde arar. Bu listede aynı adres ile ilgili birden fazla sunucu bulabilir. Komşu sunucuların listesi hem run-time zamanında da hemde konfigürasyon dosya sayesinde oluşturulur. Her sunucu çalışmaya başladığında diğer önbellek sunucuların özelliklerini bir konfigürasyon dosyasından okur. Bu dosya bulunan sunucuların sırası isteklerin gönderilme sırasını doğrudan etkiler. Eğer dosyada A sunucusunun ismi B sunucusunun isminin üstünde ise istek öncelikle A sunucusuna, daha sonra B sunucusuna gönderilir. Fakat, bu sıralama statik değildir, s sunucusu A ile iletişim kurduğunda A sunucusundan gelen data miktarını ve geçen süreyi hesaplar. Belli zaman aralıkları ile bütün sunucular için toplanan bilgiler ile sunucuların performansı hesaplanır. Bu performans için aşağıdaki metot kullanılır.

$$Hız[x]=Hız[x]*Gecikme + Okunan_Data[x] / Geçen_Süre[x] *(1-Gecikme) \quad (5.2)$$

Gecikme 0 ile 1 arasında seçilen bir sabittir. Eğer gecikme değeri çok küçük olursa, sunucuların yerleri çok hızlı değişir, çünkü sadece o anlık performanslarına göre değerlendirilir. Eğer gecikme değeri çok büyük seçilirse, çalışma sırasında sunucularda karşılaşılan sorunlar performansları etkilemez.

Eğer sunucu A eşik sürede sunucu s'ye cevap veremezse, s sunucusu A sunucusunun hızını negatif değere eşitler. Bu yüzden A sunucusu isteklerin gönderilmesi sırasında en sona gideceği için yeni istekler ona gönderilmez. Burada kullanılan eşik değeri de önemlidir. Eğer eşik değeri küçük seçilirse, s sunucu bazı ağ hatalarından dolayı başarılı olan A sunucusunu cezalandırabilir. Eğer büyük seçilirse, s sunucusu A sunucusunun çalışmadığını anlayamayacağı için gereksiz bir çok mesajı A sunucusuna göndermeye çalışır.

5.4 Önbellek Bilgi Değişim Protokolü

Daha önce anlatıldığı gibi s sunucusu aranılan u adresini kendi deposunda bulamadığında bu internet adresini diğer sunuculardan sorgular. Bu sorgulama zamanında aynı anda ipucu paylaşımı da yapılır. Bunun için verilen bütün cevaplar - olumlu veya olumsuz- ipucu içerir. Bu ipucu paylaşımı sayesinde her sunucu diğerlerinin ipucu bilgilerine de aynı zamanda sahip olur. İpucu paylaşımı sırasında 'hafıza' alanı kullanılır. Bu alan s sunucusu ile k sunucusunun en son ipucu değişim zamanını gösterir. Bu ipucular 'en son eklenenler' adında bir bağlı liste yapısında tutulur. Yeni ipucu geldiğinde listenin en başına eklenir. Eğer s sunucusu k sunucusu ile iletişim kurmaya karar verirse, bu listeden k sunucusu ile yaptığı en son ipucu değişimi zamanından sonra eklenen ipuçlarını gönderir. Sistem 2 çeşit sorgu ve 2 çeşit cevap vardır.

5.4.1. Q sorgusu

Bu sorgu bir sunucudan diğer sunucuya bilgi istemek için gönderilir. Önce sorgu tipi 'Q' kullanılır, sonra bilginin URL adresi ve en sonuna da ipuçları eklenir. Bu sıralama, sunucu mesaj tipini ve URL adresini aldıktan hemen sonra cevap verebilmesi için önemlidir. Cevap verdikten sonra gelen ipucu bilgilerine göre kendi ipucu listesini güncelleştirebilir.

Q	URL	İpucu1	İpucu2		İpucuN	.	\r\n\r\n
---	-----	--------	--------	-------	--------	---	----------

Şekil 5.3: Q sorgusu

5.4.2. P sorgusu

P sorgusu internet bağlantısı olmayan sunuculardan onların ebeveynlerine gönderilen sorgulardır.

P	URL	İpucu1	İpucu2		İpucuN	.	\r\n\r\n
---	-----	--------	--------	-------	--------	---	----------

Şekil 5.4: P sorgusu

5.4.3 Y cevabı

Bu cevaplar aranan bilginin sunucunun bilgi deposunda bulunduğunu gösterir. Bu cevaplar hem ipuçlarını hem de istenilen bilgiyi içerir.

Y	İpucu1	İpucu2		İpucuN	.	\r\n\r\n	BİLGİ
---	--------	--------	-------	--------	---	----------	-------

Şekil 5.5: Y cevabı

5.4.4. N cevabı

Bu cevaplar aranılan bilginin sunucunun deposunda bulunmadığını gösterir ve cevaplar sadece ipucu içerir.

Y	İpucu1	İpucu2		İpucuN	.	\r\n\r\n
---	--------	--------	-------	--------	---	----------

Şekil 5.6: N cevabı

İsteklerin, bilgilerin ve ipuçlarını bir mesaj toplanması ağda dolaşan mesaj sayısını azaltmaktadır. Birçok mesajı bir tek mesaj içinde toplamak ağ trafiğini azaltacaktır.

5.5. Önbellek Yer değiştirme

Önbellek yer değiştirme için LRU algoritması kullanılmaktadır. Bütün yer değiştirmeler yerel yapılmaktadır ve diğer sunucular haberdar edilmemektedir. Ayrıca ipucu listesinin boyutuyla sınırlı olduğu için bu listede yer değiştirme yapmak gereklidir. İpucu yer değiştirme algoritması olarak da LRU kullanılmaktadır

6. SONUÇ

İnternetteki büyüme her geçen gün artmaktadır ve bu artış karşısında internetin hızını korumak için yeni metotların geliştirilmesine ihtiyaç var. Bu konuda geliştirilen metotlardan biri de dağıtık önbellek sunucularının birbirlerinin kaynaklarını kullanmasına dayanan dağıtık web önbellek sistemleridir. Dağıtık önbellek sunucularının amaçları istek süresini azaltmak, sunucu yükünü azaltmak ve internetin toplam ağ trafiğini azaltmaktır. Bu dokümanda dağıtık önbellek sistemini gerçekleştiren metotlar incelenmiştir. Bu metotlar, Coğrafik push, Hiyerarşik, Uyarlanabilir, Özet, CISI ve İpucu-hiyerarşi metotlarıdır. Bu metotların dağıtık web önbellek sunucular için uyguladıkları kriterler dikkate alındıktan sonra özgün bir dağıtık web önbellek sunucu tasarımı yapılmıştır. Bu tasarım ile ilgili sonuçlar ve öneriler ise aşağıda belirtilmiştir.

6.1. Performans

Sistemin performansını ölçmek için DEC firmasından alınan erişim kayıt bilgileri kullanıldı [17]. Bu testin sonucunda %14 başarı elde edildi. Fakat elimizdeki kayıta istenilen bilgilerin %70'nin tekil bilgiler olduğundan ideal önbellek sunucunun başarısının %30 olması gerekmektedir. Fakat varolan metotta ipuçları kullanıldığı için sunucularda bulunan bütün bilgilere tam olarak ulaşamamıştır. Bunun yanında önbellek hacmi sınırlı olduğu için bazı bilgiler daha önce getirilmesine rağmen silinmiştir. Bütün bunlara karşılık sistemin başarısı %47'dir. Ayrıca testler sırasında sisteminin yeni eklenen sunucuları tanıdığı ve çöken sunucuları da kotardığı gözlemlenmiştir.

6.2. Analizler

6.2.1. Önbellek Yer değiştirme

Sistemde her önbellek sunucunun bağımsız çalıştığı ve birbirlerinin bilgi depolarına karışmadığına karar verilmiştir. Sistemin performansını artırmak için çocuk sunucuların ebeveyn sunucularında bazı bilgilerin saklanması izin vermesi gerekir. A

ve B sistemde birer düğüm ve B sunucusu hiyerarşide A sunucusunun ebeveyni olsun. Varolan sistemde, A sunucusu B sunucusunun sahip olduğu herhangi bir bilgiye sahip olursa, (komşu sunucularında getirilmiş) gerektiğinde yer kazanmak için bu bilgiyi silmek zorunda kalır. Fakat eğer A bilgiyi silmeden B'den bilgiyi saklamasını isteyebilirse daha sonra tekrar kullanabilir ve sistemin başarı oranını artırır. Bu yaklaşım sisteme 'push' özelliğinin eklenmesini demektir.

6.2.2. Önbellek Yerleştirme

Şu anki sistemde saklanan bilginin içeri ile ilgili herhangi bir şey yapmamaktadır. Web sayfaları resim veya Java programları gibi gömülü bilgiler içerebilir. Bu bilgiler sayfalar istendiğinde getirilmektedir. Gömülü bilgilerin istenildiğini veya istenilmediğini HTTP başlıklarının 'User-Agent' alanında belirtilir. Eğer önbellek sunucular bu alanı okuyup gömülü bilgilerin istenileceğini anlarsa gerekli olan bilgileri kullanıcı istemeden daha önce isteyebilir. Bu web sayfasını görmek için gönderilen mesaj sayısının azaltır, fakat HTML bilgilerini ayırmak zaman alan bir işlemdir.

6.2.3. Hata Toleransı

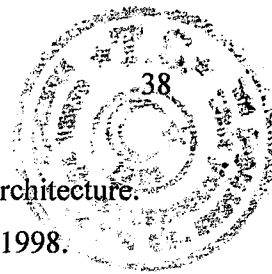
Sistem genel olarak hata toleranslı tasarlandığı halde kullanıcıların kullandığı internet gezginleri hata toleranslı değildir. Sunucular internet bağlantısı kesildiği durumda -yavaş olmasına rağmen- çalışmaya devam ederler. Fakat birçok ticari internet gezgini her protokol için bir tane önbellek sunucuya izin vermektedir. Bu yüzden sistemde çalışan başka sunucular olmasına rağmen kullanıcılar internete bağlanmamaktadır. Bunu çözmek için web önbellek ağı kullanıcı makinelerini de kapsayacak şekilde genişletilebilir. Böylece kullanıcılardaki sunucular çöktüğünde kullanıcılar sistemi tekrar ayağa kaldırabilirler.

6.2.4. Parametrelerin Etkisi

Sistemde sabit tutulan parametrelerin sistem üzerindeki etkisinin kontrol edilmesi gerekir. Gecikme deęerinin küçük veya büyük olarak seçilmesinin ve saklanacak bilgilerin büyüklüğünün eşik deęerinin deęişik deęerlerde seçilmesinin sistemin performansına olan etkilerinin gözlenmesi gerekir.

KAYNAKLAR

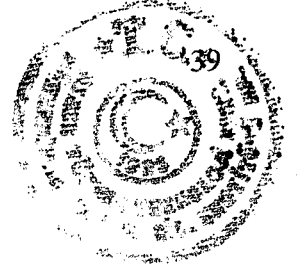
- [1] Almeida, J., Broder, A. Z., Cao, P., Fan, L. A Scalable Wide-Area Web Cache Sharing Protocol. *SIGCOMM98*, Vancouver, BC, August 1998.
- [2] Almeida, J., Broder, A. Z., Cao, P., Fan, L. Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol, *Technical Report 1361, Computer Sciences Department, University of Wisconsin-Madison*.
- [3] Blaze, M., Caching in large-scale distributed file systems. *Technical Report TR-397-92, Princeton University*, January 1993.
- [4] Cardwell, N., Karlin, A., Levy, H. M., Sharma, N., Voelker, G. M., Wolman, A. On the scale and performance of cooperative Web proxy caching. *SOSP99*, pages 16-31, December 1999.
- [5] Chase, J., Gadde, S., Rabinovich, M. Not all Hits are Created Equal: Cooperative Proxy Caching Over a Wide Area Network, June 1998.
- [6] Chankhunthod, A., Danzig, P. B., Neerdaels, C., Schwartz, M. F., Worrell, K. J. A Hierarchical Internet Object Cache. *Proceedings of the 1996 USENIX Technical Conference*, pages 153-163, San Diego, CA, January 1996. .
- [7] Dahlin, M., Kay, J., Tewari, R., Vin, H. Design Considerations for Distributed Caching on the Internet. *ICDCS99*, May 1999.
- [8] Floyd, S., Jacobson, V., Zhang, L. Adaptive Web Caching. *In Proceedings of the 1997 NLANR Web Cache Workshop*, June 1997
- [9] Floyd, S., Jacobson, V., Michel, S., Nguyen, K., Rosenstein, A., Zhang, L.



Adaptive Web Caching: Towards a New Global Caching Architecture.
Computer Networks and ISDN Systems, pages 2169-2177, November 1998.

- [10] Gwertzman, J., Autonomous Replication in Wide-Area Internetworks. *BA thesis, Harvard College, Cambridge, Massachusetts*, 1995.
- [11] Gwertzman, J., Seltzer, M. An analysis of geographical push caching. *Proceedings of International Conference on Distributed Computing Systems*, 1996.
- [12] Gwertzman, J., Seltzer, M. The Case for Geographical Push Caching. *In Proceedings of the 1995 Workshop on Hot Operating Systems*, 1995.
- [13] Krishnan, P., Sugla, B. Utility of Cooperating Web Proxy Caches, April 1998.
- [14] Kurcewicz, M., Sylwestrzak, W., Wierzbicki, A., A Distributed WWW Cache, June 1998.
- [15] Zhang, L. ,Adaptive Web Caching, <http://irl.cs.ucla.edu/AWC/simulation.html>.
- [16] Wessels, D., Claffy, K. Internet Cache Protocol RFC 2187. <http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2187.txt>, September 1997
- [17] Arlitt, M. , Williamson, C., Web Server Workload Characterization: The Search for Invariants, *In Proceedings of the SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, May 1996.
- [18] Digital Equipment Corporation, Digital's Web Proxy Traces, <ftp://ftp.digital.com/pub/DEC/traces/proxy/webtracs.html>, September 1996.

ÖZGEÇMİŞ



Kemal Yılmaz, 12 Ağustos 1978 tarihinde Erzincan'da doğdu. İlk ve orta öğrenimini İstanbul'da, liseyi Kocaeli'nde okudu. Bilkent Üniversitesi, Mühendislik Fakültesi Bilgisayar ve Enformatik Mühendisliği bölümünü 2000 yılında bitirdi. Halen TÜBİTAK Ulusal Elektronik ve Kriptoloji Enstitüsü'nde araştırmacı olarak çalışmaktadır.

