Master-thesis

What web objects contained in popular websites are delivered through hyper giants.

Fakultät IV - Elektrotechnik und Informatik
Intelligent Networks / Intelligente Netze (INET)
Research Group of Prof. Anja Feldmann, Ph.D.

Soumya Ranjan Parida
October 13, 2016

Prüfer: Prof. Anja Feldmann, Ph. D.
Betreuer: Ingmar Poese
Juhoon Kim

Die selbständige und eigenhändige Anfertigung versichere ich an Eides Statt.

Berlin, den October 13, 2016 Max Mustermann

Zusammenfassung

Mit der Verbreitung des Internets, Hyperriesen wie Google, Content-Lieferungs-Netzwerke wie Akamai usw. spielen häufig eine wichtige Rolle in den Inhalt einer Website bereitstellt. Diese Hyperriesen nicht nur verschiedene Dienste zur Verfügung stellen, sondern auch reich an Inhalt. Flash-Medien von Youtube, Login-System von Google, Facebook, Werbung von Google Ad Sinn, populäre Social-Media-Sites wie Facebook, Twitter, LinkedIn, etc. sind weit verbreitet und beliebte Dienste in den meisten Websites heute eingebettet.

Um mit dieser Nachfrage zu bewältigen, Hyperriesen haben eine große Anzahl von skalierbaren und kostengünstigen Hosting und Content-Delivery-Infrastrukturen auf der ganzen Welt einsetzen. Diese Hosting-Infrastrukturen von wenigen großen Rechenzentren, eine große Anzahl von Cache-Speicher oder eine beliebige Kombination zusammengesetzt sein. Ein solches Szenario sowie riesige Abhängigkeit zwischen populären Webseiten und Hyperriesen eine große Menge des Verkehrsflusses von Hyperriesen führen.

Um zu wissen, wie die Einbindung von beliebten Webseiten mit Hyperriesen weiterentwickelt, ist, befasst sich diese These die folgenden Forschungsfragen .Firstly gibt es jede Anwesenheit dieser Hyperriesen in interenet Architektur? Wenn Hyperriesen vorhanden sind, dann wie viel Prozent der populären Web-Sites von 100.000 Top-Websites von alexa sind mit verschiedenen Hyperriesen verbunden? Drittens Welcher Prozentsatz von verschiedenen Objekten (text / html, img, Skript, Medien etc.) werden mit Hyperriesen verbunden?

Die vorliegende Arbeit liefert eine quantitative Forschung von Web-Verbindungen für alexa Top 100.000 Websites. Wir präsentieren Ihnen die Planung, Durchführung und Analyse von verschiedenen Arten von Objekten in verschiedenen Websites enthalten, die zu unterschiedlichen Hyperriesen verbunden sind. Die Arbeit folgt zwei Pfaden. Zum einen, um verschiedene Arten von Objekten in Webseiten zu beziffern, werden wir Homepages von oben 100.000 Websites von alexa Webseite und sammeln das Vorhandensein von Hyperriesen Infrastruktur verknüpft verschiedene Objekte wie Bilder, externe Links, Skripte, eingebettete Videos, CSS-Dateien kriechen usw. in diesen Websites. Zweitens untersuchen wir die Objekte, um den Grad der Verbindung zwischen den oberen 100.000 Websites und hypergiant zu erfahren.

Die experimentellen Ergebnisse dieser Arbeit diskutiert werden durch umfangreiche Analysen der gesammelten Daten unterstützt, die in dieser Arbeit vorgesehenen Nachweise zur Stützung des Abschlusses zur Verfügung stellen.

Abstract

In a relatively short period of time, the Internet has an amazing impact on almost every facet of our lives. With it, we are able to access new ideas, more information, unlimited possibilities, and a whole new world of communities. In addition, the demand for more and richer content has increased. To fulfill this demand, the Internet has evolved immensely in the last decade. Content become the king. Websites are becoming very rich in content as well as delivering high quality content to their customers. On top, introduction of social networking platforms such as Facebook, Twitter; video sharing sites such as Youtube has changed the user interaction with Internet by enabling them to publish their own content and share with each other which led to an exponential growth of Internet traffic.

To adopt this new trend, some companies are following state-of-the-art strategies for distributing their content while offering the best user experience. They have been deploying a large number of scalable and cost effective hosting infrastructure all around the globe. These hosting infrastructure can be composed of a few large data centers ,a large number of caches or any combination. Some of these hosting companies handle a major chunk of web traffic, making them the hyper giants of today's Internet. Other websites use the infrastructure of these hyper giants to deliver their content. Such a scenario causes a large amount of traffic to flow through hyper giants, thus creating a huge dependency between them and other websites.

This thesis addresses some of the common research questions to study how these websites rely on hyper giants. Firstly, whether there any presence of these hyper giants in today's Internet? Secondly, how the dependency of websites on hyper giants is evolving with time. Finding answers to these research questions help not only to other content distributing companies, content producers, content providers, and ISPs, but also to the research community at large.

In order to conduct this study, 100,000 top ranked websites of Alexa are used. DNS resolution is carried out on all these 100,000 websites along with their embedded links to analyze their second level domains that helps in determining the hosting infrastructure they use. HTTP header analysis is performed on these URLs that helps in finding out the reliance on hyper giants. The experimental results are supported by extensive analysis of data collected from a single vantage point in Germany. The results revealed that there are 26 hyper giants which are collectively contributing xyz percenatge of total web traffic on Internet. ////Put results for web objects

Contents

1 Introduction

The Internet is the largest network system in the world and is growing even bigger. Recent study [2] shows that, by 2015 there were more than 3 billion active Internet users in the world. With increase in Internet users and their demand for more and more richer content has led to exponential increase of Internet traffic. High resolution videos, graphic-rich multimedia on-line games, interactive audio and video, high quality audio streaming etc. are contributing to large upsurge in traffic. Consequently, websites having vast and rich content require different strategies to distribute their content all over the world in order to give their user a good quality of experience.

Quality of Experience (QoE) for end users is a key factor in the success of a new Internet service. Response time on web is a really important matter of concern. Longer loading times of websites will result in poor user experience. This will have a severe impact on the digital businesses. It is found out that slow loading websites cost the U.S. e-commerce market more than 500 billion dollar annually. According to e-commerce and conversion statistics, 40% of web users abandon a website if it takes more than 3 seconds to load [Ref: hostingfacts]. There are situations where sometimes, a website receives a huge amount of hits, resulting in the need to handle the peak loads. For this reason, websites host their content using cloud based infrastructure where the required resources can be scaled on-demand. The websites that are highly rich in content, in order to give their users a better experience host their content on content delivery networks. This brings the web content closer to the users, thus reducing the response time. Some other websites use content providers to host their content. Recent studies suggest [3,4] that some of these hosting infrastructures such as CDNs, Cloud services and Content providers are responsible for a major fraction of Internet traffic, making them hyper giants.

The appearance of hyper giants has influenced the Internet topology. Hyper giants construct peering links with different autonomous systems (ASs). In this way, hyper giants are able to reduce the transit cost of traffic traversing through large ISPs and able to serve content in faster way. The producers of the content (popular websites) want their content to be delivered to end user in less time for which they rely on hyper giants. Hyper giants build a large infrastructure all around the world to deliver content ensuring a faster response. The websites use these infrastructure to store their content, such as audios ,videos, test/html files etc. Such a scenario causes a large amount of traffic flow from hyper giants. This makes some websites reliant on hyper giants. Therefore these hyper giants can directly impact the way the web objects are delivered in today's Internet. Hence, it is important for the owners of these websites to know the degree to which they rely on hyper giants to deliver their content. It is this symbiosis between these two parties that motivates our work ,giving an overview on how far the reach of hyper giants in todays Internet.

By end of this thesis we are able to provide answers for some of the important research questions which can be summarized as follows:

- Identification of hyper giants: We propose a fully automated approach to discover hyper giants such as highly distributed content delivery networks, content providers, cloud computing services etc.

- Dependency of content delivery on hyper giants: We quantify the degree of content dependency of the popular websites on hyper giants by analyzing different web objects like text files,image files,application files delivered by hyper giants to popular web sites.

This remainder of this thesis is structured as follows.This thesis is separated into 7 chapters.

Chapter ?? :It starts with over view on the evolution of Internet architecture from early 2000s to current time and the techniques used to analyze the hosting infrastructure.

Chapter ??:discusses the methodology used to identify the hyper giants in today's Internet and the dependency between popular websites on hyper giants.

Chapter ??:discusses on the technologies used for implementing the web crawler. It describes design of the web crawler, followed by the work flow.

Chapter ??:Using the methods discussed in chapter-3, tests are conducted to extract the results that are required to analyze the hosting infrastructure deployed by the websites.

Chapter ??summarizes the results to identify the hyper giants and the dependency of popular websites on hyper giants.

Chapter ??discusses the problems encountered during the thesis work, learnings during this phase, any solutions to overcome the problems encountered. It also summarizes the results and includes possible future work.

2 Background

In this chapter,we discuss how Internet architecture evolved with time.Along with this we will discuss briefly on hyper giants.In addition to this we also provide the technical background of DNS and HTTP protocol which will be used in this thesis. This chapter gives an over view of evolution of Internet architecture with time.

The Internet architecture is getting complex with time with introduction of hyper giants. Therefore it is very important to identify them and study their role in Internet. In 2010, Craig Labovitz, then of Arbor Networks [2],first time characterized hyper giant. By placing Google in this list, the author described the hyper giant as, a content provider that makes massive investments in bandwidth, storage, and computing capacity to maximize efficiencies and performance. It is also found that the traffic amount sent by hyper giants is about 30% of the whole amount across the Internet. This concept of hyper giants also aligns with Schmidts [8] assertion which talks about "gang of four" companies which are responsible for the growth and innovation of Internet. Google, Apple, Amazon, and Facebook.In [check], Shavitt and Weinsberg analyzed changes in topological structure, such as betweenness centrality and link density, by focusing on large content providers, also referred to as hyper Giants [14, 21]. They create a snapshot of the AS-level graph from late 2006 until early 2011, and then analyze the interconnection trends of the transit and content providers and their implications for the Internet ecosystem. AS graphs are built by traversing IP traceroutes 9 and resolving each IP address to its corresponding AS. Shavitt and Weinsberg proved that large content providers like Google, Yahoo!, Microsoft, Facebook, and Amazon have increased their connectivity degree during the observed period and are becoming key players in the Internet ecosystem, strengthening the idea that the Internet is becoming flatter.From this analysis, it was found that the structure of the Internet topology has changed from a hierarchical to a flat structure. This is because large content providers construct links with a lot of small ISPs. Unfortunately all the above studies are carried out by accessing the Internet traffic passing through different ASs ,not traffic delivered by local ISP.Bernhard et al.[9,10] also worked on identifying and mapping the content infrastructure that are hosting the most popular content.Based on DNS measurements and BGP routing table snapshots, the author purposes a approach to discover hosting infrastructures. Poses et el[], Palacin et [13] defined hyper giants not only content providers, they are basically content aggregators. Small companies started using high speed infrastructures to deliver their content to end users.But with increase of content these high speed infrastructures started absorbing content from the long tail, entering fully into the niche of the traditional hosting companies.

2.1 Evolution of Internet Architecture and Rise of hyper giants

In 2010, Craig Labovitz, then of Arbor Networks [2],defined a new type of network entity he argued transcended traditional "content versus carrier" dichotomy of Internet architecture.By placing Google in this list,he characterized the hyper giant as a content provider that makes massive investments in bandwidth, storage, and computing capacity to maximize efficiencies and performance.The concept of hyper giants also aligns with Schmidt's assertion which talks about "gang of four" companies which are responsible for the growth and innovation of Internet. Google, Apple, Amazon, and Facebook [8].

The Internet architecture implemented until the early 2000s was based on a multi-tier hierarchic structure.Tier 1 ISPs were on top of the hierarchy followed by the Tier 2 regional ISPs and the Access ISPs at the lower part of the hierarchy connecting the end users. In this scheme, Tier 1 ISPs were highly connected to other ISPs and offered transit services to other ISPs in lower layers.Content was distributed through Access ISPs or, in the best cases, through ISPs located at advantageous points. Traffic flows were required to go up and then down in the hierarchy to reach end users shown in figure 1.Among the different network operators,Internet traffic was exchanged at different IP exchange points according to agreements between different layer players where the dis symmetry in traffic was compensated.

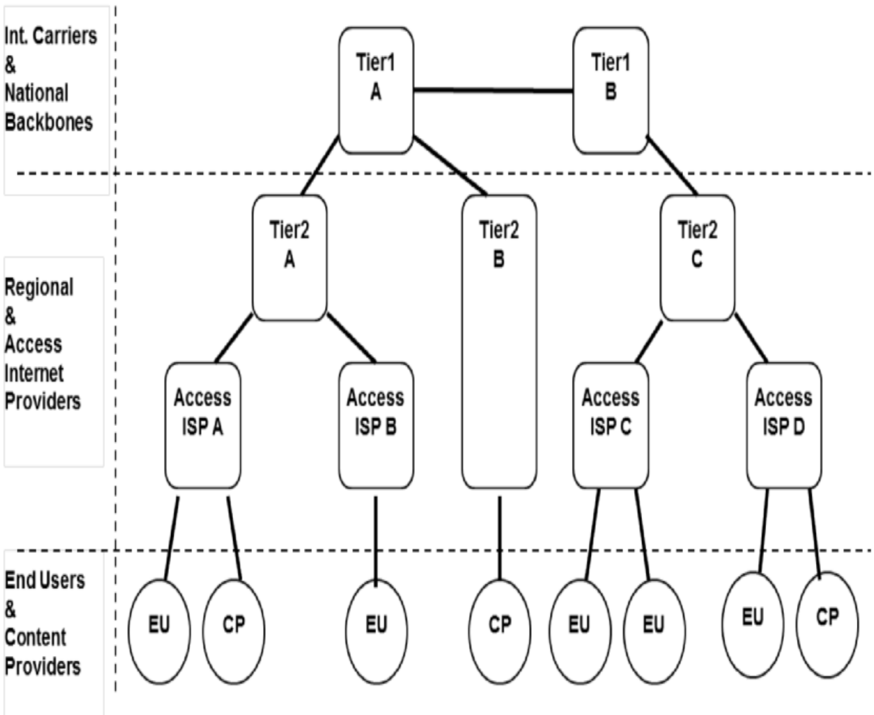

Figure 1: Traditional Hierarchic Internet Structure

But with the time,the Internet architecture has changed.Researchers found that now nobody has control over Internet ,instead each ISP has control over its network and depend upon the network connected with it.Even during last decade the old pyramidal structure of Internet architecture shown in figure-2 has been bypassed by big content providers, such as Google, Facebook, Amazon or Yahoo!, and content delivery network operators, such as Akamai.As a result now Internet's backbone has a flatter structure where there are few autonomous systems are playing major role in delivering content.They are connected to each other and have a big footprint by establishing small data centers all over the world.This help them to get as close as possible to the access networks used by their customers,bypassing intermediate Internet service providers.The trend towards flatter network architectures can also be found in the area of access networks.The researchers termed these infrastructure providers as "Hyper Giants" which include large content providers, such as Google and Yahoo, as well as highly distributed CDNs, like Akamai.

2.2 Content delivery Infrastructures

Recent traffic studies [3, 14] show that a large fraction of Internet traffic is due to content delivery and it originated by very few content delivery infrastructure (CDIs).Major CDIs include content delivery networks like Akamai,Cloudflare, content providers like Google,Microsoft,highly popular rich media sites like Youtube,Netflix and cloud comput-

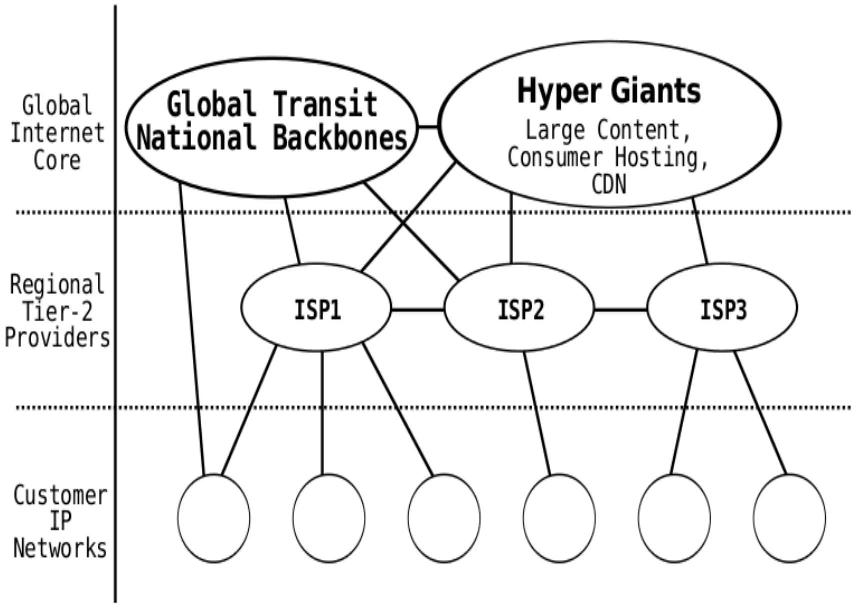

Figure 2: Modern Internet Structure

ing infrastructures like Amazon aws.Most of the CDIs have a large number of servers which located across the world.CDIs cache most of the popular content of the websites at these servers.Hence we a end user request for any content,CDIs deliver the content from the server nearest to the end user.In this way CDIs reduce the load on origin servers and at the same time improve performance for the user. CDIs follow different strategies for redirecting traffic to nearest cache server of the end user.Most of the CDIs use DNS to translate the host name of a web site request into the IP address of an server.During this translation ,DNS takes into account the location to the end user,the location of the nearest CDIs cache server ,load of the server etc.

Independent CDIs are normally referred to as CDNs.CDNs have a large number of servers all around the world and mainly responsible for delivering content of their customers to end users.Leighton [1] proposes four main approaches to distributing content in a content-delivery architecture: (i) centralized hosting,(ii) big data center based CDNs, (iii) highly distributed CDNs, and (iv) peer-to-peer networks.In centralized hosting case ,traditional architectures sites take help of one or small number of collocation sites to host their content.These centralized hosting structures are may be enough for small content sites but not for popular websites which carries huge amount of content.Big Data Center content delivery networks have Hugh number of high-capacity data centers which are connected to major backbones.Highly popular content are cached.Hence able to increase the performance of delivery compare to centralized hosting infrastructures but still are limited in potential improvements as still they are far away from the end user.Third type of model is highly distributed content delivery networks.They have high footprint all over the world.By putting their own infrastructures inside end user's ISP,they are able to eliminating peering, connectivity, routing, and distance problems, and reducing the number of Internet components.Final approach is peer to peer networks which has very little scope in delivering the content of popular websites in today's Internet world due to serious concern of the copy right issues.

Cloud infrastructure refers to the hardware and software components, such as servers, storage, networking and virtualization software that are needed to support the computing requirements of a cloud computing model. In addition cloud computing infrastructures include a software abstraction level which virtualizes resources like servers, compute, memory, network switches, firewalls, load balancers and storage. and logically presents them to users through programmatic means.Cloud infrastructure mainly present three different types of model:infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS).Cloud infrastructures deploy a large number of data centers at certain regions of the world.In case of infrastructure as service or IaaS,cloud infrastructures give access to these data centers to their users.Users can able to access and manage remote data center infrastructures, such as compute (virtualized or bare metal), storage, networking, and networking services (e.g. firewalls).SaaS uses the web to deliver applications that are managed by a third-party vendor and whose interface is accessed on the client's side.Popular SaaS offering types include email and collaboration, customer relationship management, and health care-related applications. Paas is used for applications, and other development, while providing cloud components to software. With this technology,users can manage OSes, virtualization, servers, storage, networking, and the PaaS software itself.

Content Providers also are major player in content delivery infrastructure.Companies like Google, Facebook, Netflix etc. build their own infrastructures like data centers and interconnected them with high speed backbone networks to deliver some of their very popular services.Google connects its data centers to a large number of ISPs via IXPs and also via private peering [15].Google also now provide Google Global Cache (GGC) [16] as a service where customers can optimize network infrastructure costs associated with delivering Google and YouTube content to end users by serving this content from inside their ISP networks.Through GGC,small ISPs and which are located in areas with limited connectivity can reduced the transit cost as well as websites can deliver their content with better performance.GGC also allows an ISP to advertise through BGP the prefixes of users that each GGC server should serve.The Netflix system known as Open Connect Network[17].Netfix deploy its own infrastructure inside a lot of ISPs by partnering with them to deliver its own content more efficiently.

2.3 Protocols

In this section we will discuss about the protocols used in this thesis which are domain name system (DNS) and hyper text transfer protocol (HTTP).Both protocols used in out thesis extensively to get CNAMEs of a website and to get the http header information respectively.

2.3.1 Domain name System (DNS)

Domain name system (DNS) is used to translate IP address to corresponding host names.Internally it is maintaining a hierarchal structure of domains.Before the invention of DNS on year 1983,a simple text file (hosts.txt) file was used to do this translation from IP address to host name.But with a growing number of host names it was difficult to keep maintain in hosts.txt file and Domain name system introduced.

The administration of domains is divided into different zones. The zone information is distributed using authoritative name servers.The top most level of DNS starts with root zone and the root zone information is served by root servers.Responsibility of specific parts of zone can be given to some other authoritative name servers which in turn divided responsibility with other authoritative name server.For, e.g.,the responsibility of .org domain is delegates to the Pub- lic Interest Registry by the root zone which in turn delegates responsibility for acm.org to the Association for Computing Machinery (ACM).At the end its site is responsible for its own zone and keep maintain its own database of authoritative name server.The information about a particular domain of a zone is kept in Resource Records (RRs) which specify the class and type of the record as well as the data describing it.Multiple RRs with the same name, type and class are called a resource record set (RRset).

To resolve a IP address into host name,the procedure starts with the end user's stub resolver queries to local name server called caching server.if caching server can not able to resolve it,it redirects the query to authoritative name server of the domain.If resolver does not know how to contact the corresponding authoritative name server of the domain,it redirects the query to root name server .The root name server again refers the resolver to the authoritative name server responsible for the domain just below the root server.This procedure continues till resolver is able to resolve the domain properly.

```
; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.bmw.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11656
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.bmw.com.                   IN      A

;; ANSWER SECTION:
www.bmw.com.            3600    IN      CNAME   cn-www.bmw.com.edgesuite.net.
cn-www.bmw.com.edgesuite.net. 3600 IN   CNAME   a1586.b.akamai.net.
a1586.b.akamai.net.     20      IN      A       104.121.76.73
a1586.b.akamai.net.     20      IN      A       104.121.76.64

;; Query time: 22 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Wed Oct 05 21:32:17 CEST 2016
;; MSG SIZE  rcvd: 143
```

Figure 3: DNS Reply for a host using dig command line tool

Figure-3 shows the DNS reply by the resolver when querying a host name served by a content infrastructure.Here the host name is www.bmw.com.The answer section contain a chain of CNAMEs which resolve into two ARecord set (RRset) with different IP addresses which can be for used for load balancing.

2.3.2 Hyper text transfer protocol(HTTP)

Hyper text transfer protocol (HTTP) is an application layer protocol mainly used as defector standard to transport content in world wide web.HTTP works on top of the TCP/IP protocol and follows the client server architecture via request-response communication procedure.It allows end-users to request, modify, add or delete resources identified by

Uniform Resource Identifiers (URIs).

```
HTTP Request

GET / HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0 [...]
Accept: text/html [...]
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-alive

HTTP Response

HTTP/1.1 200 OK
Accept-Ranges: bytes
Content-Type: text/html
Date: Mon, 27 Jul 2009 12:28:53 GMT
Server: Apache/2.2.14 (Win32)
Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT
Content-Length: 88
<!doctype html>
<html>
[...]
```

Figure 4: HTTP Request (top) and Response (down) for example.com

HTTP message consists of HTTP header which shows the meaning of message and HTTP body which is actual message.HTTP message can be a request message or response message.The HTTP client sends a request message to server .There are different types of methods used in HTTP request message like GET,HEAD,POST ,PUT,DELETE,CONNECT etc.But in this thesis we have used extensible GET method and the HEAD method.The GET method is used to retrieve information from the given server using a given URI. Requests using GET should only retrieve data and should have no other effect on the data.Same as GET, but it transfers the status line and the header section only.The introductory line in an HTTP request shown in figure 4 consists of a method, a server-side path, and the HTTP version in use.The introductory line in an HTTP response shown in figure 4 starts out with the HTTP version in use, followed by a standardized three-digit status code and a textual status description. The status code tells the requester about the success of the query or indicates the reason of an error.Both request and response messages are followed by multiple header lines.Some header information are valid for request ,some are for response and some are valid in both the ways.Since HTTP1.1,the Host header is mandatory for request messages.The meta information encompasses information about the file type, the character set in use, preferred language etc.HTTP also allows server to set cookies in client side which help the server to track client requests.

2.4 Conclusion

Within last decade the Internet architecture changed vastly due to the introduction of hyper giants which can be highly distributed CDNs,cloud computing CDNs etc.Todays Internet traffic is dominated by HTTP traffic.Again to deliver the content fast ,DNS protocol is used as the load balancing mechanism by these big hyper giants.

3 Methodology

This section discusses the methods to identify the presence of hyper giants in today's Internet and to find out the inter dependency between hyper giants and popular websites. It is commonly known that most of the websites use some kind of hosting infrastructure for distributing their Internet content. This hosting infrastructures can be CDN, cloud computing infrastructure or content providers. Hosting infrastructures were designed to transport and cache large amounts of Internet content, such as HTML code, JavaScript, large files, images, audio, and video. The most appropriate way to identify whether a website is using a hosting infrastructure service is to inspect its HTML code looking for URLs linked to hosting infrastructure. The redirection of a link to an external hosting company will be clear evidence that a particular website is using a hosting infrastructure.

To perform this task, as shown in figure, 100000 top ranked websites of Alexa will be crawled using scrapy engine and their HTML codes are inspected to retrieve all their embedded URLs.Then these URLs will be used to find out the link redirection and subsequently the hosting infrastructure. DNS resolution is the known solution to find out the link redirection. It will resolve the web URL into single or multiple IP addresses and their corresponding ARecord names. ARecord names show the hosting infrastructure linked to corresponding URL. Sometimes hosting companies use different naming convention to represent their ARecords. As an example, a1586.b.akamai.net. and a1586.a.akamai.net are the ARecord names when www.bmw.com is resolved. This might be because of load balancing the servers that are used to cache content. Therefore, to get the desired hosting infrastructure ,the second level domain (SLD) of each ARecord is the best suited option.

The set of IP addresses for a particular SLD shows the degree to which the corresponding hosting infrastructure is connected with different web URLs. The number of BGP prefixes show the network footprint and the number of ASNs show how the infrastructure is distributed. Therefore, to identify the hyper giants, the natural choice for the features to consider are IP addresses, AS Numbers and the BGP prefixes.

Hyper giants build a large infrastructure all around the world to deliver content ensuring a faster response. The websites use these infrastructure to store their content, such as audios ,videos, test/html files etc. Therefore, any kind of disruption in serving these contents from hyper giants will impact the websites which shows the interdependency of hyper giants with the web sites. To know which content type will be impacted more, the type of content can be studied. To evaluate the type of content,as shown in figure, HTTP header analysis will be carried out on 100000 top ranked websites of Alexa and their corresponding embedded URLs. The web object type retrieved from the HTTP header information determines the content type such as text/html, image, video, audio etc., which will be used to analyze the interdependency between hyper giants and these websites.

3.1 Identification of hyper giants

To find out the hyper giants in Internet,the hosting infrastructure which are serving popular websites will be observed. This can be determined by analyzing different features of hosting infrastructure such as IP addresses, BGP prefixes, ASN numbers etc. The methods to find out these features will be discussed in the following subsections.

3.1.1 Hosting infrastructure to BGP Prefix Mapping

From DNS resolution, the hosting infrastructure and corresponding IP addresses are determined. To find out BGP prefix routes of a particular IP address, BGP routing information

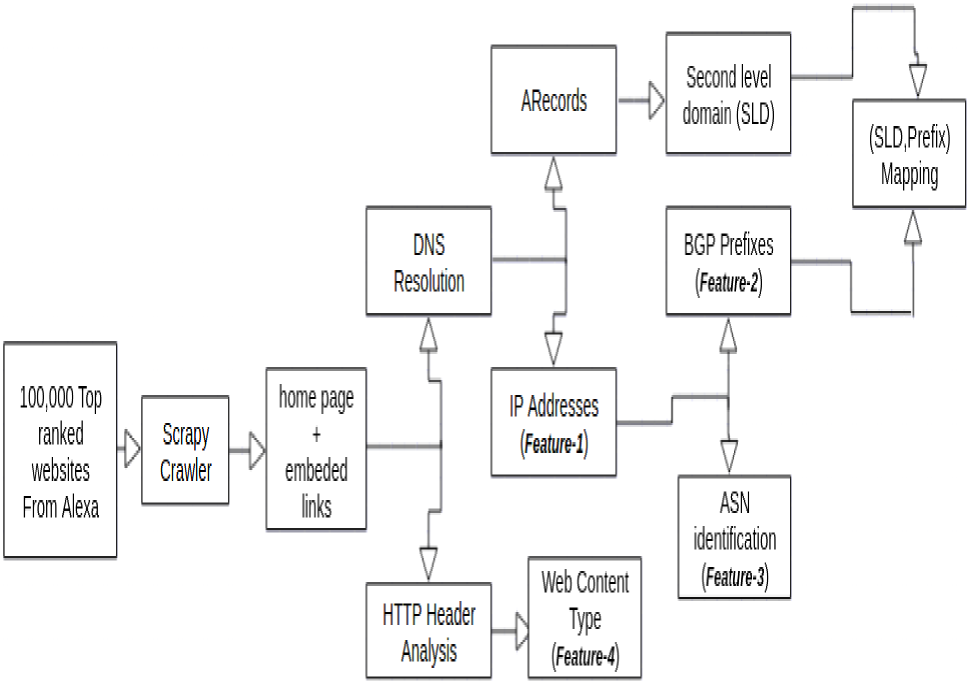

Figure 5: High level approach Part-1

from RIPE RIS [23] will be used. In figure-7, "route" shows the BGP prefixes of a IP address. From the figure this can be observed that the IP address 104.121.76.73 can be routed via two different prefixes 104.64.0.0/10 and 104.121.76.0/24. So both the prefixes for this IP will be considered for further analysis. This procedure will be carried out for each IP addresses of hosting infrastructure which will result in (hosting infrastructure, prefixes) mapping.

3.2 Clustering

From the initial analysis on the data collected for analyzing hosting infrastructure, it is found out that some hosting infrastructure are administered by the same company. For example, both the hosting infrastructure akamaiedge.net and akamai.net are administered by the same company, Akamai. Similarly google.com, googleusercontent.com, google-hosted.com and googledomains.com are administered by same company, Google. Generally companies use different ARecord names depending on the type of service they offer and the geographical location from where they offer the services. It is observed that Amazon uses two different ARecord names, namely amazonaws.us-east-1.elb.amazonaws.com, eu-west-1.elb.amazonaws.com for its customers from two geographical locations. It has to be analyzed, whether these ARecord names are pointing to same infrastructure or different. If they are pointing to the same infrastructure, they should be considered as one, else separately. This can be identified by using clustering algorithm by analyzing the BGP prefixes they share.

The clustering algorithm takes a two step approach. In the first step, the prefixes of the hosting infrastructure will be aggregated into set of parent prefixes as explained in 3.2.1. In the subsequent step, the parent prefixes of the hosting infrastructure will be compared with each of the remaining infrastructure for clustering as explained in section

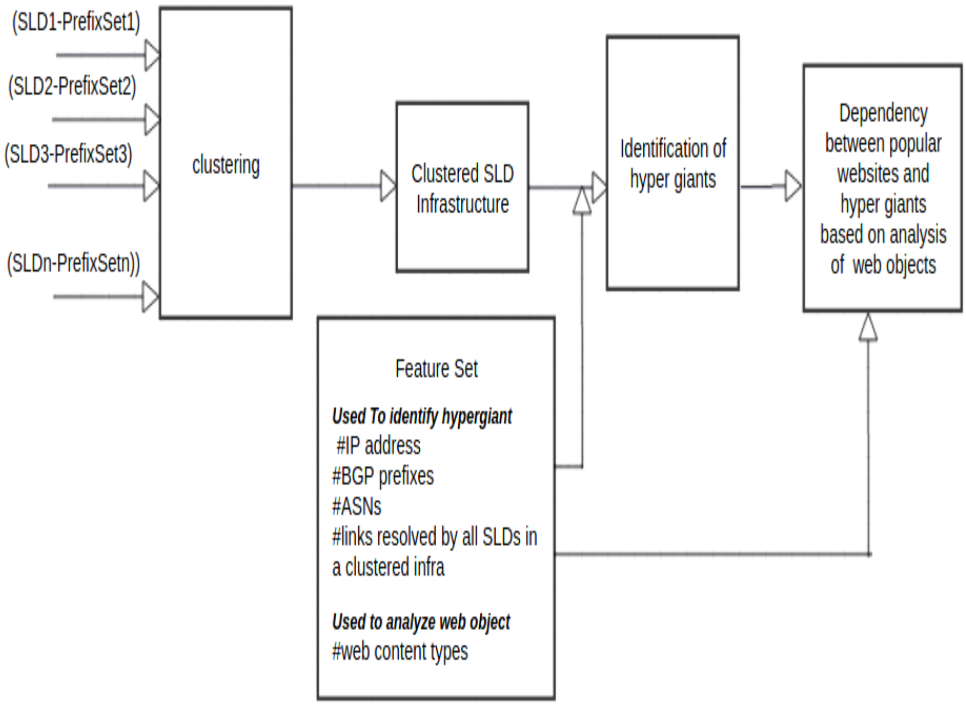

Figure 6: High level approach Part-2

3.2.2.

3.2.1 Prefix Aggregation of hosting infrastructure

From the section 3.1.2, hosting infrastructure mapped to corresponding BGP prefixes are collected. But in this mapping, it is observed that some of the BGP prefixes are subset of other BGP prefixes. For example, googledomains.com has the prefix set '216.239.32.0/19', '216.239.32.0/24', '216.239.34.0/24', '216.239.36.0/24' and '216.239.38.0/24'. Here the prefixes '216.239.32.0/24', '216.239.34.0/24', '216.239.36.0/24', '216.239.38.0/24' are subnet of prefix '216.239.32.0/19'. Hence it is ideal to aggregate them. This way all the child prefixes in each mapping are aggregated to their parent prefixes. By end of this method, each hosting infrastructure will be mapped to a set of parent prefixes. For example, the hosting infrastructure googledomains.com is mapping to the parent prefix '216.239.32.0/19', as all child prefixes got aggregated into '216.239.32.0/19' parent prefix. This procedure is repeated for all the other infrastructure mappings, which generates a complete list of hosting infrastructure to their corresponding aggregated BGP prefixes.These mapping are sorted in the decreasing order of links hosting by each of these hosing infrastructure.The motivation behind doing this, is to allow the merging of smaller infrastructure into the larger one depending on their similarity factor (sim(s1, s2)) which is discussed in next section.

3.2.2 Evaluation of Similarity between two hosting infrastructure

The possibility of clustering any two hosting infrastructures is determined by analyzing the similarity between their aggregated prefixes. This can be done by verifying if one prefix is a parent to the other. While comparing two hosting infrastructures, if any of the hosting

```
% This is RIPE NCC's Routing Information Service
% whois gateway to collected BGP Routing Tables, version2.0
% IPv4 or IPv6 address to origin prefix match
%
% For more information visit http://www.ripe.net/ris/riswhois.html
%
% Connected to backend ris-whois06.ripe.net

route:          104.64.0.0/10
origin:         AS35994
descr:          AKAMAI-AS - Akamai Technologies, Inc., US
lastupd-frst: 2016-10-04 11:53Z  198.32.176.70@rrc14
lastupd-last: 2016-10-07 11:29Z  196.46.25.29@rrc19
seen-at:       rrc00,rrc01,rrc03,rrc04,rrc05,rrc06,rrc07,rrc10,rrc11,rrc12,
               rrc13,rrc14,rrc15,rrc16,rrc18,rrc19,rrc20,rrc21
num-rispeers: 161
source:         RISWHOIS

route:          104.121.76.0/24
origin:         AS20940
descr:          AKAMAI-ASN1 , US
lastupd-frst: 2016-07-06 10:09Z  198.32.124.146@rrc16
lastupd-last: 2016-10-07 11:29Z  196.46.25.29@rrc19
seen-at:        rrc00,rrc01,rrc03,rrc04,rrc05,rrc06,rrc07,rrc10,rrc11,rrc12,
                rrc13,rrc14,rrc15,rrc16,rrc18,rrc19,rrc20,rrc21
num-rispeers: 163
source:         RISWHOIS
```

Figure 7: RIPE RIS bgp prefixes using whois command

infrastructure contain child prefix of other, then the child prefix will be replaced with the parent prefix. This will make two prefix sets homogeneous, thus helping in evaluating the degree of the similarity between them. For comparing two hosting infrastructures, the similarity equation is defined as follows,

The similarity factor sim(s1,s2) is defined as follows,

$$sim(s1, s2) = \frac{|s1 \cap s2|}{|s2|} \tag{1}$$

where s1,s2 are the bgp prefix sets.

If the similarity between two prefix sets are greater than equal to the 70%, then both the hosting infrastructure will be clustered together making them a single clustered infrastructure. It is assumption that, s2 can be clustered with s2, provided 70% of s2's prefixes are from that of s1's. This procedure will be repeated for all the hosting infrastructure. If two infrastructure has similarity factor greater than equal to 0.7, then the child hosting infrastructure will be merged into the parent hosting infrastructure and will be excluded from further comparisons. For example 'googleusercontent.com' matched with 'google.com' with similarity factor 1. This does not mean that both these hosting infrastructure has the same set of prefixes, but it means that, 'googleusercontent.com' shares 100% of its prefixes with 'google.com'. Once this is matched, 'googleusercontent.com' will not be available for any further similarity matching with any other hosting infrastructures. It is assumed that a similarity index of grater than equal to 0.7 served as a good measure to determine the degree of similarity between two hosting infrastructure which can be considered as future work for extensive analysis.A sample example is explained as follows,

Let (hosting infrastructure,Prefix set)s are,
(SLD1,[10.0.0.1/24,12.0.0.0/16,192.168.3.0/24])
(SLD2,[192.168.0.0/16,10.0.0.0/16,12.0.0.0/24])
(SLD3,[3.0.0.0/8,12.0.0.0/24,192.168.3.8/24,5.0.0.0/16])
(SLD4,[4.0.0.0/24,6.0.0.0/24,10.0.0.0/16]) (SLD5,[6.0.0.1/16,10.0.0.0/8])
After the clustering procedure, SLD2 and SLD3 will clustered with SLD1 creating (SLD1,(SLD2,SLD3))mapping and SLD5 will be clustered with SLD4 to create (SLD4,SLD5) mapping.

3.2.3 Evaluation of hyper giants

Once the clustering procedure is performed, a list of parent hosting infrastructure to child hosting infrastructure are available for further analysis. To find out hyper giants presence, the features of each clustered hosting infrastructures will be analyzed. The features include number of links, number of IP addresses, number of BGP prefixes, number of AS numbers. To find out each of these features for a clustered hosting infrastructure, corresponding child hosting infrastructures will be considered. Hence total number of links for a clustered hosting infrastructure will be the sum of all the links served by all child hosting infrastructures. To find out number of BGP prefixes,the set of two prefix sets will be taken. This will result the unique BGP prefixes. Similarly to get the AS numbers, the set of two prefix sets will be taken.

Let, p = numner of links served by SLD1 q = number of links served by SLD2 r = number of links served by SLD3
and mapping is, p ¿ (q,r) which means SLD2 and SLD3 are clustered with SLD1 then number of links of the clustered hosting infrastructure is p + q + r (Links are counted irrespective of their repetitiveness in other SLDs)
x = numner of IP addresses served by SLD1 y = number of IP addresses served by SLD2 z = number of IP addresses served by SLD3
and mapping is, x ¿ (y,z) which means SLD2 and SLD3 are clustered with SLD1 then number of IP addresses of the clustered hosting infrastructure is set(x,y,z) (Non repetitive IP addresses)
Let, i = numner of bgp prefixes served by SLD1 j = number of bgp prefixes served by SLD2 k = number of bgp prefixes served by SLD3
and mapping is, i¿(j,k) which means SLD2 and SLD3 are clustered with SLD1 then number of BGP prefixes of the clustered hosting infrastructure is set(i,j,k) (Non repetitive bgp prefixes)
Let, a = number of ASs served by SLD1 b = number of ASs prefixes served by SLD2 c = number of ASs prefixes served by SLD3
and mapping is, a-¿(b,c) which means SLD2 and SLD3 are clustered with SLD1 then number of ASNs of the clustered hosting infrastructure is set(a,b,c) (Non repetitive ASNs)

The features of clustered hosting infrastructure will be analyzed further to find out the hyper giants.

To identify the hyper giants, the hosting infrastructures with unique behavior will be separated from the others. To perform this, two main features will be taken, number of links and number of IP addresses. k-means algorithm [26] will be performed on clustered hosting infrastructure to partition the clustered hosting infrastructure in up to k clusters .The cluster co-efficient value k is chosen as 10.Clusters whose features have high values will be clustered together. On the other hand, smaller infrastructures that use very few links and IP addresses are not sufficiently different, and therefore, can be found in the same cluster.To identify the hyper giants, the clustered hosting infrastructure which clustered separately than the most of the other clustered hosting infrastructure will be considered.

3.3 Web objects delivered from hyper giants to popular web sites

To find out different web objects delivered from hyper giants to popular web sites, HTTP header information for all URLs corresponding to each clustered hosting infrastructure is analyzed. From header information, the content type of each object can be extracted. This information is useful to determine what type of web objects are served by each hyper giant.

3.4 Conclusion

This section discussed about the methods to identify the presence of hyper giants in today's Internet and to find out the inter dependency between hyper giants and popular websites. To execute both the methods,IP addresses, AS numbers, BGP prefixes are used as features for categorical analysis of clustered infrastructure. The section provided the procedures to find out each of the features for hosting infrastructures. At the end, HTTP header information is used to determine the type of web objects served by each hyper giant.

4 Implementation

This chapter discusses the implementation aspects of web crawler. It explains the choice of programming languages, development tools used to implement the crawler engine. Furthermore,emphasis is given on explaining the internal architecture behind the crawler engine,along with the details of functionality performed by each module. Focus is also given on discussing the overall work flow of the crawler engine.

4.1 Choice of programming language

Before going for implementation, it is important to know that the whole process involve two major parts. In the first step, identification of hyper giants will be performed using different features like number of links, number of IP addresses, BGP prefixes etc. In the second step the dependency between hyper giants and popular websites will be discussed based on number of web objects delivered from hyper giants to popular websites. Hence a programming language which is powerful in development as well as data analysis will be chosen. So that any kind of dependency between two steps can be handled easily. Again language should be free, open and usable by anyone who wishes to run this crawler implementation in future. Moreover the language should have good number of open libraries, big community and should have a lot of online documentation. For this purpose python language is chosen for the implementation as it fulfills all necessary requirement. For data analysis pyspark (Python version of spark) will be used.

4.2 Development Tools

For development, it is important to choose correct IDE which allows to write code in less time with minimum effort.Along with this it helps in code completion, syntax highlighting, refactoring. For this purpose "sublime editor" which is free and easy to write python code is chosen.

Now second most important aspect is to choose python web crawling tool which will suit the requirement and can be used in future. There are couple of famous web crawling tools available like urlib, beautiful soup, scrapy etc. But Python Scrapy is the best out there. Scrapy crawling is faster than any other platforms, since it uses asynchronous operations (on top of Twisted). Scrapy has better and faster support for parsing (x)html on top of libxml2. Scrapy is a mature framework with full unicode, redirection handling, gzipped responses, odd encodings, integrated HTTP cache etc. Again it is open source having a big community and documentation.

There are lot of different machine learning tools available but as it is expected that resultant data will be order of some gigabytes, it is better to use that tool which can process data faster.Python and R are popular languages for data analysis due to the large number of modules or packages that are readily available. But traditional uses of these tools are often limiting, as they process data on a single machine where the movement of data becomes time consuming, the analysis requires sampling (which often does not accurately represent the data), and moving from development to production environments requires extensive re-engineering. Spark provides a powerful, unified engine that is both fast and easy to use. Hence Pyspark will be used for initial data analysis and further analysis will be carried using python and RStudio.

Apart from above tools, some other tools used for the implementation. The following are the list of important softwares and tools used.

- dnspython:it is a DNS toolkit for python.This is used in the code to get the A records of hosts.
- pygeoip :The libarary is used to get the ASN numbers associated with IP addresses.This library is based on Maxmind's GeoIP C API.
- urlparse :this is used to convert a relative url to an absolute url.
- Public suffix List :This is the collection of all registered host names given by all internet users.The Public Suffix List is an initiative of Mozilla, but is maintained as a community resource.It is available for use in any software, but was originally created to meet the needs of browser manufacturers.In our code we use it to get second level domain from a host name.The "effective_tld_namesdat" is the file which is free downloadable from their site.
- IPython :IPython notebook is used for pyspark code writing and execution.
- matplotlib :Python library used to for making different graphs used in this thesis.
- pygal :Pygal is also another python library which we used to make graphs.

4.3 Design of Crawler Engine

In this section,the internal architecture behind the crawler engine will be discussed.The design process involves two parts,first one is the crawler engine which takes 100,000 top ranked websites of Alexa as input,crawl the websites, return the result file as output from the engine and second part is processing of result file using data processing engine which internally use "pyspark" to clean up the crawled data.

4.3.1 Crawler Engine

In this section, the internal architecture of the crawler engine will be discussed. The main objective of crawler engine is to take input data in the format of text file which contain the top 100,000 top ranked websites of Alexa. Then divide this master domain list into multiple sub domain lists with equal weighted websites and process each sub domain list using separate crawler instances. Each crawler instance work independently and store all website information like HTTP header information, DNS resolution analysis etc. in result file. The core of this cralwer engine is "Scrapy" which is used for crawling the website links. In the following sections the architecture of scrapy framework, the main data types of scrapy framework used for implementation will be discussed. In the subsequent section, the overall work flow of the crawler engine will be discussed.

4.3.2 Scrapy Framework

Scrapy framework is one of top ranked open source project, a fast web crawling framework, used to crawl websites and extract structured data from their pages. It provide option for both focused and broad crawling.In case of focused crawler scrapy crawls a specific domain while in case of broad crawling a large (potentially unlimited) number of domains can be crawled. Hence for the implementation broad crawling will be used. As can be seen from figure-7,the main components of the framework contain scrapy engine,scheduler,downloaders,spiders,item pipeline,downloader middlewares.

- Scrapy Engine : The engine is responsible for controlling the data flow between all components of the system, and triggering events when certain actions occur.

- Scheduler : The Scheduler receives requests from the engine and enqueues them for feeding them later (also to the engine) when the engine requests them.
- Downloader : The Downloader is responsible for fetching web pages and feeding them to the engine which, in turn, feeds them to the spiders.
- Spiders : Spiders use for to parse responses and extract items from them or additional URLs (requests) to follow.
- Item Pipeline : The Item Pipeline is responsible for processing the items once they have been extracted (or scraped) by the spiders. Typical tasks include cleansing, validation and persistence (like storing the item in a database). For more information see Item Pipeline.
- Downloader middlewares : Downloader middlewares are specific hooks that sit between the Engine and the Downloader and process requests when they pass from the Engine to the Downloader, and responses that pass from Downloader to the Engine.

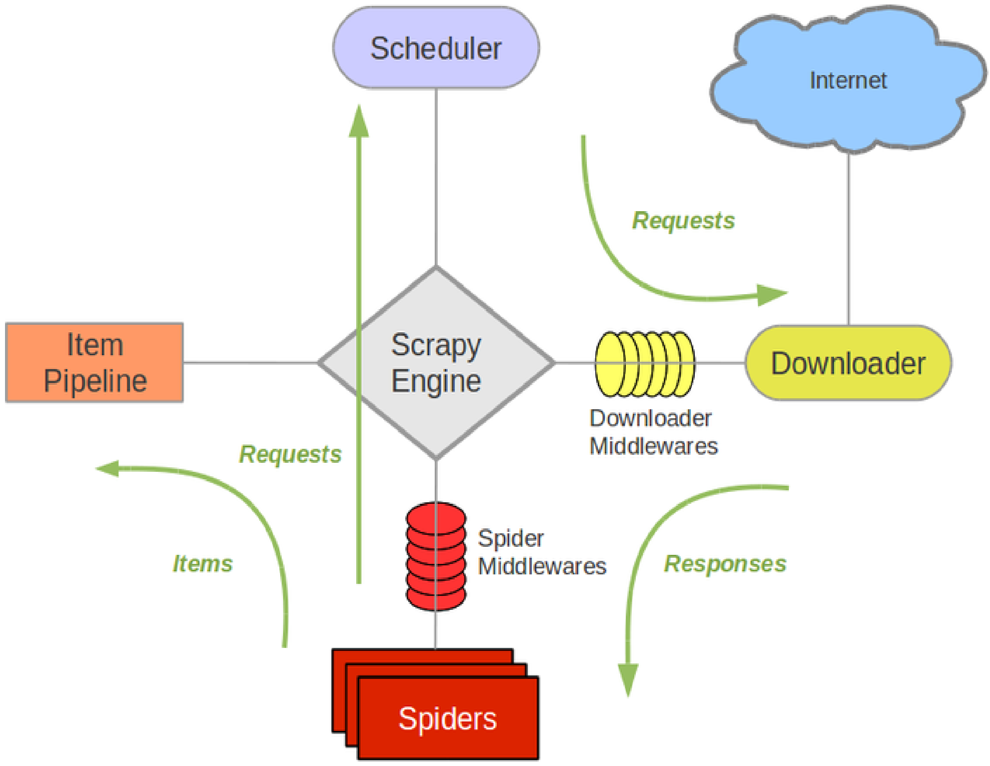

Figure 8: Scrapy Architecture

4.3.3 Scrapy data types

Scrapy uses some specific classes and strings which are used to crawl data.These are written in python and are open source.Scrapy also use twisted framework for multi threading.Some of the classes which are going to be used for implementation, are as follows,

1. Spider :Spiders are the classes used to crawl single or multiple domains using different methods like start_requests(),parse().

- start_requests() :This scrapy method is used to pass domains to parse method for further crawling process.
- parse() :he parse method is in charge of processing the response and returning scraped data and/or more URLs to follow. Other Requests callbacks have the same requirements as the Spider class.

2. Twisted Framework:Twisted supports an abstraction over raw threads — using a thread as a deferred source. Thus, a deferred is returned immediately, which will receive a value when the thread finishes. Callbacks can be attached which will run in the main thread, thus alleviating the need for complex locking solutions. A prime example of such usage, which comes from Twisted's support libraries, is using this model to call into databases. The database call itself happens on a foreign thread, but the analysis of the result happens in the main thread.

3. Requests objects:Typically, Request objects are generated in the spiders and pass across the system until they reach the Downloader, which executes the request.A Request object represents an HTTP request, which is usually generated in the Spider and executed by the Downloader, and thus generating a Response.

4. Response objects :Request object returns a Response object which travels back to the spider that issued the request.A Response object represents an HTTP response, which is usually downloaded (by the Downloader) and fed to the Spiders for processing.

4.4 Work flow

In this section the overall work flow of the crawler engine will be discussed.The crawler engine is used to perform multiple operations.It take input data in the format of text file which contain the top 100,000 top ranked websites of Alexa. Then divide this master domain list into multiple sub domain lists with equal weighted websites and process each sub domain list using separate crawler instances. Each crawler instance work independently and store all website information like HTTP header information, DNS resolution analysis etc. in result file. The core of this cralwer engine is "Scrapy" which is used for crawling the website links. The main focus is to create multiple instances of crawler engine. As scrapy is a memory greedy tool,after intensive testing it is decided to create 20 parallel threads which work independently. Each instance of the crawler will take separate input file which contain 5000 domains. Master domain list should be divided properly so that all the crawlers will complete their crawling in almost same time. Hence multiple sublists with equal weight will be created where weight refers to the rank of the websites provided by Alexa. The master domain list contain the website links and the rank of the websites given by Alexa. So after division the first sublist will contain the websites of 1st rank, 21st rank, ..., 99981st rank, second list will contain 2nd rank, 22nd rank, ..., 99982nd rank websites and so forth for other instances.After the division each sublist will contain 5000 websites with nearly equal weight of website ranks. Each sublist will be sent as input to multiple instances of crawler.

In the subsequent step each crawler instance will take 5000 domains and parse one by one domain. For each URL crawler engine downloads the corresponding web page, extract the linked URLs, and check each url to see whether the extracted url is a fresh url which has not already been seen . With this architecture a very large number of independent crawls of the white listed domains obtained from Alexa can be crawled. Along with this, the crawler engine also extract HTTP header information, ARecord details ,ASN details.

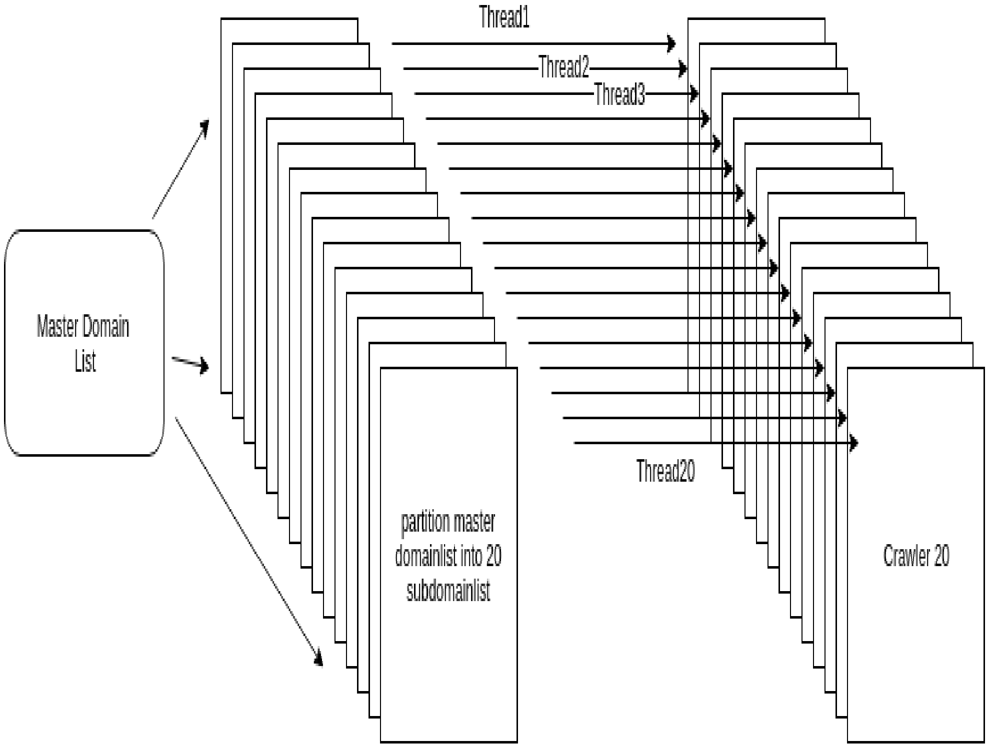

Figure 9: Pre Creawling Step

4.5 Conclusion

This section discusses the softwares used to implement the required functionality. The motivation behind choosing different languages,software tools used during the implementation are discussed. The internal architecture of the crawler engine used for crawling 100,000 top ranked websites is also explained. In the next section the hosted collected traces, in order to evaluate our methodology.

5 Measurement

In order to identify the presence of hyper giants in today's Internet and to find out the degree of dependency on hyper giants to delivery Internet web content.In order to study this, the 100,000 top ranked websites of Alexa are crawled. To find out the hyper giants, the hosting infrastructures which are serving popular websites are observed. DNS resolution and HTTP header information also extracted to identify the features related to each website link. The features are number of links served by each SLD infrastructure, number of IP addresses, number of BGP prefixes for each IP address and number of AS numbers. These features are used to cluster the SLD infrastructures following the methodology discussed in chapter-3. This chapter explains the overview of traces which are collected after each step staring from web crawling to the last step, web object identification.

5.1 Web Crawling

To obtain a good coverage of the hosting infrastructure, the 100000 top ranked websites from Alexa [26] are crawled. Alexa ranks websites based on Internet traffic-users of its tool bar for various web browsers like Google Chrome, Internet explorer, Firefox . Moreover, websites contain a lot of embedded links to images, videos, advertisements etc. which the browser of the user has to download from the various web servers. These embedded links can be from different hosting infrastructures. In our study, such embedded links has to be taken into account, as it might be served from servers other than those serving the home page. To give better understanding, while crawling facebook.com, the home page is served from Facebook data centers while the logo and other meta data come from Akamai. DNS resolution and HTTP header analysis is carried out on each web link including their corresponding embedded links to extract IP addresses, ARecord name information. The scrapy crawler queries the HTTP Get method to local DNS resolver for all the website links and store the results in a trace file.

Three sample traces are given below,

```
[55017, 0, 200, 'text/html', 19719, 'http://parsquran.com', 'a', 'parsquran.com', '74.208.215.213', 8560]
[55017, 1, 200, 'text/html; charset=UTF-8', 0, 'http://parsquran.com/site/sitemap.html', 'a', 'parsquran.com', '74.208.215.213', 8560]
[21794, 0, 200, 'text/html', 2014, 'http://gomap.az', 'a', 'gomap.az', '85.132.44.164', 29049]
```

Each trace contain 10 different features, as described below.

1. index: index shows the rank of the website
2. depth_level: 0 or 1. 0 shows that the website is the main page url and 1 shows the embedded links inside main page
3. httpResponseStatus: the HTTP return status code.
4. MIMEcontentType: this is included to know the type of element inside the web page.
5. content_length: content length gives the idea about the size of the element. The content type only extracted for the home page link.

6. URL: this is the URL to be crawled by the scrapy engine.This URL can be main page URL or embedded links.Duplicate links are omitted for the same main page by using RFPDupeFilter.RFPDupeFilter is a scrapy class which is used to detect and filter duplicate requests [3].

7. tagType :this shows the HTML element type. for example if an element is embedded in a website like ”¡img class=”desktop” title=”” alt=”” src=”img/bg-cropped.jpg”¿”,then the tagtype will be ”img”. ARecord=This contain all the aRecord names involved in a website while resolving to the ip address. for the the website ”www.bmw.com”,the

```
; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.bmw.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11656
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.bmw.com.                   IN      A

;; ANSWER SECTION:
www.bmw.com.            3600    IN      CNAME   cn-www.bmw.com.edgesuite.net.
cn-www.bmw.com.edgesuite.net. 3600 IN   CNAME   a1586.b.akamai.net.
a1586.b.akamai.net.     20      IN      A       104.121.76.73
a1586.b.akamai.net.     20      IN      A       104.121.76.64

;; Query time: 22 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Wed Oct 05 21:32:17 CEST 2016
;; MSG SIZE  rcvd: 143
```

Figure 10: dig for bmw.com

ARecord name are a1586.b.akamai.net. and a1586.b.akamai.net. which will be stored in trace will under ARecord column.

8. destIP :this column stores the corresponding resolved ip addresses of the website.For example,for the website bmw.com (figure-6),the destIP will store 72.247.184.130 and 72.247.184.137

9. ASN_Number=Field():This column stores the ASN number of a IP address.For this we use maxmind IP to ASN mapping file.

This procedure resulted in a total number of 13919464 unique links, which includes 100,000 top website links of Alexa, along with their embedded links.

5.2 Data Cleanup

From the traces that are collected in previous section, there might exist some traces for which one or more of the fields are missing, making them invalid. Such traces should be excluded from further analysis. This process of data cleaning are carried out by using regular expressions to filter out the invalid entries.

5.3 Hosting infrastructure to BGP mapping

As explained in section 3.1.1, the tests are carried to find out hosting infrastructure to corresponding BGP prefixes set. The IP addresses and corresponding BGP prefixes collected through RIPE RIS for hosting infrastructure, facebook.com is shown in table below.

SLD to BGP prefix mapping		
SLD Infrastructure	IP address	BGP Prefixes
facebook.com	'173.252.88.112', '173.252.88.113', '173.252.88.104', '173.252.88.66', '173.252.90.132', '173.252.88.73'	'173.252.88.0/21', '173.252.64.0/19'

From the above table, it can be observed that, the hosting infrastructure facebook.com in column-1 might have hosted several web links that are resolved to the above stated 8 IP addresses as stated in column-2 which can be reached through the set of prefixes as in column-3.

5.4 Clustering

There are some hosting infrastructure which are administered by same company. For example, both the hosting infrastructure akamaiedge.net and akamai.net are administered by the same company,Akamai. In such cases i tis verified whether these hosting infrastructure share common infrastructure, i which case are clustered together, following the clustering algorithm as described in section 3.2. This procedure is carried out for all the 219604 unique hosting infrastructure, which has resulted in 53852 unique clustered hosting infrastructure.

To find out hyper giants presence, the features of each clustered hosting infrastructures are analyzed. To find out the feature set of each clustered hosting infrastructure, the corresponding feature set of all the hosting elements involved is aggregate together as described in section 3.2/

The below table lists top 20 clustered hosting infrastructure on decreasing order of their number of links they host.

Clustered SLD infrastructures						
sl. no.	Clustered SLD Infrastructure	SLDs	links	IP addresses	ASNs	prefixes
1	cloudflare.net	17711	1295505	29893	17	78
2	akamaiedge.net	158	597533	3396	9	27
3	google.com	251	240910	195	1	22
4	yunjiasu-cdn.net	6068	210463	5907	21	77
5	us-east-1.elb.amazonaws.com	4254	199109	10885	31	115
6	wpengine.com	3963	136400	4072	19	115
7	anycast.me	2667	116024	2207	3	13
8	ourwebpic.com	722	113746	772	16	16
9	eu-west-1.elb.amazonaws.com	1946	100265	4476	25	31
10	kxcdn.com	1547	82120	1235	7	7
11	edgecastcdn.net	73	79750	383	2	11
12	jiashule.com	1769	79038	2249	28	100
13	cloudflare.com	3	78907	34	1	5
14	alikunlun.com	964	76647	1155	17	45
15	d5nxst8fruw4z.cloudfront.net	3331	75660	1619	3	7
16	incapdns.net	35	66882	382	7	27
17	fastlylb.net	73	65360	201	0	4
18	dynect.net	1	62760	46	2	5
19	ap-northeast-1.elb.amazonaws.com	1308	60343	3105	22	23
20	d2t8dj4tr3q9od.cloudfront.net	2363	59807	1316	8	7

The columns 1-6 from table represent name of the clustered hosting infrastructure, number of hosting infrastructure that are merged into the same cluster, number of links collectively hosted by the cluster, collective set of IP addresses of the cluster, collective number of ASNs of the cluster and collective BGP prefixes of the cluster respectively. From table, it can be observed that 17711 number of SLDs merged with cloudflare.net. From the table it can be observed that two different clustered hosting infrastructure in sl. no. 1, 13, belonging to the same company, cloudflare but remained in different clusters.This is because, they have different infrastructure from each other in terms of BGP prefixes. Same is the case for hosting infrastructure in sl. no. x,y,z administered by Amazon. This might be because of big hosting infrastructure use different infrastructure for various operations.

5.5 Candidate Analysis for hyper giant

From figure it can be seen that, almost 80% of the total unclustered hosting infrastructure got clustered into first 3.73% of hosting infrastructure. Many of the other hosting infrastructure share their infrastructures with these top 3.73% of clustered infrastructures. Hence there is a possibility to get the hyper giants in these 3.73% of clustered hosting infrastructure. But there are hosting infrastructure who have their own infrastructure in the form of data centers. Hence they don't share any other hosting infrastructure. Like facebook.com who has its own infrastructure in the form of data centers all over the world. In fact from figure ,it is also observed that almost 82.45% of clustered infrastructure who does not share their infrastructure with no more than another hosting infrastructure. It means there are companies who work independently by creating their own infrastructure.This can be data centers all over world. Although these 82.45% of clustered infrastructure do not share their infrastructure with no more than single infrastructure, still some of them serve a large number of links which make them another candidate for hyper giant analysis.

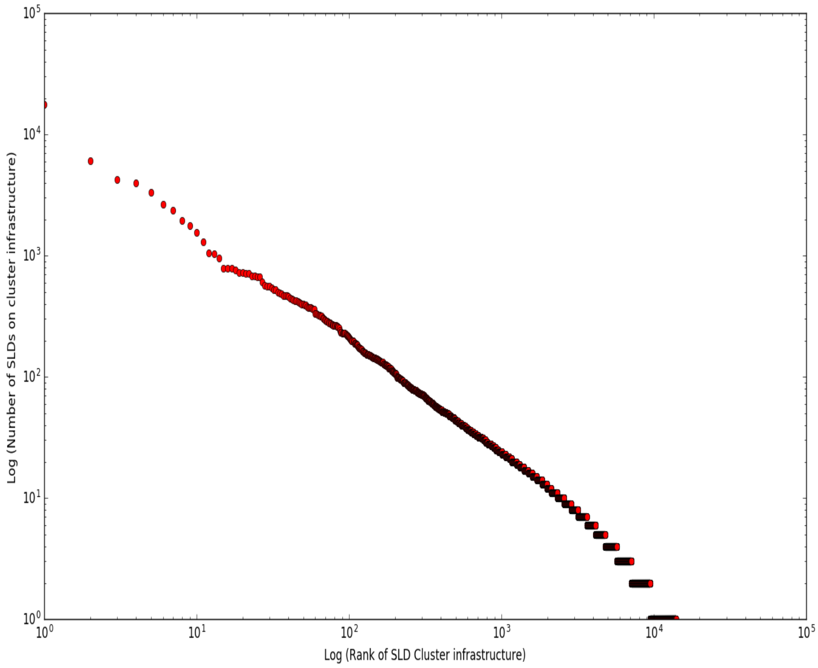

Figure 11: Number of SLDs served by different SLD infrastructure clusters.

Hence to get a better picture, the clustered hosting infrastructure will be analyzed based on how many links they served. Hence by sorting all clustered hosting infrastructure in their decreasing order of links, it is found that top 5.95 % (=3205) clustered hosting infrastructure serve almost 80% of links and have 78.65 % of hosting infrastructure. Hence hyper giants can be found in this range.

5.6 Collection of web objects

Different web objects embedded in home pages of 100,000 top ranked web sites of Alexa are also collected.After crawling, it is observed that total 285 different types of objects from 13919464 links present in Internet. 38% of all the URLs constitute of object type text, while 42 % of URLs constitute of object type image.

5.7 Conclusion

This chapter outline the measurements obtained after crawling the 100,000 top ranked websites of Alexa. This has resulted in total of 13919464 links which include both the home pages of these top ranked websites and their corresponding embedded links. After applying DNS resolution on these links, 219604 hosting infrastructure are detected. On clustering these infrastructure, it has resulted in total 53852 clustered hosting infrastructure. It is found that the top 5.95% clustered infrastructure alone are serving 80% of all URLs. It is also found out that 78.65 % of hosting infrastructure pools into the same 5.95% clustered infrastructure. It is for this reason, it can be conveniently assumed that the hyper giants are exists in these top 5.95 % of clustered hosting infrastructure, owing to zipf's law. The measurements revealed that all the URLs constitute 38% of object type text/html, while 42 % of URLs are of the object type image.

6 Results

This chapter analyzes the measurements obtained from chapter 5 to identify hyper giants by studying the features of clustered hosting infrastructure such as number of links served by the cluster, number of IP addresses that belong to the cluster, number of BGP prefixes to reach the cluster, number of AS numbers that belong to cluster. Once hyper giants are determined, the dependency of popular websites on them will be examined by taking into consideration that the type of web objects like images, videos, HTML files etc delivered by these hyper giants.

6.0.1 Identifying hyper giants

6.0.2 case 1 :number of Links Vs number of IP addresses

In this section we will take the top 3205 SLD infrastructures and cluster them based on their number of links to ip addresses they served.

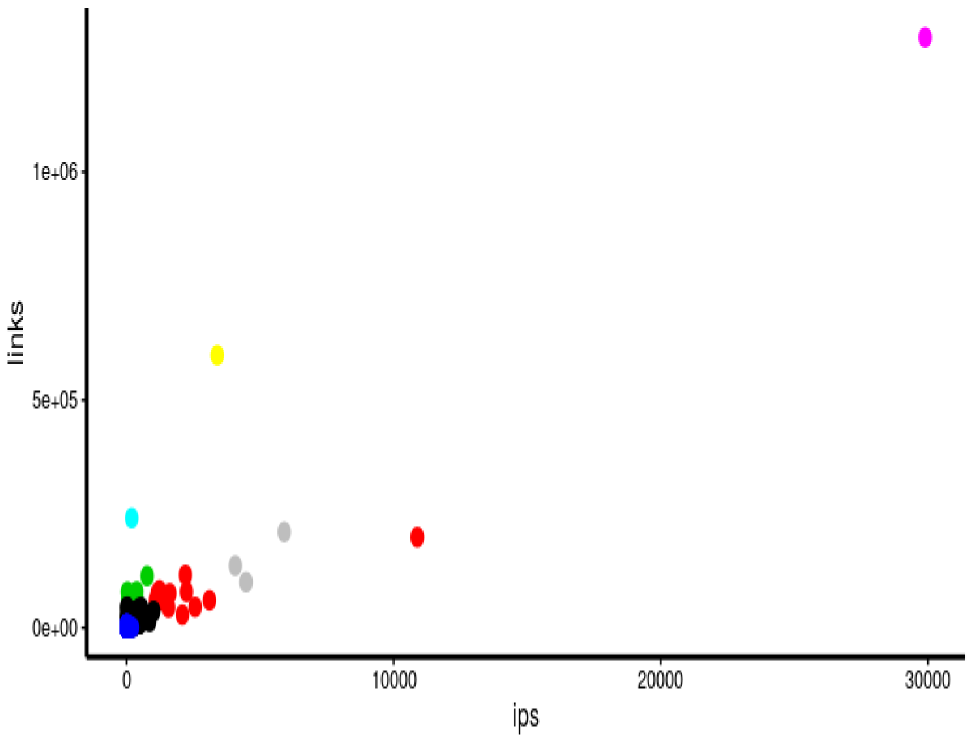

Figure 12: Clustering based on links and IP address features

We used k-means clustering algorithm and number of cluster parameter 10.We found 6 different clusters which are clubbed total 26 SLD infrastructures and showing unique behavior.Like cloudflare.net is clustered separately as it is serving very huge number of links as well as having very high number of ip addresses.It means lots of small SLDs are serving through cloudflare.net and it has footprint all over the world.Similarly us-east-1.elb.amazonaws.com clustered separately as it has less high of links but serving a high number of ip addresses.third cluster contain google.com which is serving high number of links but not very high number of ip addresses.In this way we are able to identify total 6 clusters which resulted in a total of 26 SLD infrastructures. But it is difficult to categorize them into some specific type of infrastructure based on only links to ip address analysis.

Hence these 26 SLD infrastructures will be further analyses taking into account their prefixes to their asn numbers.

6.0.3 case 2 :number of BGP Prefixes Vs number of ASNs

From last section we identify the SLD infrastructures which are having unique behavior. But we couldn't able to classify them .Therefore in this step we will check how they are distributed all other world.This requires these clusters to be analyzed using their corresponding ASN to prefix numbers.This is because number of ip prefixes shows the footprint of the infrastructures across the world and the number of asn numbers show the degree of distribution of infrastructures across the world.

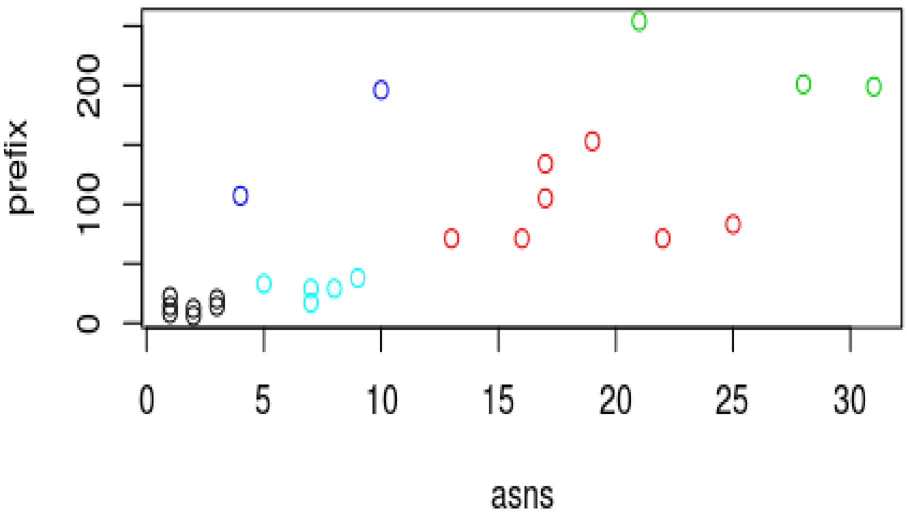

Figure 13: Classification of hyper giants

From figure-14 ,we can cluster all the clustered SLD infrastructures into 5 parts based on their ¡number of prefixes,number of ASNs¿ analysis as below.

- very high,very high : In total 3 different clustered SLD infrastructures are clustered under this. They are yunjiasu-cdn.net,jiashule.com, us-east-1.elb.amazonaws.com.These three SLD infrastructures contain very high number of prefixes as well as they have high number of ASN numbers,which shows that they have footprint all over the world as well as they are distributed across the world.We can classify them as highly distributed CDNs.

- high,high : Total 7 SLD infrastructures clubbed inside this. wpengine.com ,alikunlun.com,cloudflare.net,ourwebpic.com, us-west-2.elb.amazonaws.com,eu-west-1.elb.amazonaws.com and ap-northeast-1.elb.amazonaws.com.They have high number of footprint and high number of asn numbers.This shows they have presence in few of the regions.We can classify them as distributed CDNs.

- high,less : netdns-cdn.com and cdntip.com These SLD infrastructures have high number of prefixes but have less number of ASN numbers.It means they have footprint all over the region but they normally administered through very few ASN numbers.Hence these can be categorized into cloud computing infrastructures.

- medium,medium : Five different SLD infrastructures are clubbed into same cluster.They are, akamaiedge.net,ap-southeast-1.elb.amazonaws.com, kxcdn.com,incapdns.net,d2t8dj All these infrastructures have few prefixes and they also not distributed which gives an evidence of multi homes data centers or web hosting companies.

- less,less : The rest of the CDN infrastructures are clubbed into same cluster which having very few number of prefixes also very few number of IP addresses.Hence they can be treated as content providers.Google and Microsoft both clustered under this.

6.0.4 Conclusion

From the above section,we are able to identify total 26 hyper giants which are having influence in Europe region.They are highly distributed CDNs,cloud platforms,CDNs,content producers etc.In next section we will see how they influence on number of objects delivered by them.

6.1 Popular websites dependency on hyper giants

In this section,first we will see different types of objects delivered through 219604 unique SLDs and compare this with web objects delivered by 26 hyper giants. Then we will examine each hyper giant separately and try to find what kind of data object they deliver.

6.1.1 Object types delivery through whole SLD infrastructures Vs hyper giants

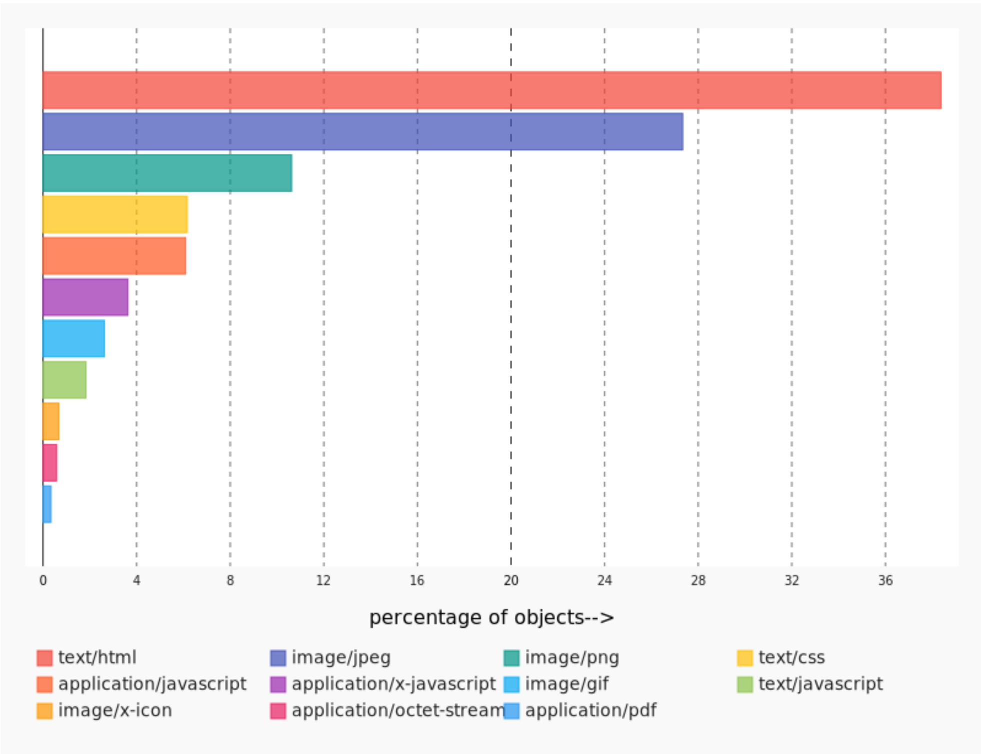

Figure 14: top 10 object percentage used by all SLDs

Figure-15 and figure-16 shows top 10 object types in form of percentage delivered by both hype giants and slds.In case of hyper giants top 10 object types deliver almost 98.82 % of all object types.Again we can see that most of these object types are HTML files.HTML carries almost 68.27% of object type in compare to other object types.Similarly 75.04% of text object types which contain text/html,text/css,text/xml,text/js etc., are delivered through hyper giants where as only 19.29% of image files delivered through hyper giants.

In case of SLDs,top 10 object types carries almost 97.94% of all object types which is very much similar to the percentage of objects delivered by hyper giants.The highest web object type delivered through all SLDs as well as hyper giants is text/html but it can be observed that in case of all SLDs,text/html carries almost 38% of all web objects which is around 68.27% in case of delivering through hyper giants.In case of all SLDs ,object types

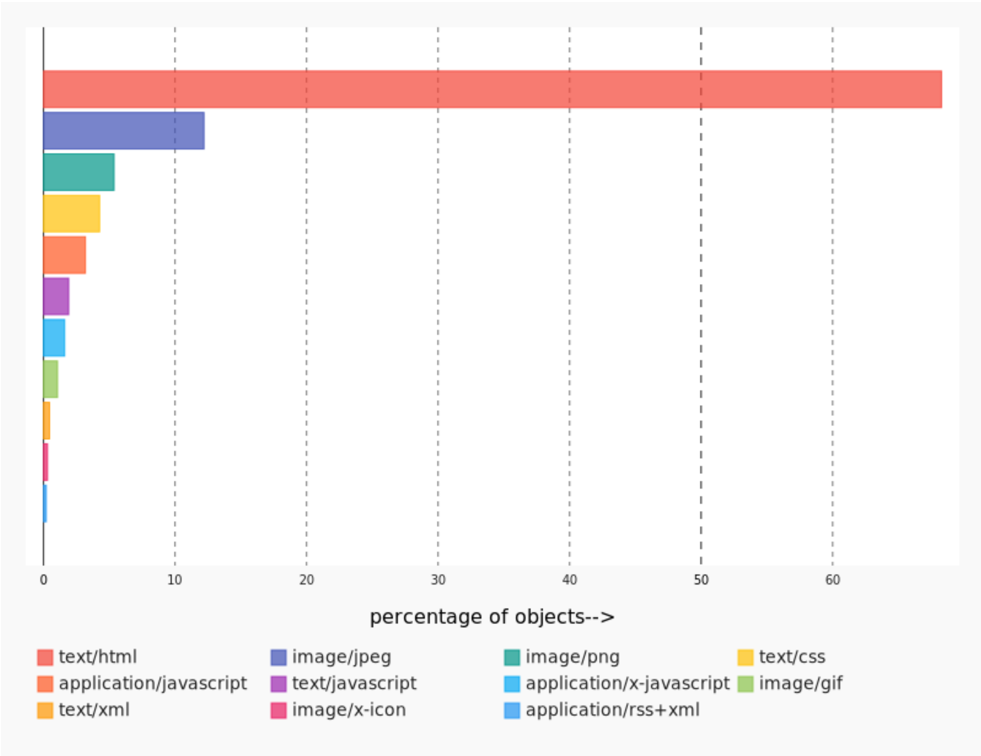

Figure 15: top 10 object percentage used by all hyper giants

are distributed properly.If we add image/jpeg,image/png,image/gif then all SLDs serve more image files than HTML files.But same is not the case for hyper giants.Similarly it can be seen that almost 46.77% of text files delivered through whole sld set which is almost 1.5 times more in case of hyper giants.But in case of image files ,SLDs serve almost 42 % of all web objects which is almost double the image files served through hyper giants.This different behavior might be because popular content websites normally store more dynamic files in CDNs where as in case of image files they store at their own servers.

6.1.2 Object types delivered from hyper giants to popular web sites

In this section we will see what kind of data mostly delivered through the identified hyper giants.In today's Internet ,content plays most vital role.Hence it is important to observe what kind of data mostly delivered through hyper giants .

Hyper giant Object List			
Country Name	text	image	application
alikunlun.com	75.99	20.32	3.65
d2t8dj4tr3q9od.cloudfront.net	49.86	41.10	8.68
ap-northeast-1.elb.amazonaws.com	90.28	6.23	3.44
ap-southeast-1.elb.amazonaws.com	87.94	7.36	4.62
cdntip.com	87.94	7.36	1.67
d5nxst8fruw4z.cloudfront.net	31.90	57.42	10.23
eu-west-1.elb.amazonaws.com	87.25	7.32	5.39
fastlylb	69.79	21.30	8.69
google.com	82.54	15.79	1.65
jiashule.com	88.85	8.33	2.80
kxcdn.com	75.62	17.98	6.34
pbwstatic.com	83.18	16.57	0.24
akamaiedge.net	73.65	20.35	5.93
anycast.me	79.64	14.90	5.37
cloudflare.net	64.86	11.98	23.12
cloudflare.com	78.15	15.62	1.70
dynect.net	94.53	3.74	3.74
edgecastcdn.net	53.23	38.44	8.17
incapdns.net	64.86	7.61	4.81
netdna-cdn.com	21.08	55.04	23.39
ourwebpic.com	86.56	11.58	1.84
us-east-1.elb.amazonaws.com	88.99	6.18	4.73
wpengine.com	80.67	12.42	6.84
us-west-2.elb.amazonaws.com	85.30	8.15	6.47
windows.net	80.52	12.99	6.41
yunjiasu-cdn.net	87.76	9.86	0.0

The table shows all 26 hyper giants and the percentage of text,image and application web objects delivered by them.We can see from table than most of the hyper giants deliver very high percentage of text files which contain text/html,text/css etc.But their are few exceptions .Like both the cloudfront SLDs are providing very high number of images compare to other hyper giants. Similar kind of observation can be seen for netdna-cdn which provides more image files than text files.

From last section we observed yunjiasu-cdn.net,jiashule.com, us-east-1.elb.amazonaws.com are highly distributed CDNs.It can be observed that all the three CDNs are delivering very high number of text files compare to image and application files.This might be because of their footprint all over the world and also have massively distributed CDNs.Hence they cache more of the HTML files at edge servers to provide better performance.Similarly observation can be seen from distributed CDNs .These CDNs also deliver high number of HTML files compare to image files but as they have presence in some regions the number of HTML files are not that comparable to highly distributed CDNs.Third infrastructure type we observed was cloud computing infrastructures and netdns-cdn.com,cdntip.com clustered under that.From the table it can seen that netdns-cdn.com delivers more images and very less number of HTML files.Again cdntip delivers highest number of application data.The data centers provide both images and HTML files in a very balance way.cloud front delivers 49% of HTML file and 41% of image files.Similarly akamaiedge.net provide around 20% of images.Content providers like window.net and google.com delivers very high number of links compare to number of images.This is evident as they have more content .

6.1.3 Conclusion

From the section we can infer that both SLDs and hyper giants deliver maximum number of text/HTML files but it also found that hyper giants delivered almost 1.5 times more HTML files than SLDs.Same kind of observation can be seen for image files where SLDs deliver double the image content than hyper giants.Again we found that highly distributed CDN generally deliver more HTML links compare to image or application files which might be because of their massive CDN distribution.Distributed CDNs provide high number of HTML files because of their presence in few regions.Cloud computing cdns provide more images than HTML file which might be because cloud computing provide scalability.Data centers delivers both HTML file and images in almost same ratio.Content providers also delivers more HTML files compare to image files which because of their rich content.

7 Conclusion

In this thesis,we introduce a automated process to find out the prominent infrastructures in today's Internet as well as to classify these prominent infrastructures to find out the presence of hyper-giants using DNS measurement and bgp prefixes.We presented a clustering algorithm which will help to find out which SLDs are sharing their infrastructures.The advantage of this automated approach is that it uses each to retrieve SLD and bgp prefixes ,hence this procedure can be used in future.

Along with this we measure what object types are delivered by major hyper giants.This will help researchers to get a better view to classify hyper giants based on their object type delivery.Not all popular websites provide same kind of content.Some websites are popular for delivering videos and some other are for user content.Hence with this change of content type,we provide a overview of hyper giants according to different object type they serve.

The data is collected at a single vantage point at Germany.This thesis was able to identify high distributed CDNs,cloud service providers,content providers etc. and their role which will mostly hold good for across Europe.

Furthermore our thesis is an important step towards answering some of the very crucial questions for highly distributed CDNs,distributed CDNs,content providers etc.It will give them idea to find out how other CDNs distribute their infrastructure as well as their network footprint distribute across different region which will give them a competitive advantage over their competitors in content delivery market place.Moreover it will help the research community to discover the Internet architecture changes with time.They also can able to track the hyper giants and their dependency with other popular websites.

8 Future Work

There are certain areas which can be investigated further in future which are not covered in the current scope of this thesis.This section will discuss about these points.

- First of all the thesis is done at single vantage point at Germany.Hence the identification of hyper giants,their role and their relationship with popular websites can be changed when the whole procedure will be done in whole world basis.Hence it will be interesting to see how the clustering algorithm works when taking all the SLDs across the world.
- Secondly while clustering the hyper giants,the k-means parameter is taken by going through very small number of observations.Hence in future this can be tested more precisely which will help to cluster the hosting infrastructures at a granular level.
- In the clustering algorithm,we club two SLDs if one SLD prefixes matches with other SLD prefixes by more than equal to 70%.This matching index is chosen after extensively testing.Hence this can be taken as future work to see what is the best matching index.
- After clustering algorithm,we have chosen to take top 5.95 % (=3205) clustered SLD infrastructures which serve almost 80% of links and have 78.65 % of SLDs .In this way we argumented to get most of the big hosting infrastructures as well as content providers for further analysis.Hence this can be taken for future work to validate the argument properly.

9 Appendix

10 Cluster SLD infrastructure

Three different Clustered SLD infrastructure and corresponding SLD infrastructures are shown in table,

Clustering	
Clustered SLD Infrastructure	SLD infrastructures
'twitter.com'	'digits.com', 't.co'
'google.com'	'googledomains.com', 'chromium.org', 'brokeroficial.com', 'google.ps', 'google.pt', 'teraxhif.com', 'google.hn', 'google.com.do', 'google.al', 'crossmediapanel.com', 'spellup.withgoogle.com', 'google.co.zw', 'shpargalkablog.ru', '1e100.net', 'google.hr', 'google.com.bz', 'google.pl', 'google.hu', 'google.co.ma', 'mrdoob.com', 'aminhaalegrecasinha.com', 'gosarkarinaukri.co.in', 'google.com.ly', 'google.com.bd', 'richmediagallery.com', 'lovefortechnology.net', 'golang.org', 'zagat.com', 'google.co.mz', 'google.com.lb', 'google.com.ag', 'google.com.gh', 'google.com.gi', 'rootupdate.com', 'google.am', 'google.com.cu', 'google.co.za', 'google.bs', 'google.co.zm', 'google.is', 'google.co.bw', 'fashionablygeek.com', 'google.co.ke', 'google.it', '2ality.com', 'osvelhotesdosmarretas.com', 'google.iq', 'google.jo', 'google.ae', 'google.ad', 'gametop.com', 'soberlook.com', 'gundam-vs.jp', 'google.ie', 'google.bj', 'google.co.jp', 'atomlabor.de', 'animesan1.com', 'google.com.vc', 'google.at', 'google.as', 'joancoscubiela.cat', 'google.no', 'google.nl', 'miroivanov.com', 'events.withgoogle.com', 'thisisyouramigaspeaking.com', 'google.ng', 'nanox-cp.jp', 'mediacrooks.com', 'schema.org', 'echridz.com', 'google.com.jm', 'google.co.tz', 'doubleclick.net', 'google.com.br', 'google.com.uy ,......
'awsdns-40.com'	'awsdns-16.com', 'srv1-trovit.ma', 'awsdns-33.com', 'awsdns-49.com', 'awsdns-22.com', 'awsdns-50.com', 'awsdns-21.com', 'awsdns-36.com', 'awsdns-52.com', 'awsdns-29.com', 'awsdns-06.com', 'awsdns-23.com', 'awsdns-12.com', 'awsdns-39.com', 'awsdns-31.com', 'awsdns-47.com', 'awsdns-43.com', 'awsdns-15.com', 'awsdns-02.com', 'awsdns-41.com', 'awsdns-00.com', 'awsdns-51.com'

11 List of Acronyms

CDN	Content delivery network
SLD	Second level domain
ASN	Autonomous system number
BGP	Border gateway protocol
HTTP	Hypertext transfer protocol
HTML	Hypertext markup language
DNS	Domain name system
IP	Internet protocol
QoS	Quality of service
XML	Extensible markup language
ISP	Internet service provider
CDI	Content delivery infrastructure
IaaS	Infrastructure as a service
PaaS	Platform as a service
SaaS	Software as a service
IXP	Internet exchange point
GGC	Google global cache
CNAME	Canonical name
RR	Resource record
URIs	Uniform resource identifiers
URL	Universal resource locator
IDE	Integrated development environment

References

[1] T. Leighton *Improving Performance on the Internet.* Commun. ACM, 52(2):4451, 2009.

[2] C. Labovitz, S. Lekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian. *Internet Inter-Domain Traffic.* In Proc. ACM SIGCOMM, 2010.

[3] I. Poese, B. Frank, B. Ager, G. Smaragdakis, and A. Feldmann. *Improving Content Delivery using Provider-Aided Distance. Information.* In ACM IMC, 2010.

[4] G. Maier, A. Feldmann, V. Paxson, and M. Allman. *On Dominant Characteristics of Residential Broadband Internet Traffic.* In Proc. ACM IMC,2009.

[5] Nygren, R. K. Sitaraman, and J. Sun. *The Akamai Network:A Platform for High-performance Internet Applications.* Syst. Rev., 44:219, August 2010.

[6] R. Krishnan, H. Madhyastha, S. Srinivasan, S. Jain,A. Krishnamurthy, T. Anderson, and J. Gao. *Moving Beyond End-to-end Path Information to Optimize CDN Performance.*

[7] Internet evolution *https://atos.net/content/dam/global/ascent-whitepapers/ascent-whitepaper-internet-evolution.pdf*

[8] Schonfeld, E. Eric Schmidts *Gang Of Four:Google, Apple,Amazon,And Facebook.* TechCrunch.Retrieved from https://techcrunch.com/2011/05/31/schmidt-gang-four-google-apple-amazon-facebook/

[9] Bernhard Ager,Wolfgang Mhlbauer,Georgios Smaragdakis,Steve Uhlig *Web Content Cartography.*

[10] Bernhard Ager *Impact of Location on Content Delivery.* http://people.ee.ethz.ch/ bager/papers/A-ILCD-11.pdf

[11] Yuval Shavitt, Udi Weinsberg. *Topological Trends of Internet Content Providers.* SIMPLEX 12: Proceedings of the Fourth Annual Workshop on Simplifying Complex Networks for Practitioners (2012): 13-18.

[12] Xenofontas Dimitropoulos, Dmitri Krioukov, Marina Fomenkov, Bradley Huffaker, Young Hyun, kc claffy, George Riley *AS Relationships: Inference and Validation.* SIGCOMM Computer Communications Review 37, no. 1(2007): 31-40.

[13] Manuel Palacin, Miquel Oliver, Jorge Infante, Simon Oechsner and Alex Bikfalvi *The Impact of Content Delivery Networks on the Internet Ecosystem.* Journal of Information Policy, Vol. 3 (2013), pp. 304-330

[14] A. Gerber and R. Doverspike. *Traffic Types and Growth in Backbone Networks.* OFC/NFOEC, 2011.

[15] Mike Axelrod *The Value of Content Distribution Networks and Google Global Cache.*

[16] Benjamin Frank, Ingmar Poese, Georgios Smaragdakis, Anja Feldmann, Bruce M. Maggs, Steve Uhlig, Vinay Aggarwal, Fabian Schneider *Collaboration Opportunities for Content Delivery and Network Infrastructures.* 2013

[17] Netflix Open Connect *https://openconnect.netflix.com/en/*

[18] *http://www.hpl.hp.com/research/idl/papers/ranking/adamicglottometrics.pdf.*

[19] Scrapy *http://doc.scrapy.org.*

[20] *Technology: How and Why We Crawl the Web.* Alexa. Archived from the original on April 2, 2014. Retrieved November 6, 2011.

[21] P. Mockapetris *Domain Name System.* https://www.ietf.org/rfc/rfc1034.txt

[22] K RISTOL , D., AND M ONTULLI , L. *HTTP State Management Mechanism.* RFC 2109.

[23] K RISTOL , D., AND M ONTULLI , L. *HTTP State Management Mechanism.* RFC 2965.

[24] Lada A. Adamic ,Bernardo A. *Huberman Zipfs law and the Internet* . Glottometrics 3, 2002,p 143-150

[25] RIPE NCC *RIPE Routing Information Service.* http://www.ripe.net/ris/.

[26] Alexa *http://www.alexa.com/topsites*