Youtube Content Delivery Network

Huiwen Duan, Heikki Nurminen

***Abstract*—This report presents an analysis on how Youtube content delivery network works. We have established Amazon instances on different locations and performed requests to a list of Youtube links. We have observed how Youtube respond to requests from various end client and how a new content gets populated over the CDN.**

Keywords—content delivery network; cloud; retrieval latency; youtube

I. Introduction

TODAY a large number of service providers such as Youtube, Google and the like are employing large data centers to fulfill the requests from the increasing number of users. The data centers are usually allocated at different geological locations and are connected to one or more internet service providers at the nearby region. When a request is received from user, the system will redirect the user to one of its data centers. The purpose of such content distribution networks is to serve content to end-users with high availability and high performance [1]. By utilizing routing policies, the service provider will dynamics manage the traffic between end users and data centers.

Our assignment topic was 5.1 Experimentation with the Youtube Content Delivery Network (CDN). Youtube as the most popular video service of the Internet, has long relied on CDN. Even before Google bought Youtube in 2006, its videos were delivered by several video data centers [2]. At that time, all the centers were located in US. Nowadays Google has build large data centers in different locations to fulfill the massive video requests from their users. There are at least nineteen data centers in the US, twelve in Europe, one in Russia, one in South America, and three in Asia. [3]

In this assignment we explored how Youtube's data centers serve customers in different areas. In the first phase of our assignment we gathered 20 popular video links from Youtube and then downloaded them from 6 different locations to examine the location of response servers. In the second phase, we uploaded a brand new video to Youtube and monitor how it is spread over CDN and the retrieval latency from different request sources.

II. Experiment setup

In our experimentation we used three Amazon EC2 instances in following locations: Singapore, Ireland and California. Each running Ubuntu 12.04.2 LTS 64 bit micro instance. We chose Amazon EC2 micro instance for this experiment because it has several advantages over the rest of the solutions. Micro instances are a very low-cost instance option, providing a small amount of CPU resources. Micro instances may opportunistically increase CPU capacity in short bursts when additional cycles are available. They are well suited for lower throughput applications. One ECU provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor [4]. Since we are only using the instances to examine how YouTube distribute their videos and how a newly uploaded will be populated in their data centers, Amazon micro instance will well fulfill our needs on performing the non-CPU-intensive tasks. To access those Amazon instances, we also reserved and attached three public IPs respectively to instances at Singapore, Ireland and California.

In Finland, we used our home computers (Windows 7 and Linux mint 15 64 bit) and broadband connections (Sonera) in addition to Aalto University's desktops.

To perform the instructed tasks, we have installed Pytomo, Tshark and Lynx on each instances. Pytomo is Python based tomographic tool to perform analysis of video streaming sites [5]. Pytomo allowed us to crawl a list of Youtube videos and output information of each video, including the responded server IP, average ping time, time to get the first byte and some other data. To our relief, Pytomo works in all of our experimentation setups. In the second task, we chose tshark as our network traffic analyser because it can output a file containing the packets information in the given time interval, allowing us to monitor the traffic without graphic user interface and calculate the time duration between HTTP get request from our instance and HTTP response from Youtube server. Since we have limited CPU memory and computing power, we use text browser Lynx to perform the request to our newly uploaded Youtube video.

III. Measurement

In the first task our measurements took place between 13-28.10. During this time we used our instances, home and university's computers to crawl through our link list. In Linux version of Pytomo, we used the following command:

```
start_crawl.py -r 1 -D
10 -R [list of 20 links separated by
whitespace]
```

Parameter -r 1 specifies that only one round will be executed, -D 10 means that each video is downloaded only for a 10 seconds and finally -R [links] specifies which videos are about to be crawled. On Windows the command is slightly different but behaves the same, and thus, is omitted from this report. All the measured data was stored automatically to /database folder as a .db format. To evaluate server location, we pinged IP addresses from each location and Aalto's network and then made wise guess. We are aware that this far from precise way to estimate location. However, pointing each IP address to a single data center is out of the scope of this paper. More details

about the collected data will be examined in the Outcomes chapter.

In the second task we examined how our newly uploaded video will be populated over the Youtube CDN. First we start tshark by

```
tshark -a duration:60 -w singapore.pcap
```

meaning the duration of traffic capturing is 60 seconds and the captured data will be stored to singapore.pcap file. Then we used lynx to perform the HTTP request to our video link by

```
lynx http://www.youtube.com/watch?v=NWEY_Ckn7v8
```

The retrieval latency is obtained by calculate the duration between the first HTTP GET request sent to the content server and the first HTTP response from the content server. On windows, we used Wireshark, which is basically tshark with GUI. Retrieval latency is calculated similarly as with tshark.

IV. Outcomes

In this chapter, we have grouped and analysed the data based on responded server IPs, the average ping time to server IP and the retrieval latency. Detailed data can be observed in the following section.

A. Task 1

In total, we received video data from 118 servers. 83 of those IPs were unique in this data set. In figure 1, server location is presented regarding estimated location.

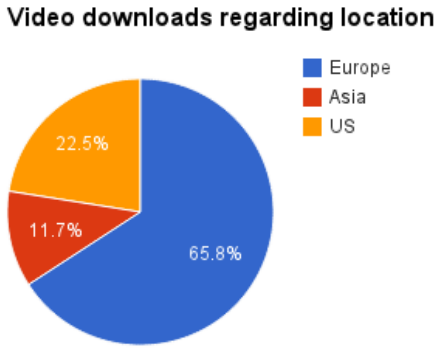

Fig. 1. Video download locations

Europe seems to be dominant in this aspect. This is hardly a surprise while 4 of our test locations are in Europe. In figure 3, video sources are shown in relation to amount of request made in this experiment.

Figure 2 acts as an example of how we collected data from all the test locations. In this case, most of the videos are downloaded from addresses 80.239.229.xxx, while 173.194.48.xxx

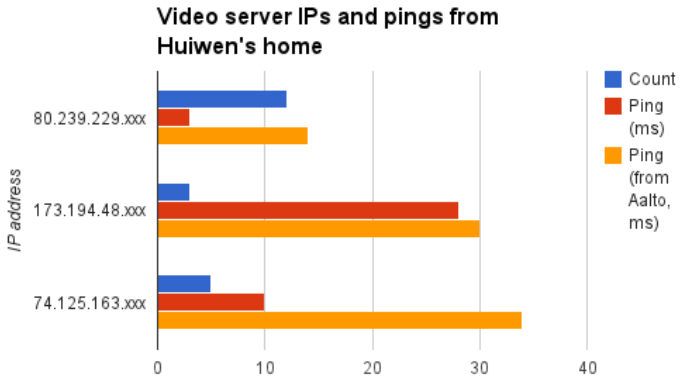

Fig. 2. Video download locations

and 74.125.163.xxx received much less requests. There is a noticeable change in ping times, but on the other hand, maximum ping reaches just above 30ms, which suggest that the data center is located somewhere in Europe.

IP addresses are simplified in order to make results more readable and to reveal which addresses belong to a certain data center. Our instance in Ireland downloads all the videos from Europe. However, from four different data centers. Same idea can be seen in the case of California and Huiwen's home. Still, latency varies among data centers in the same continent. This means that, even though Youtube knows which data center would be the closest one, it decides to utilize some other one. This behaviour is even highlighted when looking data from Singapore. Our instance in Singapore uses 4 data centers from Asia and 3 from US. As following table explains, RTT grows drastically when downloading from US.

IP Address	Ping (ms)	Ping (from Aalto, ms)
US 74.125.212.xxx	241	150
Asia 173.194.49.xxx	4	341
Asia 74.125.96.xxx	14	348
US 74.125.215.xxx	171	180
Asia 74.125.10.xxx	46	320
US 74.125.213.xxx	187	193
Asia 173.194.59.xxx	75	279

Our results so far confirm what assignment instructions and [2] suggests. Youtube does not heavily prioritize geographical location when deciding which data center should serve users from various locations. In the light of our data, this seems pretty coherent solution because only Singapore's instance is suffering from requesting far from initial location.

The only really peculiar result is from Aalto University's network, as 109.105.109.xxx addresses seem to have no relation with Google or Youtube. According to several sources those addresses are in the possession of NORDUnet, which is an organization maintaining research and education networks in Nordic countries. FUnet, on the other hand is NORDUnet's suborganization that handles connections within Finland. Fi-

nally, it appears that Google and NORDUnet have private peering agreement [6] which explains why our test shows mainly (19/20) NORDUnet's addresses.

B. Task 2

In this task, the content server IPs have been captured and the retrieval latency of our newly uploaded video has been calculated to observe how a video will be populated over the Youtube CDN.

Client Location	Content server IP
Singapore	74.125.235.2
Ireland	74.125.24.93
California US	74.125.239.128
Heikki's home	80.239.229.219
Huiwen's home	80.239.229.248
Aalto	173.184.32.25

Table above shows the content server IP for responding requests from Amazon instances and Aalto University are all from United States, while responded content server for testers home are from Europe. It is interesting to note that in the case of Singapore and Ireland, Youtube did not direct the client to its nearby data center but to the servers that are considered geographically further. Youtube might employ their own route policies to optimize the load balances among its data centers. However, the content server IP might not be so accurate since Google may choose to hide their servers IP due to security reasons.

Figure 4 demonstrates the retrieval latency from different locations. It can be seen that the respondent time to requests sent from home and Aalto are much shorter than they are from the Amazon EC2 instances. There are several reasons that might explain this situation. Firstly, the network bandwidth that each end client possesses are different. As stated in Amazon documentation, the network performance of micro instance is comparatively low, which as a result lead to significantly higher retrieval latencies. While both testers use Sonera ISP with slightly different connection speeds, results show almost identical retrieval time. And in the case of Aalto, it might be for security concerns that the request need to pass several more routers to reach the destination content servers and the response need to pass firewalls to reach the client.

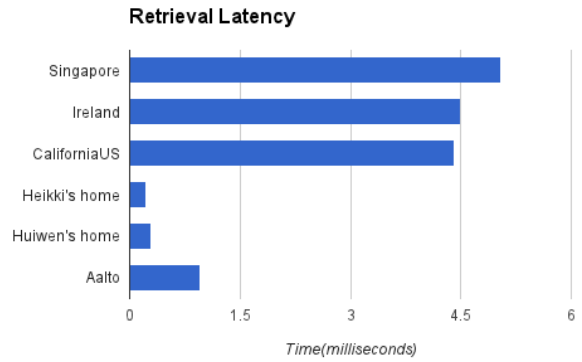

Fig. 4. Retrieval latency

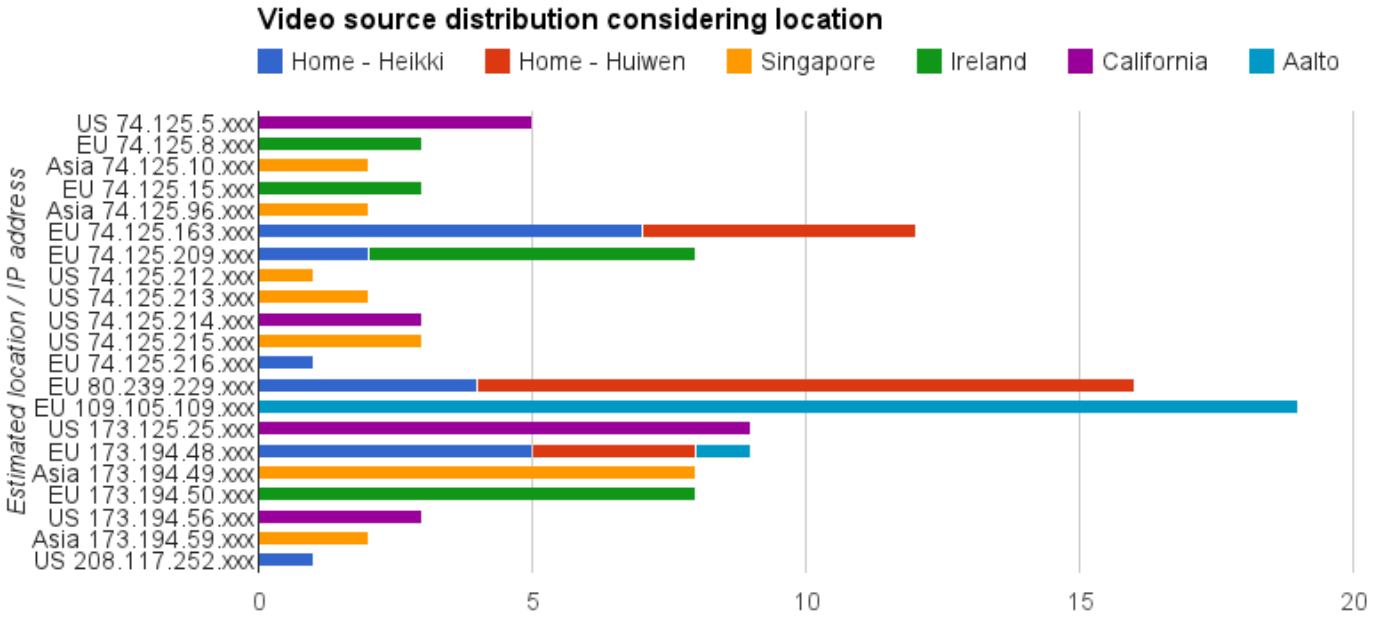

Fig. 3. Content server IP location

V. Conclusion

In this report, motivated by the popularity of distributed system over the network, we have studied how Youtube content delivery network work. To provide service to its ever growing users all over the world, Youtube has deployed many data centers in various locations.

Before we started the tasks, our hypothesis was that Youtube will redirect the client's video streaming request to the data center that is geologically nearby the client, however the results were slightly deviated from what we expected. In the Outcomes chapter, we have presented and analyzed our data. It clearly shows that Youtube employs much more complicated algorithm to solve how users request are routed from the front-end server than just determining which one is the closest. As [2] presented, Youtube might still distribute request based on data center size and capacity. If this is true, we can draw a conclusion that European and US data centers are quite close to equal size because very few requests route from EU to US and vica versa. On the other hand, Asian centers seems to be slightly smaller than in the US, because request flow from Singapore to US.

References

[1] Wikipedia, Content delivery networks, October 2013, Accessed last time 25.10. 2013. http://en.wikipedia.org/wiki/Content_delivery_network.

[2] V. K. Adhikari and S. Jain and Z. Zhang, *Youtube traffic dynamics and its interplay with a Tier-1 ISP: An ISP perspective*, In Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, IMC'10, pages 431-443, New York, NY, USA, 2010.

[3] Maps of Google data centers, Pingdom.com, April 2008, Accessed last time 25.10.2013. http://royal.pingdom.com/2008/04/11/map-of-all-google-data-center-locations/.

[4] Amazon instances type, Amazon.com, Accessed last time 25.10.2013. http://aws.amazon.com/ec2/instance-types/.

[5] Pytomo, Accessed last time 25.10.2013. https://code.google.com/p/pytomo/.

[6] NORDUnet traffic with Google - Google private peering, Accessed last time 25.10.2013. http://stats.nordu.net/stat-q/r-all?q=all&name=Google.