Where Do You "Tube"? Uncovering YouTube Server Selection Strategy

Vijay Kumar Adhikari, Sourabh Jain, Zhi-Li Zhang
Computer Science & Engineering
University of Minnesota - Twin Cities
Minneapolis, Minnesota 55455
Email: {viadhi,sourj,zhzhang}@cs.umn.edu

***Abstract*—YouTube is one of the most popular video sharing websites in the world. In order to serve its globally distributed users, it requires a massive-scale video delivery system. A major part of the whole system is to decide exactly what server machine is going to serve a client request at any given time.**

In this paper, we analyze DNS resolutions and video playback traces collected by playing half a million YouTube videos from geographically distributed PlanetLab nodes to uncover load-balancing and server selection strategies used by YouTube. Our results indicate that YouTube is aggressively deploying cache servers of widely varying sizes at many different locations around the world with several of them located inside other ISPs to reduce cost and improve the end-user performance. We also find that YouTube tries to use local "per-cache" load-sharing before resorting to redirecting a user to bigger/central cache locations.

I. Introduction

YouTube, which started as a "garage-project" to share videos online in 2005, has seen an explosive growth in its popularity. Today it is indisputably the world's largest video sharing site. It serves millions of users across the world every day. However, due to its ever increasing popularity and demand, it is subjected to a continual expansion to accommodate the growing demands. As shown in a recent study [2], in 2008, it used 6 large data-centers located with in United States to serve the videos to users (while LimeLight CDN was used to push the most popular videos to users). However, these data-centers were not enough to meet the increasing global demand, and sometimes after it was bought by Google, it started expanding its video distribution network by using Google's infrastructure.

In a recent work [1], Adhikari et.al. used a reverse engineering based methodology to uncover the basic principles behind the design of YouTube video delivery cloud. It showed that YouTube video delivery cloud consists of three components: the video-id space, hierarchical logical video server DNS namespaces and physical video cache servers. YouTube then uses a combination of static and dynamic load balancing approaches to distribute the demand to its physical resources. As shown in that work, the video-id space consists of fixed length unique identifier referring to each YouTube video, which are then mapped to hierarchical video server namespaces using static hashing. In addition, YouTube DNS servers map these server hostnames to IP addresses corresponding to physical hosts in a client-location aware manner.

This work is supported in part by the NSF grants CNS-0905037 and CNS-1017647, and the DTRA Grant HDTRA1-09-1-0050.

In this paper, we use the same active measurement infrastructure used in [1] to provide insights into the server selection strategy employed by YouTube video delivery network. In particular, (i) we extract and chart the physical resources currently used by YouTube to serve videos, (ii) provide details on how various strategies are used by YouTube to distribute the video request to its geographically distributed global cache servers, and (iii) how these strategies interact with each other.

Our study shows that YouTube uses 3 different approaches to distribute the load among various servers.

a. Static load sharing using hash based mechanism: As noted in [1], YouTube maps each video-id to a unique hostname in each of the namespace in the hierarchical DNS based host namespaces. This provides a very coarse grain load-sharing that allocates equal number of videos to each of the hostnames in the primary namespace.

b. Semi-dynamic approach using location aware DNS resolutions: YouTube maps each DNS hostname to an IP address, which represents a physical video cache, based upon the user location and current demand. As seen in our experiments, YouTube redirects user to a geographically close cache location during the "normal" hours. However, during the "busy" hours it uses DNS based resolutions to direct the user to a slightly farther location, which helps in avoiding geographical hot-spots. This is one of the new insight we gained by analyzing a large number of continuous DNS resolutions over more than a month.

c. Dynamic load-sharing using HTTP redirections: Finally, to further balance the load on different physical servers YouTube caches used HTTP redirections to direct user from a busy server to a not so busy video server. It helps in smoothing the skewed load distribution caused by the combination of video popularity and the spontaneous video demands. The interesting and new observation we made by analyzing the redirection logs is that YouTube uses local "intra-tier" load-sharing before resorting to "inter-tier" load-sharing. In addition, none of these approaches require any centralized coordination.

Our findings also show that YouTube caches are present at more than to 45 cities in 25 different countries around the world. These trends suggest that Google is aggressively pushing its content close to users by placing a large number of "cache servers" at various geographical locations around the world. Moreover, several of these caches are co-located with ISP-PoPs, which not only helps in reducing the bandwidth cost for both the ISP and YouTube, but also improves the performance for the ISP users.

978-1-4577-0638-7/11/$26.00 ©2011 IEEE

The remainder of the paper is organized as follows. We present background and related work in Section II. Section III describes and characterizes various YouTube cache locations. We discuss YouTube's server selection and load balancing strategies in Section IV and conclude the paper in Section V.

II. Background & Related Work

In this section we first summarize the related work. We then provide an architectural overview of YouTube video delivery cloud based on the findings from a recent study [1] which forms the basis for this paper.

A. Related Work

The most relevant to our work is the recent study carried in [1]. In this study, authors used an active measurement testbed comprising of several Planet-Lab nodes and open recursive DNS servers. Using the testbed authors played a large number of videos and analyzed the detailed video playback logs to distill out the basic design principles behind the YouTube video delivery cloud. However, the work mostly focuses on the organization of video servers and provides very limited analysis of mechanisms used to perform load-balancing and how it impacts the video delivery performance. In this work, we use the same testbed to extract the key load-sharing mechanisms used by YouTube. In addition, we also provide a detailed charting of various YouTube cache servers, their locations and other characteristics. Although we borrowed the active measurement platform, and some of the data collected from [1], this work differs significantly from [1] in terms of its contribution. In particular, this paper tries to understand how YouTube decides which server is going to serve any particular video request and what factors affect this decision process. In another recent study [2], authors utilize the Netflow traffic data *passively* collected at various locations within a tier-1 ISP to uncover the locations of YouTube data centers, and infer the load-balancing strategy employed by YouTube at that time. The focus of the study was on the impact of YouTube load-balancing on the ISP traffic dynamics, from the perspective of the tier-1 ISP. As the data used in the study is from spring 2008, the results reflect the YouTube delivery infrastructure *pre Google re-structuring*. Another piece of relevant work in terms of the active measurement methodology is the study in [6], where the authors compare two design philosophies used in content distribution networks (CDNs) such as Akamai and Limelight, and conduct a comparative performance evaluation (e.g., delay and availability performance) of these CDNs through measurement.

There are several other existing studies of YouTube, which mainly focus on user behaviors or the system performance. For instance, the authors in [3] examined the YouTube video popularity distribution, popularity evolution, and key elements that shape the popularity distribution using data-driven analysis. The authors in [5] investigate the (top 100 most viewed) YouTube video file characteristics and usage patterns such as the number of users, requests, as seen from the perspective of an edge network. Another study[8] analyzed network traces for YouTube traffic at a campus network to understand benefits of alternative content distribution strategies. A more recent work [7] studies the impact of the YouTube video recommendation on the popularity of videos. Our study compliments and advances these works by shedding lights on the multi-step load-balancing and server selection strategy employed by YouTube to serve video contents from geographically diverse cache locations.

B. YouTube Video Delivery Cloud

YouTube video delivery cloud consists of three major components: video id space, hierarchical logical video servers represented using multiple *anycast*[1] (or logical) DNS namespaces, and a 3-tier physical server cache hierarchy. In the following we present a brief overview of these components, while detailed description can be found in [1].

YouTube Video Id Space: Each YouTube video is uniquely identified using a fixed length flat identifier. These identifiers construct the video id space.

Hierarchical Cache Server DNS Namespaces: YouTube defines multiple (*anycast*) DNS namespaces, each representing a collection of *logical* video servers with certain roles. Together, these (anycast) DNS namespaces form a hierarchical organization of *logical* video servers. Logical video servers at each level of this organization are mapped to IP addresses (of "physical" video servers residing at various locations) within a particular tier of the physical cache hierarchy. There are a total of three sets of *anycast* namespaces, which we refer to as primary, secondary and tertiary namespaces; each namespace has a specific format. E.g. hosts in primary namespace use the following format: `v[1-24].lscache[1-8].c.youtube.com`. As seen in the above example there are a total of 24×8 or 192 such hostnames in the primary namespace. Similarly, there are 192 hostnames in the secondary namespace, and 64 hostnames in tertiary namespace. In general only the DNS names belonging to the primary namespace are visible in the URLs or HTML pages referencing videos; whereas DNS names belonging to the other two namespaces occur mostly only in the URLs used in dynamic HTTP request redirections during video playback.

Physical Cache Servers: Although there are three unique logical namespaces used to represent the DNS names for the hosts in various logical cache servers, each anycast DNS hostname may map to a large set of IP addresses, which we refer to as the physical cache servers. E.g., hostname `v1.lscache1.c.youtube.com` in primary namespace maps to 75 different IP addresses depending on time and the location of the user.

Datesets. For this study we used several datasets that were collected using our active measurement platform using 500 PlanetLab nodes and 1000 open recursive DNS servers.

We obtained video playback traces by playing approximately half a million videos used in [1]. We also obtained

[1]Here by an *anycast* (DNS) namespace we mean that each DNS name is *by design* mapped to multiple IP addresses ("physical" video servers).

IP geolocation data from the same study. Additionally, we collected DNS resolution data for the YouTube hostnames from all the PlanetLab nodes continuously for a month. During that period we also monitored the change in view-counts for all 437K videos.

III. Charting YouTube Global Caches

Before we try to answer how YouTube's server selection strategy works, we need to have an understanding of what server resources YouTube is currently using and where those server resources are. Since we do not have a way to precisely know how many or what types of server machines YouTube has at its disposal, we use the IP addresses seen at each of the locations as an approximation for the number of servers at that location. We understand that behind each IP address, there might be multiple physical machines not directly observable from the outside.

In the following, we provide a detailed charting of YouTube physical video servers in terms of the IP addresses used and their geographical locations. For these analyses we extract the YouTube server IP addresses by resolving YouTube video server hostnames from various PlanetLab nodes and open recursive DNS servers continuously for more than a month. We also utilize the IP to city mapping data used in [1] to chart the geographic distribution of YouTube caches.

A. IP addresses for YouTube servers

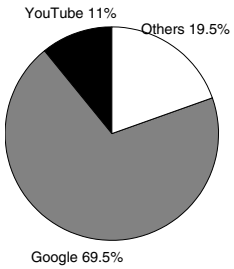

Fig. 1. Organizations where the YouTube video-serving IP prefixes come from.

Using the DNS resolutions of YouTube servers' logical DNS namespaces we extracted a total of 6000 unique IP addresses. In terms of the namespace hierarchy, there are 5000 IP addresses corresponding to primary logical server namespace, and 636 and 320 IP address for secondary and tertiary namespaces respectively.

With the large number of IP addresses in hand, we next try to find what address space those IPs come from. We performed WHOIS queries [4] for all the extracted IP prefixes to find the corresponding owner organizations for them. Our results show that although most of the IP prefixes (/24s) belong either to Google or to YouTube, there are approximately 20% of the prefixes that belong to several other ISPs. The distribution of the organizations that own these prefixes is shown in Figure 1. In this figure the "others" category includes several regional ISPs such as Bell-Canada, Comcast and some Internet Exchange Points (IXPs).

To see whether YouTube caches with IP addresses coming from other ISPs differ from the ones with IP addresses coming from YouTube or Google's address space, we analyzed the video playback log where we had used spoofed DNS response to send a client to any arbitrary YouTube location. The data suggested that the caches using any other ISP's address space only served that ISP's customer cone.

To verify this, we sent video playback request to such YouTube caches from a large number of PlanetLab nodes. We saw that only requests coming from a PlanetLab node that was in the ISPs customer cone could download a video. All other clients PlanetLab nodes received a redirect message instructing them to try some other location. We used traceroutes and BGP AS paths information to learn if a PlanetLab node is inside the customer cone of an ISP or not.

B. YouTube Cache Locations and Sizes

Fig. 2. Three tiers of YouTube video cache locations(P=Primary,S=Secondary,T=tertiary).

Using the IP address to location mapping data we identified a total of 45 cities where YouTube video caches are present. Out of these 45 locations, 43 distinct locations correspond to primary caches, while secondary and tertiary caches mapped to only 8 and 5 distinct locations respectively. The geographical span of these locations is shown in Figure 2. The primary, secondary and tertiary cache locations are indicated by letters P, S and T respectively in that figure. We note that at some locations, more than one tier of the cache hierarchy are present.

Although there are 45 unique locations where YouTube caches are present, not all the locations are the same in terms of the available resources. To understand the diversity in the size of different cache locations, we plot the distribution of YouTube cache sizes in Figure 3. We use the number of IP addresses present at each location as a proxy for the size of that cache location. In this figure the x-axis represents the YouTube locations, and the y-axis represents the size in terms of the number of IP addresses. As seen in this figure, the sizes of the cache locations vary widely. In particular there are locations, which have large number of IP addresses, e.g. Frankfurt cache location has more than 600 IP address, while cache located in Athens has merely 9 IP addresses. Moreover, we also find that in general YouTube caches hosted inside other ISPs or IXPs are much smaller compared than the ones whose IP addresses are coming from YouTube/Google address space.

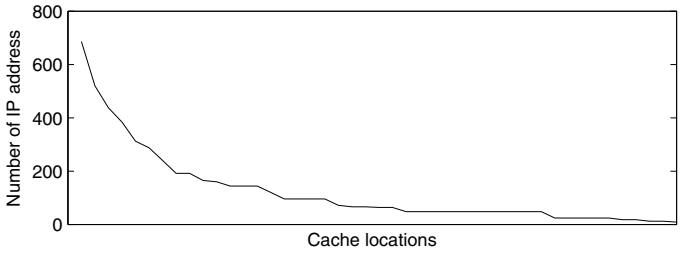

Fig. 3. Distribution of size of YouTube cache locations (cities) in terms of number of IP addresses seen.

In summary, we see that YouTube datacenters are spread all over the world, and they vary widely in their sizes as well as in terms of who they serve. While most of the YouTube locations can serve any client from any geographic location, the locations hosted inside other ISPs only cater to the customer base of those ISPs.

IV. Physical Server Selection & Load Balancing

We saw that YouTube aggressively pushes its content closer to user using a large number of geographically distributed video cache servers. In this section, we use that information to uncover the several mechanisms used by YouTube to serve each user request, so as to maximize user performance and also to evenly distribute load on its cache servers.

When a user wants to watch a video, the video ID gets mapped to a unique hostname. The hostname then maps to different IP addresses based upon multiple factors. YouTube's server selection strategy can therefore be understood by examining this hostname to IP mapping. In this section we investigate the spatial and temporal factors that influence what IP the hostname gets mapped to.

A. Location Awareness in Server Selection

In order to understand how the user's location plays a role in the mapping of YouTube cache server hostname to IP address mapping, we resolved the hostnames corresponding to primary logical servers from all the vantage points. We find that each primary logical server hostname maps to approximately 75 unique IP addresses. We also computed ICMP ping latencies between the vantage points and all the mapped IP addresses. These round trip delay measurements show that the hostname to IP address mapping provided by YouTube DNS servers tries to direct the user to a close physical cache server. We note that the closeness we measured is in terms of network latencies. However, we found that in most of the cases, closeness in terms of network latencies also translated to closeness in geographical distances as expected.

To demonstrate the location-awareness in the DNS resolutions, we plot the percentile for the latency between the vantage point and the mapped IP address, which is shown in Figure 4. In this figure, X-axis represents the percentile corresponding to the latency. This indicates that although the hostnames could have been mapped to any of the 75 IP addresses, in almost all cases, YouTube maps them to a physical cache server (or IP address) that is close to the client requesting a video.

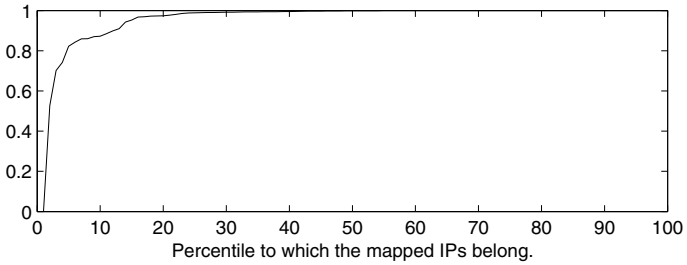

Fig. 4. CDF plot showing which decile mapped IPs belong to in terms of ICMP ping latency.

We also visualize the location-awareness in DNS resolution using a geographic map as shown in Figure 5. In this figure we tag each PlanetLab node's location using one of the 5 YouTube tertiary cache location that it is mapped to. Here, each balloon represents a PlanetLab node and the letter indicates the tertiary cache location that node gets redirected to using DNS based resolutions. We can clearly see that in almost all of the cases, the nodes pick the closest tertiary cache location.

Fig. 5. Locations of PlanetLab nodes indicating their tertiary-cache location choices (B=Brussels, C=Council Bluffs, F=Frankfurt, I=Washington DC, N=Mountain View).

These results show that YouTube tries to send users to one of the closest cache locations although the same hostname can in fact be mapped to a large number of IP addresses in a large number of cache locations.

B. Temporal Patterns in DNS Mappings

To understand temporal changes in DNS resolutions, we continuously resolve the logical server hostnames from all the vantage points for a month-long period.

Our analysis of these DNS resolutions revealed that we can group the vantage points into two distinct groups based on how frequently the mappings change. In the first group of vantage points the mappings change during a certain time of the day, and the pattern repeats every day. However, the pattern is more continuous for the vantage points in the second group.

To illustrate this we plot the IP address mappings for one of the primary logical server's hostname. Figure 6 shows an

example plot for the second group. In this figure, the X-axis represents the time which is divided in the intervals of 5 minutes each, and Y-axis represents the mapped IP address. As seen in this figure the mapped IP address changes almost every time. It suggests that YouTube is trying to use 4 distinct physical servers to represent the instance of one particular primary logical server, and changes the mapped IP address to divide the client request to each of these physical servers. On the other hand, there are other locations where new IP addresses only show up at specific times of the day. Our analysis of DNS resolution pattern for these locations shows that each logical server hostname maps to a fixed IP address most of the time during the day, however, during the certain hours of the day we see a large number of distinct IP addresses for the same hostname. We show an example of such a location in Figure 7. In this figure, we see that DNS servers primarily map the hostname to IP 3. However, at "busy" hours of the day, the DNS server also starts mapping the hostname to other IP addresses. In this week long data, we can clearly see 7 specific periods (one every day) in which we see additional IP address other than IP 3.

C. Need for Dynamic Load Balancing

We have seen that YouTube uses a static hash-based scheme to map videos to hostnames. However, the static mapping of videos to hostname can lead to uneven load on the logical servers due to the different popularity for each video. For instance, if a video becomes popular, the logical server corresponding to that video will experience more traffic than other servers. These uneven loads for logical servers can then translate into uneven load distribution for physical servers corresponding to them. For instance, at each cache location if different logical names map to different physical hosts, the physical host that is responsible for the logical server corresponding to the popular video will experience more load than others.

To understand this uneven load among different servers, we tracked the global video view counts for approximately half a million videos over a month. In an ideal case, we needed the number of times each physical server served a video. However, as this information is not publicly available, we used the global video view count as a proxy to the load on each of the logical servers. We see that even when we look at aggregate video counts over a month that includes videos views from all over the world, different logical servers receive widely differing video requests. To demonstrate this we plot the distribution of video request on each logical server during a month for our sample of half a million videos in Figure 8. As seen in this figure, some logical servers are responsible for approximately 4 times more video requests than others. In fact, we can expect that during smaller time windows and at different location this distribution is likely to be even more skewed, which can result into highly uneven load distribution on different physical servers.

We can see that in this setting, it is very hard to achieve an even load sharing among different physical servers at a given cache location. In addition, these static hash-based load distribution and DNS based dynamic mappings, by themselves, can not ensure that a location will be able to handle a sudden increase in video demand. To deal with these problems, YouTube uses a more dynamic approach using HTTP redirection, where a server responsible for serving a video sends a HTTP 302 header back to the client asking it to get the video from a different server.

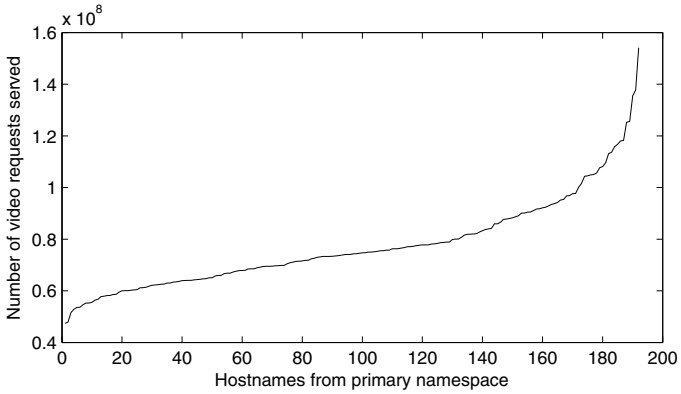

Fig. 8. Distribution of video requests per primary cache hostnames.

To understand this HTTP redirection based dynamic load sharing approach, we extract all the instances where a user is redirected to a different physical server using such redirections. Our analysis of these redirect logs reveals two distinct patterns in redirections. In the first set of instances, a logical server in one namespace redirects to another logical server in the same namespace. We refer to these cases as "intra-tier" redirections. In rest of the cases, we see that a server in one namespace redirects to a server in a higher tier of cache hierarchy (such as a server from primary namespace redirecting to a server in secondary namespace). These cases are referred to as "inter-tier" redirections.

In case of intra-tier redirection, we looked at locations of the servers involved in the redirection. A careful inspection of these redirections showed that the IP addresses of the two servers (the one that redirects the client and the one which receives and handles the redirected requests) are in the same location (city). This means in case of intra-tier redirections, YouTube is trying to redirect a user to another physical server at the same location.

This suggests that YouTube is trying to distribute the load locally: i.e. a busy server in one location redirecting a client to a less busy server in the same cache location. In such cases it is expected that the busier servers should be descernible from less busy ones. To see if this is indeed the case, we look at the individual IP addresses involved in redirections at individual cache locations. We see that at each location and at any given time period, there are clearly distinct set of servers, one that send to redirect to others, and another set that handles redirects but rarely redirect clients to others.

On the other hand, in case of inter-tier redrections, clients are sent to higher tiers of the cache hierarchy. As there are

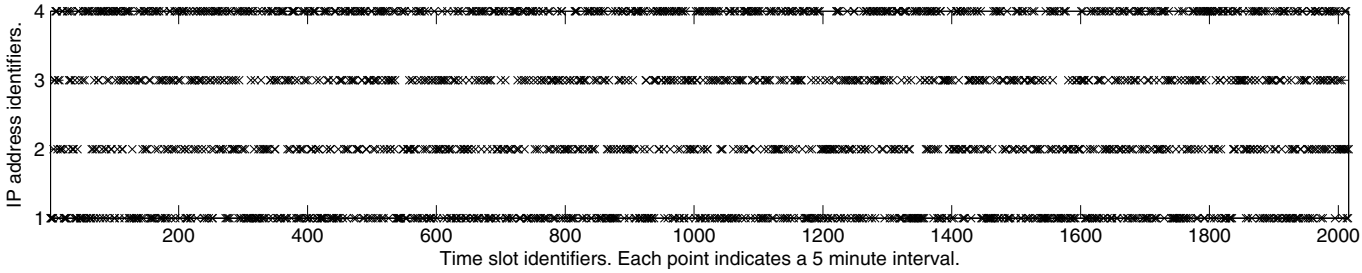

Fig. 6. Example plot showing hostname to IP mapping changing continuously.

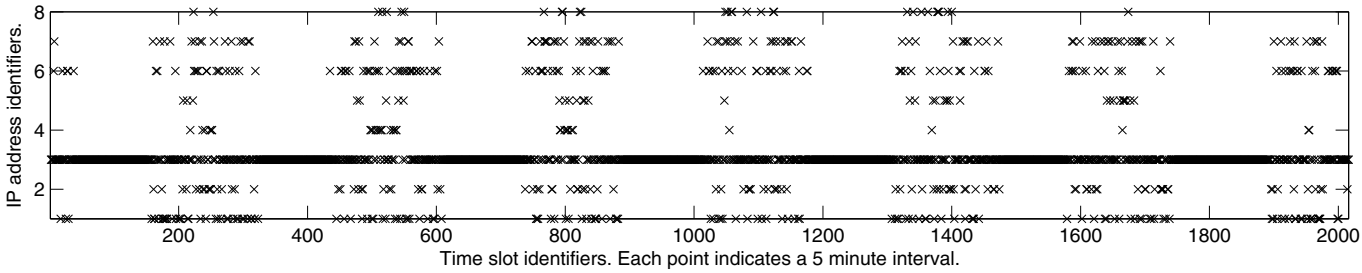

Fig. 7. Example plot showing hostname to IP mapping changing only during peak hours.

smaller number of secondary and tertiary cache locations, this usually results in clients being redirected to a farther location.

We see that the use of HTTP redirects helps YouTube to achieve a more fine granular load distribution among servers. The very interesting aspect of YouTube's use of HTTP redirections in load balancing is that it does not need any global coordination. The intra-tier load sharing only needs information gathering at the "local" level to see which servers are busier than which others. The local load sharing only requires sharing load information among nodes in the same location. Redirecting to a higher tier also does not need any global coordination since the target hostname can easily be obtained by the same static hashing mechanism. Each video ID, as we previously mentioned, maps to a unique hostname at each level of the cache heirarchy.

However, the use of redirections also has its own drawbacks. First of all, since YouTube first tries to load-balance locally, it leads to multiple redirections if the host that receives the redirection can not serve the video. In addition, each redirection requires a client to make an additional HTTP request, it also leads to higher delays before the video starts playing back. Moreover, inter-tier redirections generally leads a client to a distant cache location because the higher tier caches are only present at small number of locations.

V. Conclusion

In this paper, we examined YouTube's video distribution architecture, to uncover key aspects that determine how video servers are selected when users try to watch YouTube videos. We found that YouTube uses a large number of data-centers and caches that vary in size, geographic locations and other characteristics. We also uncovered key ideas behind the application-level redirection mechanism that YouTube uses and how that impacts the performance as observed by its users. These results demonstrate how one of the best content distribution network works. As video content delivery is getting increasingly popular, understanding current "best" practices provides valuable insights on how any such future systems can be developed and deployed.

In the future, we plan to build upon these findings to investigate other aspects of YouTube video delivery system such as the effects of popularity of videos and size of cache locations on server selection. We also plan to perform similar analysis of several other video content providers. The overarching goal of these projects is to identify the aspects of the current Internet architecture that make it difficult to build efficient large-scale video content distribution systems so as to inform the design of next-generation internetworks tailored towards large-scale content distribution.

References

[1] Adhikari, V. K., Jain, S., Chen, Y., and Zhang, Z.-L. Reverse Engineering the YouTube Video Delivery Cloud. In *IEEE HotMD 2011*.

[2] Adhikari, V. K., Jain, S., and Zhang, Z. YouTube Traffic Dynamics and Its Interplay with a Tier-1 ISP: An ISP Perspective. In *IMC '10* (2010), ACM.

[3] Cha, M., Kwak, H., Rodriguez, P., Ahn, Y.-Y., and Moon, S. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *IMC '07* (2007), ACM.

[4] Daigle, L. WHOIS Protocol Specification. RFC 3912 (Draft Standard), Sept. 2004.

[5] Gill, P., Arlitt, M., Li, Z., and Mahanti, A. Youtube traffic characterization: a view from the edge. In *IMC '07* (2007), ACM.

[6] Huang, C., Wang, A., Li, J., and Ross, K. Measuring and evaluating large-scale CDNs (withdrawn). In *Proc. of IMC* (2008).

[7] Zhou, R., Khemmarat, S., and Gao, L. The Impact of YouTube Recommendation System on Video Views. In *IMC '10* (2010), ACM.

[8] Zink, M., Suh, K., Gu, Y., and Kurose, J. Characteristics of youtube network traffic at a campus network - measurements, models, and implications. *Comput. Netw.* (2009).