Big Data Initiative: Effective Caching in Online Video Platforms

Rongrong Bao Atabak Hafeez
Tom Wiesing Jinbo Zhang

Abstract

Data on the internet grows by 50 percent annually. More than 90% of the data has been generated in recent years. This is the time for big data. How can we effectively transfer this huge amount of data?

We want to investigate caching techniques used by online video platforms and in particular by YouTube. YouTube is a leading online video provider worldwide. Before 2012, video streaming in YouTube was done using Real Time Messaging Protocol (RTMP)-based servers. This requires a streaming server and a near-continuous connection between the server and user. Requiring such a streaming server can increase implementation cost and RTMP-based video streaming is at risk of being blocked by firewalls. In 2012, this was replaced by HTTP (Hypertext Transfer Protocol) based servers known as MPEG DASH (Dynamic Adaptive Streaming over HTTP). HTTP is the protocol used by websites to bring their content to the users. By using this technology it was possible to use existing optimizations in the form of HTTP-Caching. This capability decreased total bandwidth costs associated with delivering the video since videos would be served from web-based caches rather than the origin server. This improved quality of service, since cached data is generally closer to the viewer and more easily retrievable.

The essay will explain and discuss different kinds of caching techniques, optimizations, data analysis and prediction techniques used by YouTube, including their advantages/disadvantages and potential social impacts.

Contents

1 Introduction

In Viktor Mayer-Schoenberg's book *Big Data: A Revolution That Transforms How we Work, Live, and Think* (with Kenneth Cukier), he refers to the fact that big data doesn't simply use random analysis (sampling) as a shortcut instead, it uses all data analysis and processing. There are four well-known features of big data: volume, velocity, variety, value.

The USIDC (United States Internet Data Center) pointed out that the data on the Internet will grow by 50 percent annually. More than 90% of the data today has been generated in recent years. In addition, the worldwide industrial equipment, such as automobiles, motion, vibration, temperature, and even the change of chemical substances in air humidity, also produced vast amounts of data. Internet, cloud computing, mobile Internet, car networking, phones, tablets, computers and a variety of sensors are the sources of big data. The ideal of big data is in the following aspects:

1. to provide products or service businesses that a large number of consumers can take advantage of big data precision marketing;

2. some small companies or businesses can take advantage of big data to do service transformation;

3. under the pressure of the traditional Internet, companies need to take advantage of the times of the value of big data.

Real-time big data analytics can be of immense importance to a business, but a business must first determine if the pros outweigh the cons in their particular situation, and if so, how those cons will be overcome . This is still a relatively new technology, so it is expected to evolve in the future and hopefully resolve some of its current challenges.

How big is big data? Using only the Internet as an example, during a day, the entire contents of the Internet can be engraved to produce 168 million DVDs; the amount of emails sent are as much as 294 billion; the amount of community posts sent over is two million, same as the number of letters printed in 770 years of Time magazine. As of 2012, the amount of data available on the Internet already jumped from TB (1024GB = 1TB) level to PB (1024TB = 1PB), from EB (1024PB = 1EB) to the ZB (1024EB = 1ZB) level. The IDC (International Data Corporation) showed that by 2020, the size of data generated worldwide will reach 44 times the amount of today.

The Internet is mostly based on and benefits from Big Data. In this paper we will discuss YouTube – the biggest video-sharing website in the world – as a typical example of a video sharing platform that uses big data. YouTube has an average of one hour of video uploads per second, and an average of 35 million video uploads daily. According to the article 120+ amazing YouTube statistics, YouTube now has more than 1 billion users, which is almost a third of total daily consumption of the world's Internet video viewing time. By 2015, the YouTube viewing time increased by 60%, which is on the highest level of growth. Without the process of development of Big Data, YouTube will not be able to keep up with the needs of the users in the world.

2 What is Caching?

Before we delve into the specifics of how YouTube does caching and the advantages and disadvantages of the different kinds of caching, we will explain what caching itself is and explain the different techniques used by YouTube.

Caching is a method to store data in a cache. A cache is a basically temporary storage area on the local hard disk of a user. This storage area may contain data such as HTML (Hypertext Markup Language - a language to describe web documents or pages) pages, images, files, and web Projects in order to make it faster for the user to access it, which helps improve the efficiency of the computer and the overall efficiency of the task at hand. The important thing to note here is that it occurs mostly without the user being aware of exactly which data has been stored in the cache. For example, when a user returns to a web page they have recently accessed, the browser can pull those files from the cache instead of the original server because it has stored the user's activity. The storing of that information saves the user time by getting to it load faster, reduces local memory usage and lessens the traffic on the network.

2.1 Caching and Buffering in YouTube

To explain caching in YouTube and how it has changed and updated, we also need to understand the notion of buffering. Buffering involves pre-loading data into a certain area of memory known as a "buffer" in the local machine. This is basically a a more specific kind of caching which YouTube uses to store the loaded video on to the local memory of the browser in use.

In 2013, YouTube made a design decision in their buffering system where they moved from Real Time Messaging Protocol (RTMP)-based Dynamic Streaming to MPEG DASH (Dynamic Adaptive Streaming over HTTP).

To a user, this is important because it changes the extent to which you can cache your YouTube video before viewing it. Basically, while YouTube was using RTMP-based Dynamic Streaming, if a user had a relatively slow connection, which would not allow them to view the video as smoothly as one would want, he/she could pause the video and view it later when the whole video is buffered or cached to the local storage of the browser. This technology required a near-continuous connection between the server - the original storage location where one's local computer is connected to retrieve the video - and the player on one's browser.

With YouTube's shift to using MPEG DASH, being able to buffer the whole video and then coming back to it was no longer possible. MPEG DASH uses standard HTTP (Hypertext Transfer Protocol - a set of standards which defines how messages are formatted and transmitted across the World Wide Web) web servers to deliver streaming content, obviating the need for a streaming server. In addition, HTTP packets are firewall (a set of programs that block unauthorised access to a computer) friendly and can utilise HTTP caching mechanisms on the web. To an average user, this means that now when he or she pauses a video because the video is not very smooth, the video buffers for a while and then

stops buffering. The cache, hence, does not at this stage store the whole video. The only condition in which the video may cache the whole video is if you start the video and watch the whole way through. In this case, if the user turns off the internet, it would be possible for the user to re-watch the video without having to reload it from the server.

2.2 Caching on the Servers

So why did YouTube decide to make this significant change in their protocol? This is mainly because this makes streaming high quality videos more efficient by caching on the servers.

Caching on the server means that when a user connects to a server, the connection is not direct. There is something in-between known as a caching server. The caching server acts as a web proxy server so it can serve those requests. After a web proxy server receives requests for web objects, it either serves the requests or forwards them to the origin server (the web server that contains the original copy of the requested information). Using MPEG DASH, YouTube was able to exploit this feature resulting in higher resolution videos being available to the user in a more efficient manner.

This form of caching also aims to make sure that the user can experience the best quality of video according to the available bandwidth speed. For example, a video may start playing at 360P resolution, but if the system detects that the bandwidth is now able to handle 720P, it will shift to that. Thus, using MPEG DASH, YouTube only caches chunks of the video and only a small chunk of them are loaded when a video is paused.

3 Advantages of Caching

As the data size on the Internet increases significantly, caching plays a more and more important role in this big data era. Elegantly designed caching techniques can notably reduce the burden on a server. As a typical high-traffic website and a leading website in video-sharing field, YouTube uses two techniques to deliver web contents effectively – Distributed Caching and Content Delivery Network. We will have a close look on these two techniques.

3.1 Distributed Caching

A distributed cache is an extension of the traditional concept of cache used in a single locale. A distributed cache may span multiple servers so that it can grow in size and in transactional capacity[10]. With distributed cache, a website can response more requests simultaneously. Before we talk about distributed caching technique more deeply, we will introduce some terms to help us better understand the idea behind it. In communication networks, a *node* is either a connection point, a redistribution point, or a communication endpoint. Speaking of distributed network, the nodes are clients, servers or peers. By storing the data not on the individual web server's memory but on other cloud resources, *distributed cache* offers high throughput, low-latency access to commonly accessed application data.

One significant advantage for distributed caching, is that when the application scales by adding or removing servers, or when servers are replaced due to upgrades or faults, the cached data remain accessible to every server that runs the application. For example, If we have data `1, 2, 3, 4, 5` and servers `A, B, C`, and we store `1, 2, 5` in `A`, `3, 4, 5` in B, `1, 2, 3, 4` in `C`. If one server is down, we won't lose any data, because every piece of data has a copy in another server.

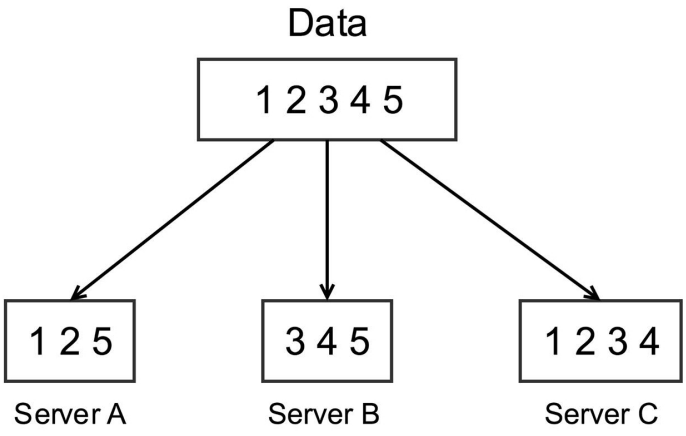

Figure 1: An example illustrated how distributed caching works

By distributing data efficiently and effectively, caching can dramatically improve application responsiveness. Comparing to access data from relational databases, which requires a lot computations, accessing data from cache is much faster. Caching works best for application workloads that do more reading than writing of data, and when application users share a lot of common data. So this is why such technique is very useful for YouTube: those popular videos will be watched again and again.

Figure 2: A very popular video in YouTube which is watched more than 700 million times

In order to get data from cache, the data must exist in cache before retrieving. There are several strategies for putting data into a cache [1]:

- On Demand

  The application tries to retrieve data from cache, and when the cache doesn't have the data (a "miss"), the application stores the data in the cache so that it will be available the next time. The next time the application tries to get the same data, it finds what it's looking for in the cache (a "hit").

- Background Data Push

  Background services push data into the cache on a regular schedule, and the application always pulls from the cache. This approach works great with high latency data sources that don't require you always return the latest data.

Strategy 2 is clearly the better for YouTube. YouTube will the count the watching times in order to filter the popular videos from non-popular ones, or count the watching

times per minute in order to filter the trendy videos from the outdated ones, thus the background services have enough information to determine which videos will be pushed into cache. Also, once the video is uploaded, it will not change anymore, so the cached data will always stay tuned with the original one.

When we type `www.youtube.com` in our browser, how can they determine which server will respond our request? There are several different techniques for determining the responsive server, in this paper, we will only focus on one of the them – *Content Delivery Network*.

3.2 Content Delivery Network

A *Content Delivery Network* (or CDN for short) is a system of distributed network that delivers web pages and other web contents to a user based on the geographic locations of the user, the origin of the web page and a content delivery server [2].

This service is effective in speeding the delivery of content of websites with high traffic and websites that have global reach. The closer the CDN server is to the user geographically, the faster the content will be delivered to the user. CDNs also provide protection from large surges in traffic.

Servers nearest to the website visitor respond to the request. The CDN copies the pages of a website to a network of servers that are dispersed at geographically different locations, caching the contents of the page. When a user requests a web page that is part of a content delivery network, the CDN will redirect the request from the originating site's server to a server in the CDN that is closest to the user and deliver the cached content. The CDN will also communicate with the originating server to deliver any content that has not been previously cached.

Thus, the CDN can reduce traffic on the primary network, which improves video content and web performance overall.

4 Disadvantages of Caching

While caching videos in the users web browser has many useful effects as discussed above it also has several bad side effects. In this section we will first explain some of the technical side effects and then continue discuss some social implications.

4.1 Technical side effects

In order to take advantage of caching of videos the browser needs to create caches for all the videos that the user plays. Additionally every time a video is played the browser has to check if a cached version of the video already exists. If this is the case the video can be played from the cached version. If this is not the case however the browser has to either create a cached version of the video or play a direct version from the server. Both of these decisions have a negative performance impact since the video has to be downloaded from the server which is slower then playing a local version.

Furthermore a cached version might not always be up-to-date. It is conceivable that when a news station has uploaded a video to YouTube and later on finds an error in their report that they might update their video in order to provide the most correct and up-to-date information. If this video has been cached by a viewer of the video they might no longer have the newest version available in their cache. Thus in order to use caches properly every time the cache is used the web browser needs to check if it is still up-to-date. This process can be further complicated if only parts of the video are cached. Furthermore if the cache is invalidated (meaning it is found to no longer be up-to-date) it is inefficient to reload the entire video if only parts of it have changed.

It is also possible that some videos are cached locally even though the user does not want to watch them. This is for example the case when advertisements are played before the video. They are not desired by the user and might nonetheless be cached. Sometimes the users might also watch a few seconds of the beginning of the video and then decide not to watch the remaining part. This can cause both a waste of bandwidths and local hard drive spaces.

Since caching technically copies the videos from the server to the hard disks of the local user, the video is no longer stored in only one central location. That means that even if the video has to be deleted, for example for legal reasons, it might still be available locally. This might result in legal hurdles if a license has to be obtained in order to show the video.

Additionally some of the videos the user watches might remain cached on the users hard drive even after they have watched the video. While a single video does not take too much space this can cause a problem if there are many of these cached videos. It is especially important for mobile users as these typically have less space available on their devices. Furthermore the user might not want a record of the videos they watched for privacy reasons.

4.2 Social implications

When using predictive caching YouTube (or other online video providers) are using data from the user. They try to analyse the videos the user has watched in the past and want to predict which videos the user might watch in the future. Some users might not desire getting videos predicted for several reasons. Especially when on a connection with limited bandwidth the user might object to having this bandwidth used up with videos that he might not watch anyways.

Additionally several users might not want their video history to be recorded. This can easily be considered an intrusion into the users privacy. This information could also be used against the user especially when the wrong videos are predicted by the algorithm.

YouTube sells anonymized customer data to advertisers. If sold to advertisers, this data is commonly used to predict products the user might be interested in. Related adverts are then played on the next video the user watches. This brings money to the content provider but does not give any significant advantage to the user.

5 Conclusion

- what is good about caching (try to summarize advantages section)
- what is bad about caching (try to summarize disadvantages section)
- harmonise this:
  - it cannot be solved by technical solutions alone
  - social impacts need to be considered
  - like every other use of big data this involves privacy issues
  - in general: data collection can be both good and bad
    * bad due to privacy
    * good because it can help make things a lot more efficient
    * options should be offered to the user (so that the users can decide if they want a better expericence or better privacy protection)

6 Bibliography

[1] Rick Anderson, Tom Dykstra, and Mike Wasson. Distributed caching (building real-world cloud apps with azure). http://www.asp.net/aspnet/overview/developing-apps-with-windows-azure/building-real-world-cloud-apps-with-windows-azure/distributed-caching, 2015. [Online; accessed November 2015].

[2] Vangie Beal. What is content delivery network (cdn)? webopedia. http://www.webopedia.com/TERM/C/CDN.html, 2015. [Online; accessed November 2015].

[3] Citrix. What is caching? https://www.citrix.com/glossary/caching.html. [Online; accessed November 2015].

[4] Apache Traffic Server Documentation. Http proxy caching. http://trafficserver.readthedocs.org/en/latest/admin/http-proxy-caching.en.html. [Online; accessed November 2015].

[5] Jan Ozer. What is mpeg dash? http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-MPEG-DASH-79041.aspx, November 2011. [Online; posted November 2011].

[6] Kaushik Pal. Weighing the pros and cons of real-time big data analytics. https://www.techopedia.com/2/31245/technology-trends/big-data/weighing-the-pros-and-cons-of-real-time-big-data-analytics. [Online; accessed November 2015].

[7] Tim Siglin. Online video jumps on the big data bandwagon. http://www.streamingmedia.com/Articles/ReadArticle.aspx?ArticleID=91353&PageNum=2, August 2013. [Online; posted August/September 2013].

[8] Craig Smith. 120 amazing youtube statistics. http://expandedramblings.com/index.php/youtube-statistics/. [Online; accessed November 2015].

[9] Heriot-Watt University. Pros and cons of web caching. http://www.macs.hw.ac.uk/~hamish/3NI/topic172.html. [Online; accessed November 2015].

[10] Wikipedia. Distributed cache. https://en.wikipedia.org/w/index.php?title=Distributed_cache&oldid=681699661, 2015. [Online; accessed November 2015].