Notably, YouTube used two techniques to deliver web contents effectively – Distributed Caching and Content Delivery Network.

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.[1]. Before we talked about distributed caching technique more deeply, we 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 remains accessible to every server that runs the application. For example, If we have data `1, 2, 3, 4, 5` and servers `A, B, C` If we store `1, 2, 5` in `A`, `3, 5, 4` 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.

Graph needed here.[2]

By distributing data efficiently and effectively(typically by different hashing techniques), 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.

[1] `https://en.wikipedia.org/wiki/Distributed_cache`

[2] Someone please?

Figure 1: 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:[3]

- On Demand / Cache Aside

  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.

[3] http://www.asp.net/aspnet/overview/developing-apps-with-windows-azure/building-real-world-cloud-apps-with-windows-azure/distributed-caching

- Circuit Breaker

  The application normally communicates directly with the persistent data store, but when the persistent data store has availability problems, the application retrieves data from cache.

Strategy 2 is clearly the best 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 won't 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 the paper, we will mainly focus on one of the them – Content Delivery Network.

A Content Delivery Network is a system of distributed servers (network) that deliver web pages and other Web content to a user based on the geographic locations of the user, the origin of the web page and a content delivery server.[4]

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 webpage 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.[5]

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

[4] `http://www.webopedia.com/TERM/C/CDN.html`

[5] These three part need to polish.