Finding a Needle in Haystack: Facebook's Photo Storage

OSDI'10

201924169 Ju-Won Park

June 3, 2019

Problem statement

- Facebook currently stores over 260 billion images, which translates to over 20 petabytes of data. Users upload one billion new photos (60 terabytes) each week and Facebook serves over one million images per second at peak.

Solution approach

- In this paper they instrumented the entire Facebook photo-serving stack obtaining traces representative of Facebooks full workload.
- The Haystack has three main components: Store, Directory and Cache. The Store keeps all the actual data in the form of huge files. The Directory keeps metadata about the images and the Cache functions as an internal cache, which shelters the Store from most frequent requests. Next the authors describe single details, such as insertions, deletions, mapping of image locations and recovery from failures.

Strong points

- Improvements of the photo cache involves research on improving cache eviction policy algorithms.
- Improvements on where to place the photo resize functionality.

Weak points / Limitations

- This paper does not introduce any particularly interesting ideas and is not written in an informative way.
- It is also not explained in which cases requests from client browsers go to CDN, and when they go directly to Haystack Cache.

Questions

- They mention that a photo will only be cached if the request comes from a write enabled sever, with the intuition that recently written photos are more popular. Wouldn't it be easier/better to use some sort of timestamp on the photo?

New ideas / Comments

- Use the notion that when people view a photo in a particular albums, they also view other photos in that album to optimize the read operations.
- If we store every photo in an album in a contiguous region in a physical volume, we can then retrieve every photo in that album by a single disk access and put them in the cache.