

CSE 486/586 Distributed Systems Web Content Distribution---2 Case Study: Facebook Haystack

Steve Ko
Computer Sciences and Engineering
University at Buffalo

CSE 486/586

Recap

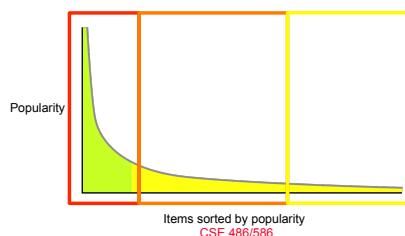
- Engineering principle
 - Make the common case fast, and rare cases correct
 - (From Patterson & Hennessy books)
 - This principle cuts through generations of systems.
- Example?
 - CPU Cache
- Knowing common cases == understanding your workload
 - E.g., read dominated? Write dominated? Mixed?

CSE 486/586

2

Recap

- “Hot” vs. “very warm” vs. “warm” photos
 - Hot: Popular, a lot of views
 - Very warm: Somewhat popular, still a lot of views
 - Warm: Unpopular, but still a lot of views in aggregate



CSE 486/586

3

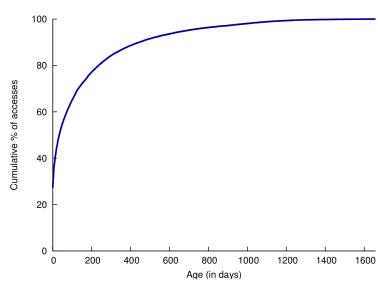
“Very warm” and “warm” Photos

- Hot photos are served by a CDN.
- Warm photo characteristics
 - Not so much popular
 - Not entirely “cold,” i.e., occasional views
 - A lot in aggregate
 - Does not want to cache everything in CDN due to diminishing returns
- Facebook stats (in their 2010 paper)
 - 260 billion images (~20 PB)
 - 1 billion new photos per week (~60 TB)
 - One million image views per second at peak
 - Approximately 10% not served by CDN, but still a lot

CSE 486/586

4

Popularity Comes with Age



CSE 486/586

5

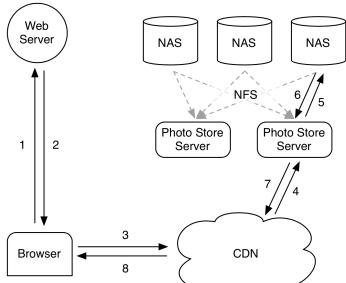
Facebook Photo Storage

- Three generations of photo storage
 - NFS-based (today)
 - Haystack (today): Very warm photos
 - f4 (next time): Warm photos
- Characteristics
 - After-CDN storage
 - Each generation solves a particular problem observed from the previous generation.

CSE 486/586

6

1st Generation: NFS-Based



CSE 486/586

7

1st Generation: NFS-Based

- Each photo → single file
- Observed problem
 - Thousands of files in each directory
 - Extremely inefficient due to meta data management
 - 10 disk operations for a single image: chained filesystem inode reads for its directory and itself & the file read
- In fact, a well-known problem with many files in a directory
 - Be aware when you do this.
 - The inode space (128 or 256 bytes) runs out.
 - A lot of operations necessary for meta data retrieval.

CSE 486/586

8

CSE 486/586 Administrivia

- PA4 due 5/6 (Friday)
- Final: Thursday, 5/12, 8am – 11am at Knox 20

CSE 486/586

9

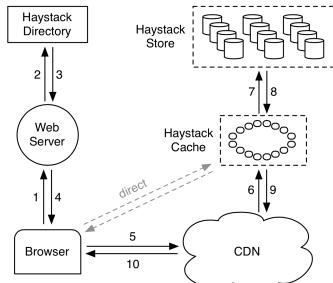
2nd Generation: Haystack

- Custom-designed photo storage
- **What would you try? (Hint: too many files!)**
 - Starting point: One big file with many photos
- Reduces the number of disk operations required to one
 - All meta data management done in memory
- Design focus
 - Simplicity
 - Something buildable within a few months
- Three components
 - Directory
 - Cache
 - Store

CSE 486/586

10

Haystack Architecture



CSE 486/586

11

Haystack Directory

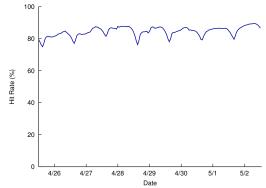
- Helps the URL construction for an image
 - `http://(CDN)/(Cache)/(Machine id)/(Logical volume, Photo)`
 - Staged lookup
 - CDN strips out its portion.
 - Cache strips out its portion.
 - Machine strips out its portion
- Logical & physical volumes
 - A logical volume is replicated as multiple physical volumes
 - Physical volumes are stored.
 - Each volume contains multiple photos.
 - Directory maintains this mapping

CSE 486/586

12

Haystack Cache

- Facebook-operated CDN using DHT
 - Photo IDs as the key
- Further removes traffic to Store
 - Mainly caches newly-uploaded photos
- High cache hit rate (due to caching new photos)

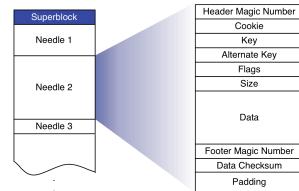


CSE 486/586

13

Haystack Store

- Maintains physical volumes
- One volume is a single large file (100GB) with many photos (needles)

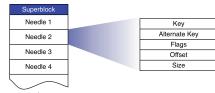


CSE 486/586

14

Haystack Store

- Metadata managed in memory
 - (key, alternate key) to (flags, size, volume offset)
 - Quick lookup for both read and write
 - Disk operation only required for actual image read
- Write/delete
 - Append-only
 - Delete is marked, later garbage-collected.
- Indexing
 - For fast memory metadata construction



CSE 486/586

15

Daily Stats with Haystack

- Photos uploaded: ~120 M
- Haystack photos written: ~1.44 B
- Photos viewed: 80 – 100 B
 - Thumbnails: 10.2%
 - Small: 84.4%
 - Medium: 0.2%
 - Large: 5.2%
- Haystack photos read: 10 B

CSE 486/586

16

Summary

- Two different types of workload for a social networking Web service
 - Posts: read/write
 - Photos: write-once, read-many
- Photo workload
 - Zipf distribution
 - “Hot” photos can be handled by CDN
 - “Warm” photos have diminishing returns.
- Haystack: Facebook’s 2nd generation photo storage
 - Goal: reducing disk I/O for warm photos
 - One large file with many photos
 - Metadata stored in memory
 - Internal CDN

CSE 486/586

17