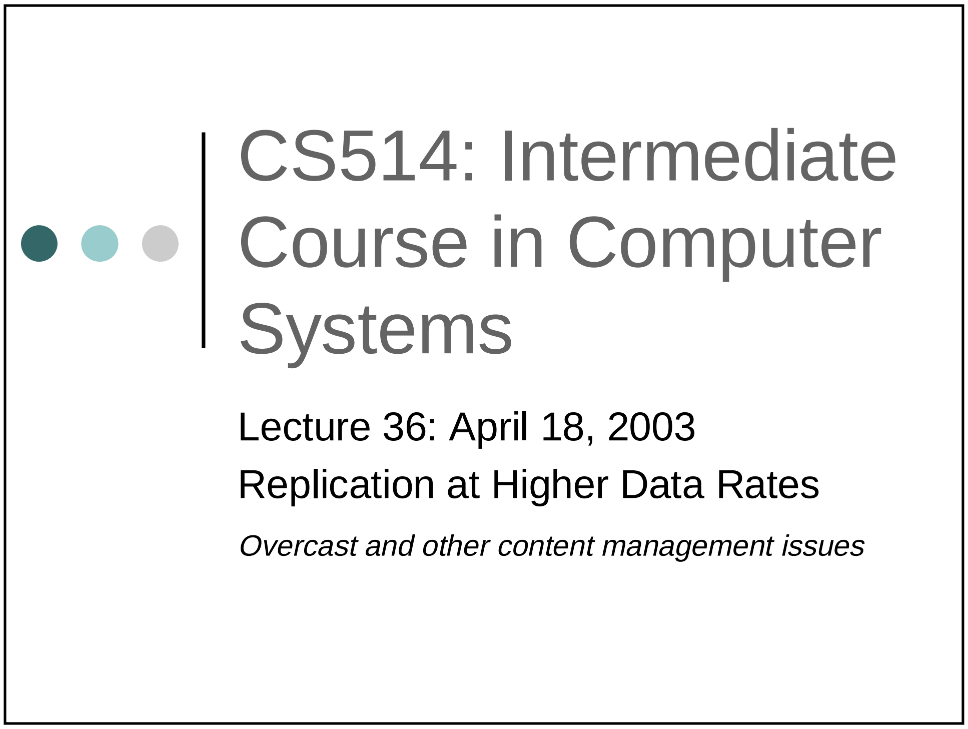
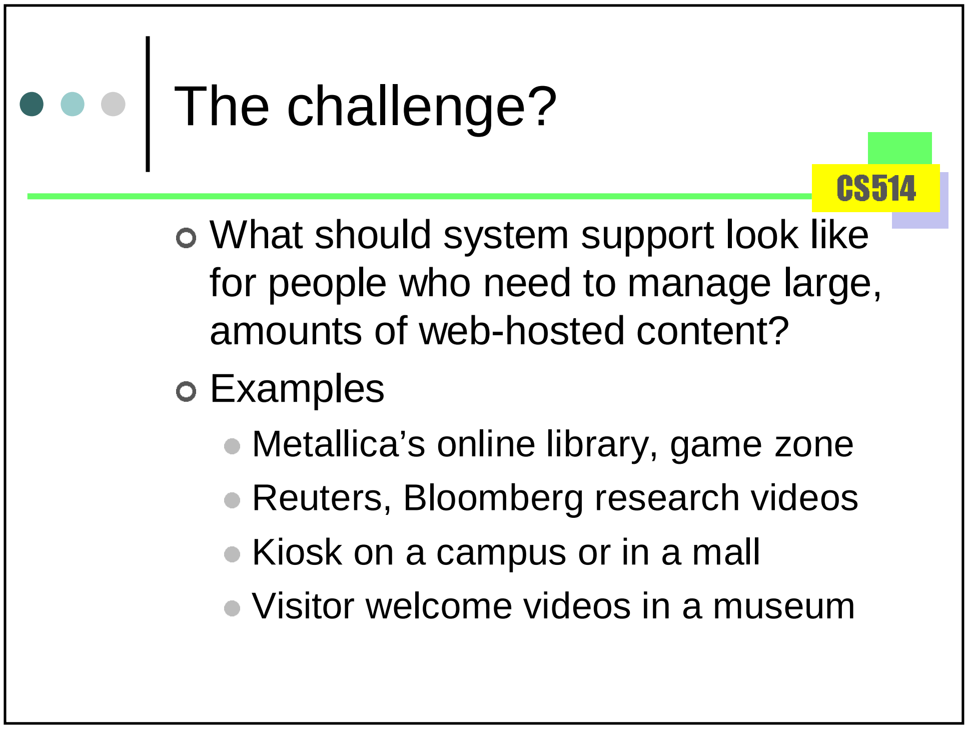


CS514: Intermediate Course in Computer Systems

Lecture 36: April 18, 2003

Replication at Higher Data Rates

Overcast and other content management issues



The challenge?

- What should system support look like for people who need to manage large, amounts of web-hosted content?
- Examples
  - Metallica’s online library, game zone
  - Reuters, Bloomberg research videos
  - Kiosk on a campus or in a mall
  - Visitor welcome videos in a museum

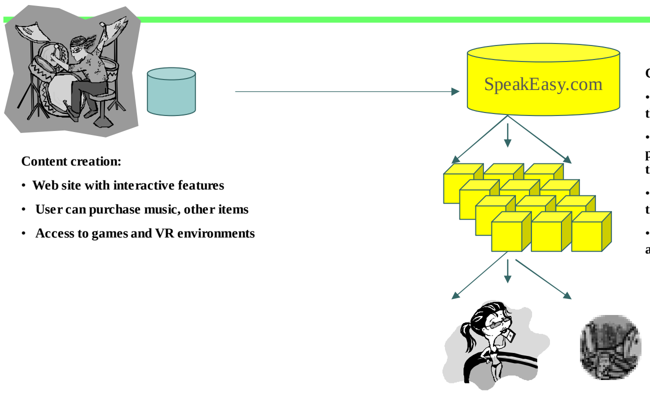
What makes it hard?

CS514

- Developer builds the “site” on a small set of computers
  - Probably uses fancy site-design tools
  - But the system runs in-house
- Then hands off to a data farm
- And they may want to hand some objects to Akamai or other hosting companies

Overall Architecture

CS514



Content creation:

- Web site with interactive features
- User can purchase music, other items
- Access to games and VR environments

Content hosting:

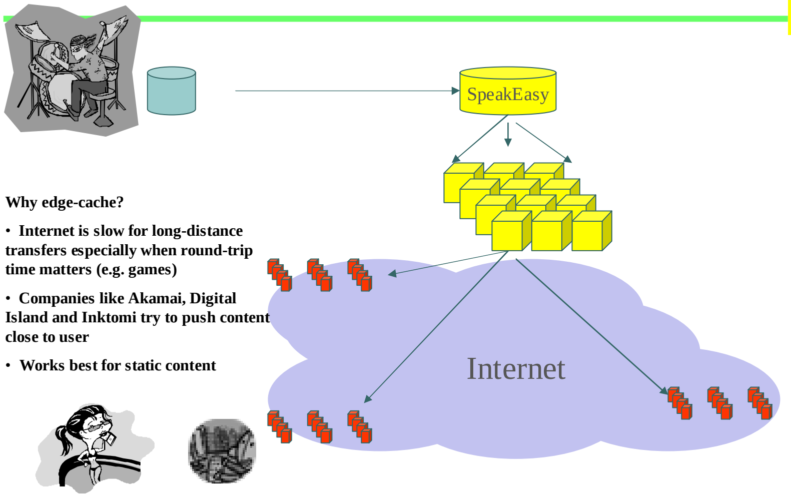
- Large numbers of copies of any objects that are static
- For dynamically generated but not personalized content, run little programs on the servers (servlets)
- For interactive purchases, requests go to the SpeakEasy transactional system
- Also hosts software for authenticating use and downloading decryption keys

Content playback

- Favor streaming of encrypted data for high-value content (much harder to rip)
- Anticipate many access points per-user: cell phone, home computer, PDA
- User highly motivated to obtain broadband access for computer to exploit high-bandwidth content such as music videos, online games, VR role-playing games, etc. Can purchase content from the site so initial access is in part just a lure to get the user online

Edge Caching: Used if SpeakEasy performance looks like a problem

CS514



Why edge-cache?

- Internet is slow for long-distance transfers especially when round-trip time matters (e.g. games)
- Companies like Akamai, Digital Island and Inktomi try to push content close to user
- Works best for static content

What makes it hard?

CS514

- At "access time" must
  - Authenticate user
  - Validate that access is legitimate, perhaps charge a per-use fee
  - Track copies of the content within the system
  - Defend against DDoS attacks or attempts to steal content

A user-friendly but secure site

- Media owner's goal:
  - Make it extremely hard to rip content
  - Break content into many pieces.
    - Make it hard to rip the site's overall content without stealing most or all of these pieces.
    - Watch for users who download unusually large amounts of content, do so in unusually short amounts of time, or seem to access from an unusual number of platforms.
  - But don't create a barrier that turns off users

A user-friendly but secure site

- User experience goals
  - User only has to register once and it is easy to do
  - Some period of free access for users who put the Metallica CD into their computer CD drive – make this as easy as possible
  - Perhaps pop-up window from CD gives unrestricted access for a period of time without requiring any kind of authentication at all?

Accounting

CS514

- Media owner might need to track
  - Which users are accessing content
  - What content they access
  - What IP address they come in from
  - What class of device they are on (PDA, cell phone, PC)
  - Category of connection they are using
- Use this data to
  - Detect users who are gaming the system, e.g. by sharing login id's
  - Customize content to match market

What makes it hard?

CS514

- Mustn't violate privacy protections
- Copies are no longer under direct control
  - The server "experience" is out there
  - The client experience is way out there
  - The truth is out there…
- Lack tools for gathering this data

What makes it hard?

- Issue of supporting updates
  - With static media, our challenge isn't so tough
    - Just use some form of encryption
    - Security policy would then enforce authentication
    - But would still need to worry that legitimate user A starts to make unauthorized copies
  - With dynamic updates, much harder!

Technology options

- Could depend entirely on caching
  - If someone access file A,
    - Check for a fresh new copy
    - If found, download it to cache
    - Next user will get a cache hit
  - This is pretty popular!
  - But per-user access keys seem incompatible with such a scheme

Technology options

CS514

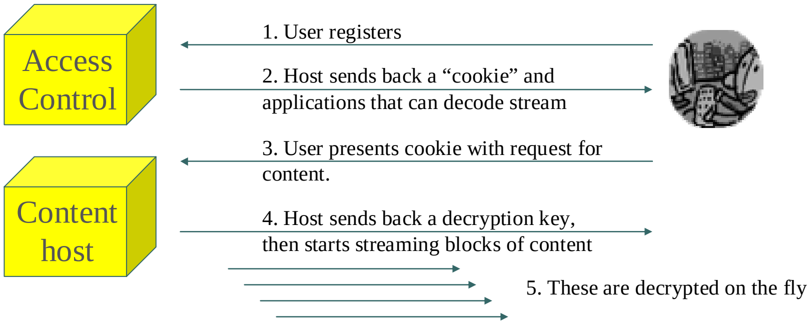
- Or, could trust the servers
  - Sometimes, you put encrypted data on the server
  - Plus a “servlet” that
    - Obtains a key on your behalf
    - Decrypts data, then reencrypts with private key
    - Finally, ships you the resulting personal copy (perhaps streaming)

Encryption and Streaming

CS514

- Media owner will encrypt high-value content
  - Videos, tracks from new and older CDs, material associated with games, etc
  - Each “object” has its own encryption key. Data is encrypted in blocks of a few k-bytes at a time
- User accesses these objects via a small application downloaded from site
  - It authenticates access
  - Obtains keys for objects that will be downloaded
  - Downloads streams and decrypts on the fly

Encryption and Streaming



Access Control

Content host

1. User registers

2. Host sends back a “cookie” and applications that can decode stream

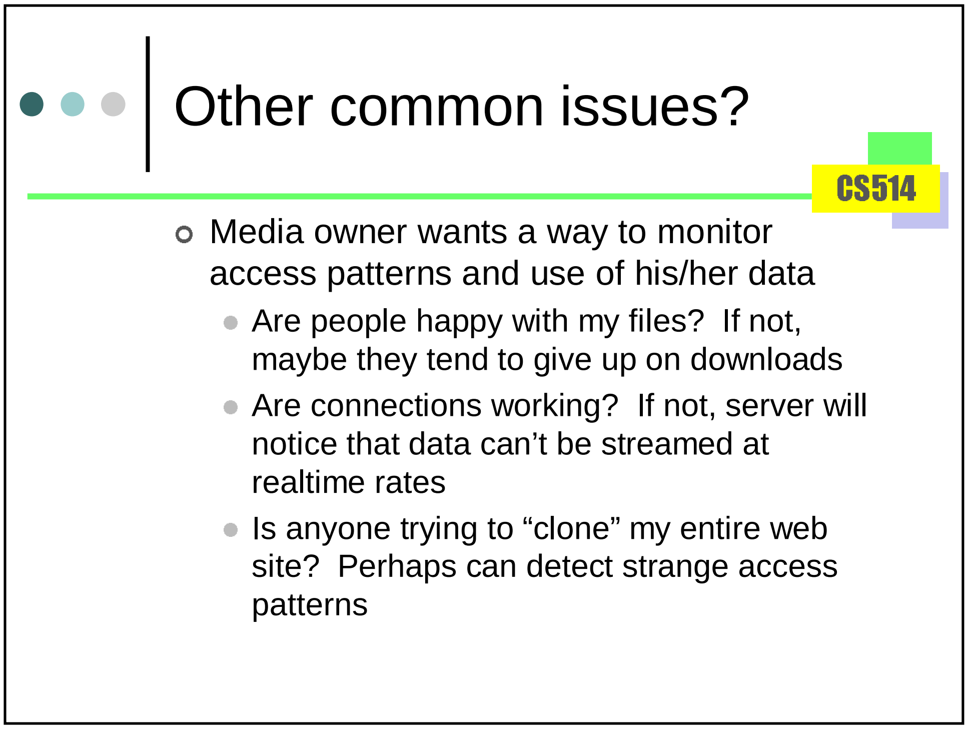
3. User presents cookie with request for content.

4. Host sends back a decryption key, then starts streaming blocks of content

5. These are decrypted on the fly

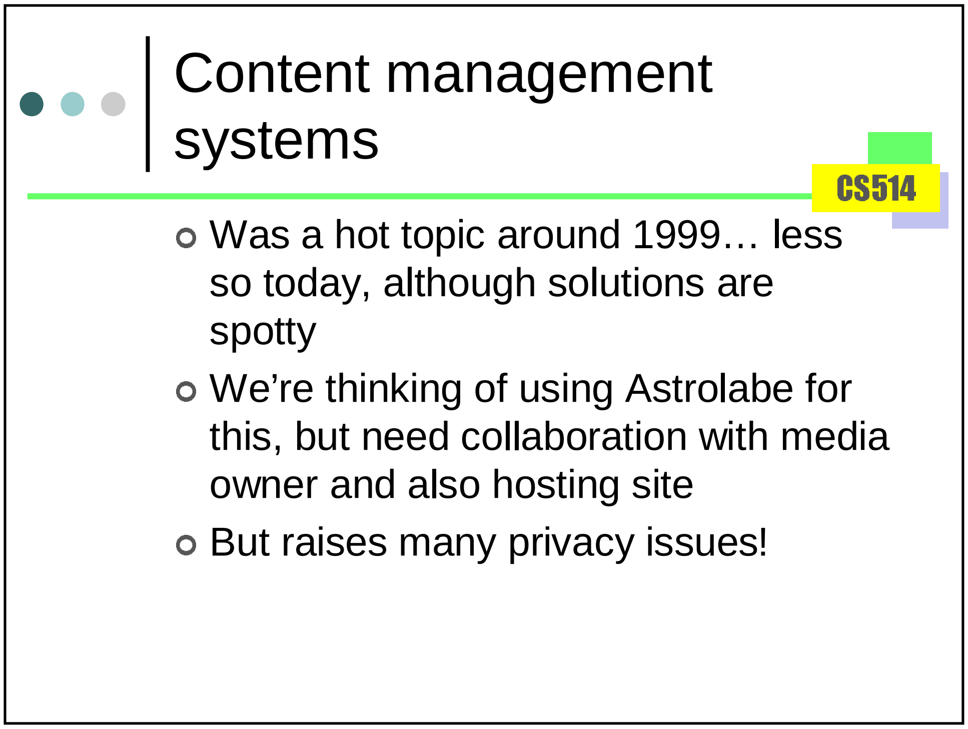
Overcast

- This system focuses on how those trusted servers could update very large media files
- Focus is on forms of multicast using overlay networks
- Click here for John Janotti’s OSDI talk



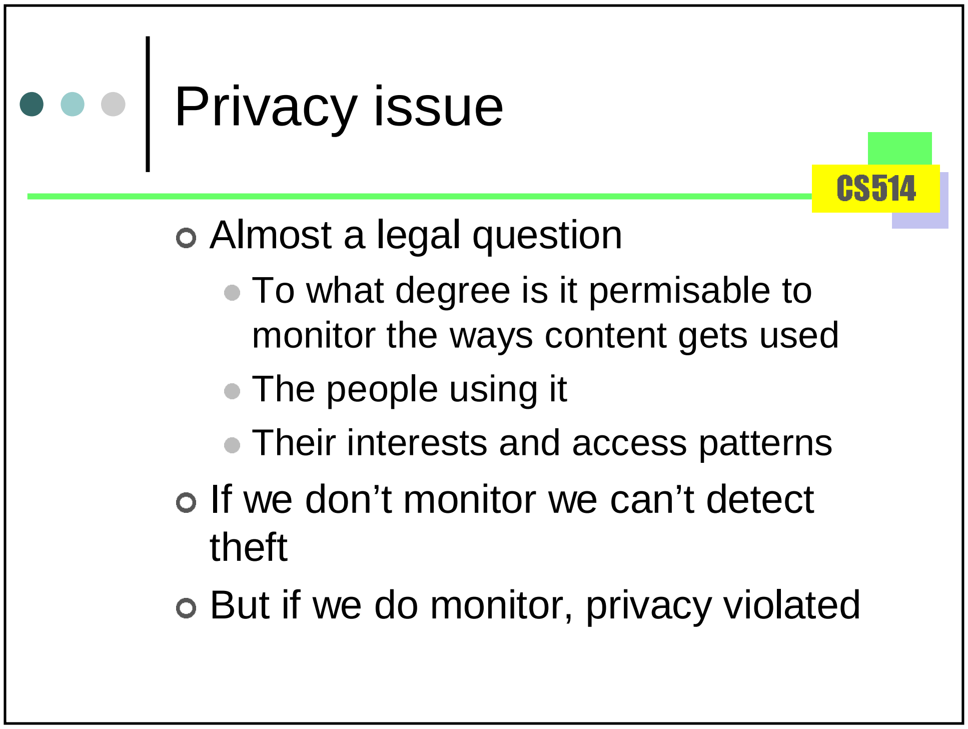
Other common issues?

- Media owner wants a way to monitor access patterns and use of his/her data
  - Are people happy with my files? If not, maybe they tend to give up on downloads
  - Are connections working? If not, server will notice that data can't be streamed at realtime rates
  - Is anyone trying to "clone" my entire web site? Perhaps can detect strange access patterns



Content management systems

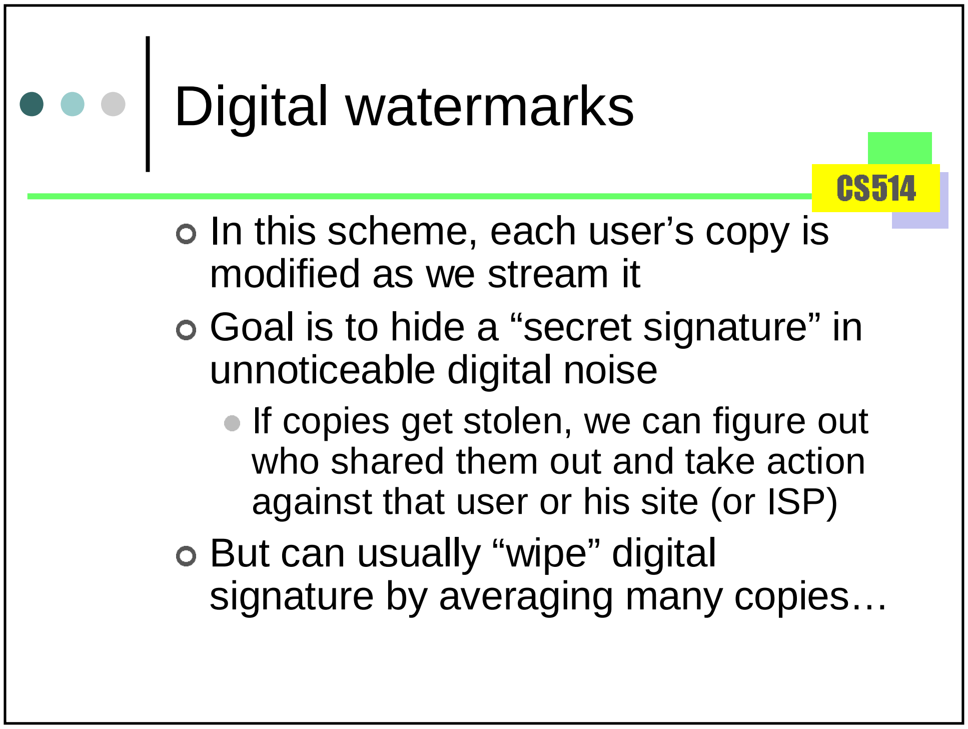
- Was a hot topic around 1999… less so today, although solutions are spotty
- We're thinking of using Astrolabe for this, but need collaboration with media owner and also hosting site
- But raises many privacy issues!



Privacy issue

CS514

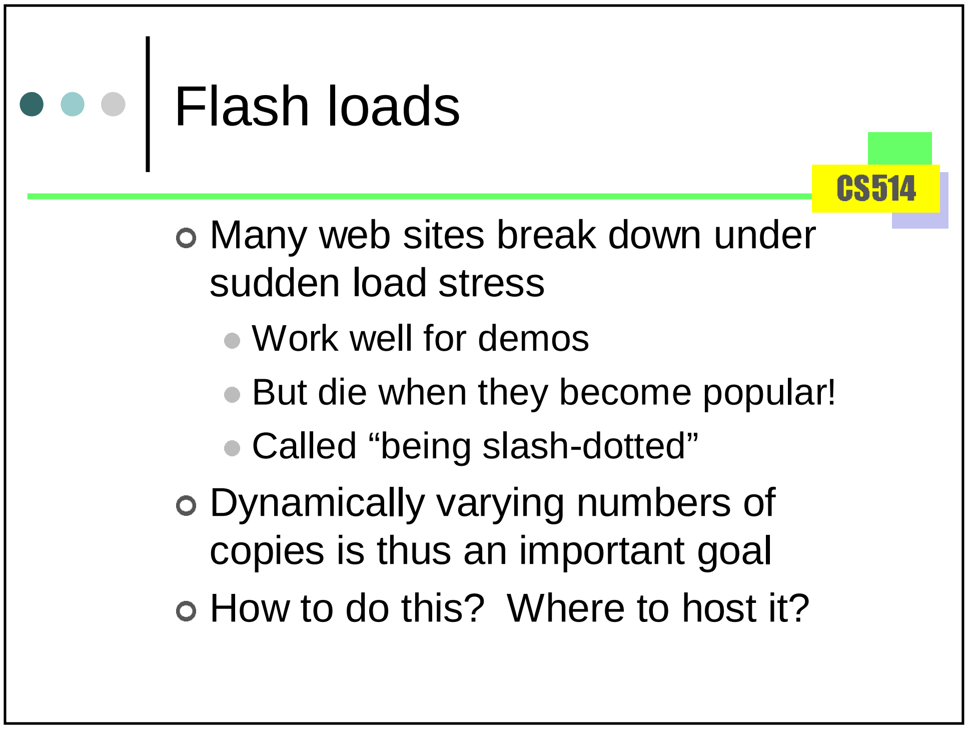
- Almost a legal question
  - To what degree is it permisable to monitor the ways content gets used
  - The people using it
  - Their interests and access patterns
- If we don't monitor we can't detect theft
- But if we do monitor, privacy violated



Digital watermarks

CS514

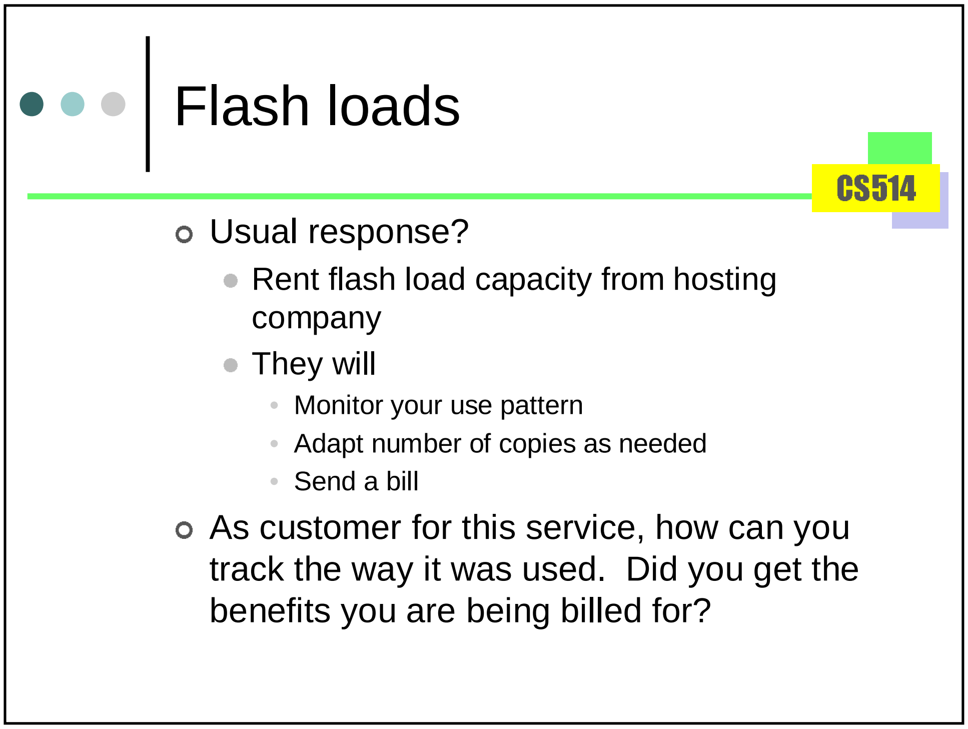
- In this scheme, each user's copy is modified as we stream it
- Goal is to hide a "secret signature" in unnoticeable digital noise
  - If copies get stolen, we can figure out who shared them out and take action against that user or his site (or ISP)
- But can usually "wipe" digital signature by averaging many copies…



Flash loads

CS514

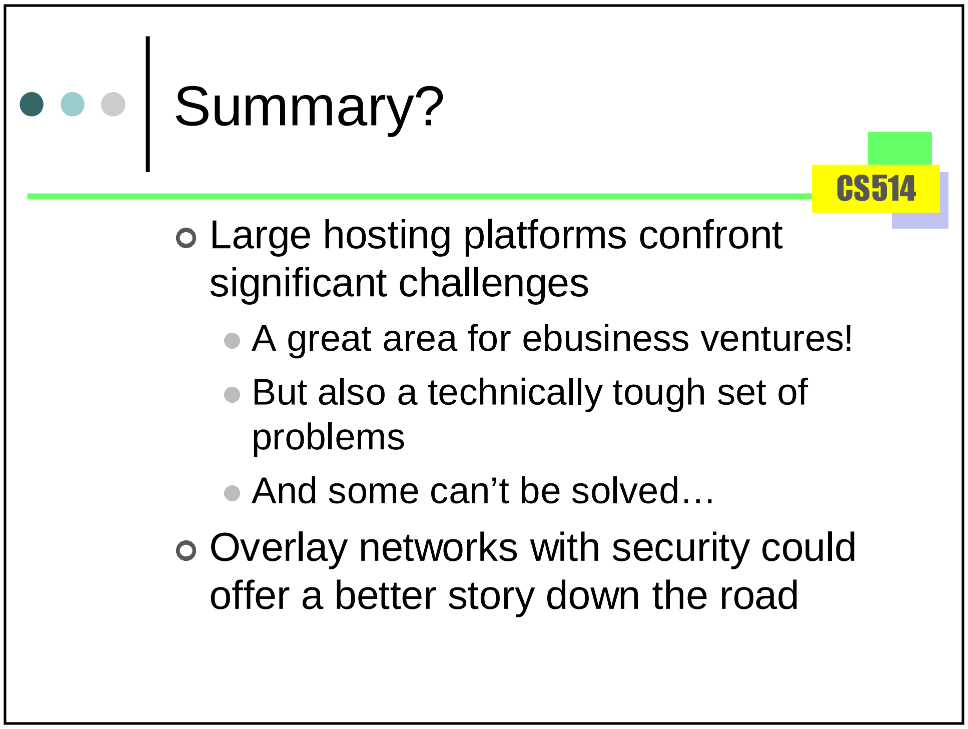
- Many web sites break down under sudden load stress
  - Work well for demos
  - But die when they become popular!
  - Called “being slash-dotted”
- Dynamically varying numbers of copies is thus an important goal
- How to do this? Where to host it?



Flash loads

CS514

- Usual response?
  - Rent flash load capacity from hosting company
  - They will
    - Monitor your use pattern
    - Adapt number of copies as needed
    - Send a bill
- As customer for this service, how can you track the way it was used. Did you get the benefits you are being billed for?



Summary?

CS514

- Large hosting platforms confront significant challenges
  - A great area for ebusiness ventures!
  - But also a technically tough set of problems
  - And some can’t be solved…
- Overlay networks with security could offer a better story down the road