CMPSCI 677 Operating Systems **Spring 2014**

Lecture 8: February 19

Lecturer: Prashant Shenoy *Scribe: Ashish Jain*

8.1 Technical Challenges

8.1.1 Servers

Storage: We need lot of storage space which if very difficult to obtain and manage.

I/O: There is too much load on the server when many viewers are accessing the server. Real-time streaming is necessary; when we dont get data in real-time it becomes useless. A very small window of distortions is allowed. There is a need for continuous play out.

Computing in realtime: The challenges are not limited to Storage and I/O capabilities there is also a challenge on computing in realtime. The following tasks need to be achieved by the server.

Encryption We need encryption so that only a viewer with a key can view the picture.

Adaptation Since bandwidth is not guaranteed we need to change the quality of video to suite the network capabilities.

Transcoding The will be a need to Change for one format to other. Eg: Mobile devices use different format, Desktop uses different format.

8.1.2 Server Hierarchy

Prof. Zink talks regarding the cache functionality of distributed systems, and questions why the data should be placed as close to the viewers as possible. The answer being, latency issues, bandwidth consumption (If the data is closer to viewer the load on network reduces)

8.1.3 General OS Structure and Retrieval Data Path

The path for retrieving data is from filesystem (kernel) Application (User) communication system (Kernel), these frequent activities between Kernel and user space puts a lot of load onto the system.

8.1.4 Server Internals Challenges

There are data retrievals from disk and push to network for many users. He talks about various important resources such as memory, busses, CPU, storage system, communication system.
Prof gives an example to overcome these challenges, where the data packaged such that when computing the data, the systems need not to go to user space. (E.g.: scheduling, placement, caching/prefetching, admission control, merging concurrent users)

8.1.5 Timeliness Streaming

Explains the emphasis of retrieving data earlier, noted 3 points referring to the graph shown in the slides 1) Data must arrive before consumption time. 2) Data must be sent before arrival time. 3) Data must be read from disk before sending time.

8.2 Watch Global, Cache Local: YouTube network traffic at a Campus Network Measurements and Implications

8.2.1 Motivation

YouTube is different from traditional VoD, Access to YouTube from a campus network, Influence on content distribution paradigms? Correlation between global and local popularity?

Methodology used is to Monitor YouTube traffic at campus gateway, obtain global popularity, video clip traffic analysis, trace-driven simulation for various content distribution approaches. Trace driven means does not use stochastic approaches to generate simulation data, but used info from the data collected.

8.2.2 How YouTube works

Client makes a HTTP get message, which is responded by the YouTube Web server with a HTTP redirect message. Using the response the client sends an HTTP get message to the CDN server. Monitor box kept between the Client and the YouTube Servers at the campus gateway.

8.2.3 Monitoring YouTube Traffic

Monitor web server access where the Destination or source IP of YouTube web server pool are analyzed for HTTP GET and HTTP 303 See other messages. Monitoring Video Stream and monitor WWW access information to identify video stream to construct a flow with following details. 1) Duration of streaming session 2) Average data rate 3) Amount of transferred payload data

8.2.4 Measurement Results: Observations

1) No strong correlation between local and global popularity observed: 0.04 (Trace1), 0.06 (Trace2), 0.06 (Trace3). 2) Neither length of measurement nor number of clients observed seems to affect local popularity Distribution. 3) Video clips of local interest have a high local popularity. 4) No matter the lengths of the traces taken per video status is similar, there is a limit on caching.

8.2.5 Distribution infrastructures

Trace-driven simulation based on traces 1, 2, and 3. Create sequential list of requests. Make use of results from stream flow analysis. Simulation: Peer to peer used peer availability based on flow trace file information, Window-based availability approach, and Client availability influences hit rate. Simulation: Proxy caching used FIFO cache replacement. Effective low cost solution since storage in the order of 100 GB is required. Hit rates quite similar for all three traces compared to P2P results.

8.2.6 Conclusion

- No strong correlation between local and global popularity observed.
- Neither length of measurement nor number of clients observed seems to affect local popularity distribution.
- Video clips of local interest have high local popularity.
- Demonstrated implications of alternative distribution infrastructures.
- Client-based caching, P2P-based distribution, and proxy caching can reduce network traffic and allow faster access.

8.3 Watching User generated videos with Prefetching

8.3.1 Video prefetching scheme

Prefetching Agent (PA): Selects videos to be pre fetched and retrieve their prefixes, stores prefixes of pre fetched videos at clients (PF-Client) or proxy (PF-Proxy). Predict videos that are most likely to be watched, determines videos to prefetch from incoming requests.

- How to select videos to prefetch? PA predicts a set of videos to be requested. There are two main sources of video requests: 1. search results list. 2. Related videos list. PA uses the top N videos from these lists.
- How Often Users Click on Related Videos and Search Results? Determine the referrers of each video request in the traces from URL patterns, e.g., feature=related, feature=channel, From inference: look at a browse session to infer requests from Search Result list. Related Video lists and Search Results lists are the most frequently used referrers.
Evaluation Methodology: Issue the requests based on real user request traces, keep track of the videos in PAs storage. Evaluation metric- Hit ratio: How many requests we can serve from the PAs storage? Hit ratio = Hit requests/ All requests.
- Combining Caching with Prefetching: Cache-and-Prefetch can reach up to 81- Analyzing Hit Ratios: Only half of the hit requests come from RV lists, Requests from SR lists is a large portion of the hit requests especially in PF Proxy setting, Recommendation system is a good indicator of topic interest. RV lists overlap with the video requests generated from other sources (esp. in PF-Proxy) up to 70- Storage requirements: Require only 5 TB to reach 81- Impact of Storage space: Hit ratio decreases with the storage space size, Still can achieve hit ratio of around 60

8.3.2 Conclusion

Watching videos with prefix prefetching leads to following results. - Delay and Pauses are often - Prefix prefetching is feasible during browsing - Related videos are good interest predictors - Prefetching can reach hit ratio over 81

8.4 Cache-centric Video Recommendation: An Approach to Improve the Efficiency of YouTube Caches

8.4.1 Motivation

YouTube is most popular user generated video service. Billions of videos with unequal popularity lead to long tail. Effective caching is difficult with such a long tail. Users usually select next video from related list.

Caching and Prefetching of related list have shown to be effective.

8.4.2 Approach

Reordering of related list based on the content in cache. To verify the feasibility of reordering, we perform chain analysis. We also perform the RTT analysis to understand the origin of videos.
Reordering Approaches
- Content centric reordering- related list selection based on content, Position might change based on reordering.
- Position centric reordering- Related list selection based on position of original list, Content might change based on reordering.

8.4.3 Conclusion

- We take advantage of user behavior of watching videos from related list.
- Our approach is to reorder the related list to move the content in the cache to top of the list.
- We present two approaches to reordering selection: Position centric and Content centric.
- Position centric selection leads to a high cache hit rate and reduction in server load due to reordering.