Multimedia Streaming

Mike Zink

Technical Challenges

- **Servers (and proxy caches)**
  - **storage**
    - continuous media streams, e.g.:
      - 4000 movies * 90 minutes * 10 Mbps (DVD) = 27.0 TB

        15 Mbps = 40.5 TB

        36 Mbps (BluRay) = 97.2 TB
      - 2000 CDs * 74 minutes * 1.4 Mbps = 1.4 TB

Technical Challenges

- Servers (and proxy caches)
  - I/O
    - many concurrent clients
    - real-time retrieval
    - continuous playout
      - DVD (~4Mbps, max 10.08Mbps)
      - HDTV (~15Mbps, BlueRay ~36Mbps)
    - current examples of capabilities
      - disks:
        - » mechanical: e.g., Seagate X15 - ~400 Mbps
        - » SSD: e.g., MTRON Pro 7000 – ~1.2 Gbps
      - network: Gb Ethernet (1 and 10 Gbps)
      - bus(ses):
        - » PCI 64-bit, 133Mhz (8 Gbps)
        - » PCI-Express (2 Gbps each direction/lane, 32x = 64 Gbps)
  - computing in real-time
    - encryption
    - adaptation
    - transcoding

Outline

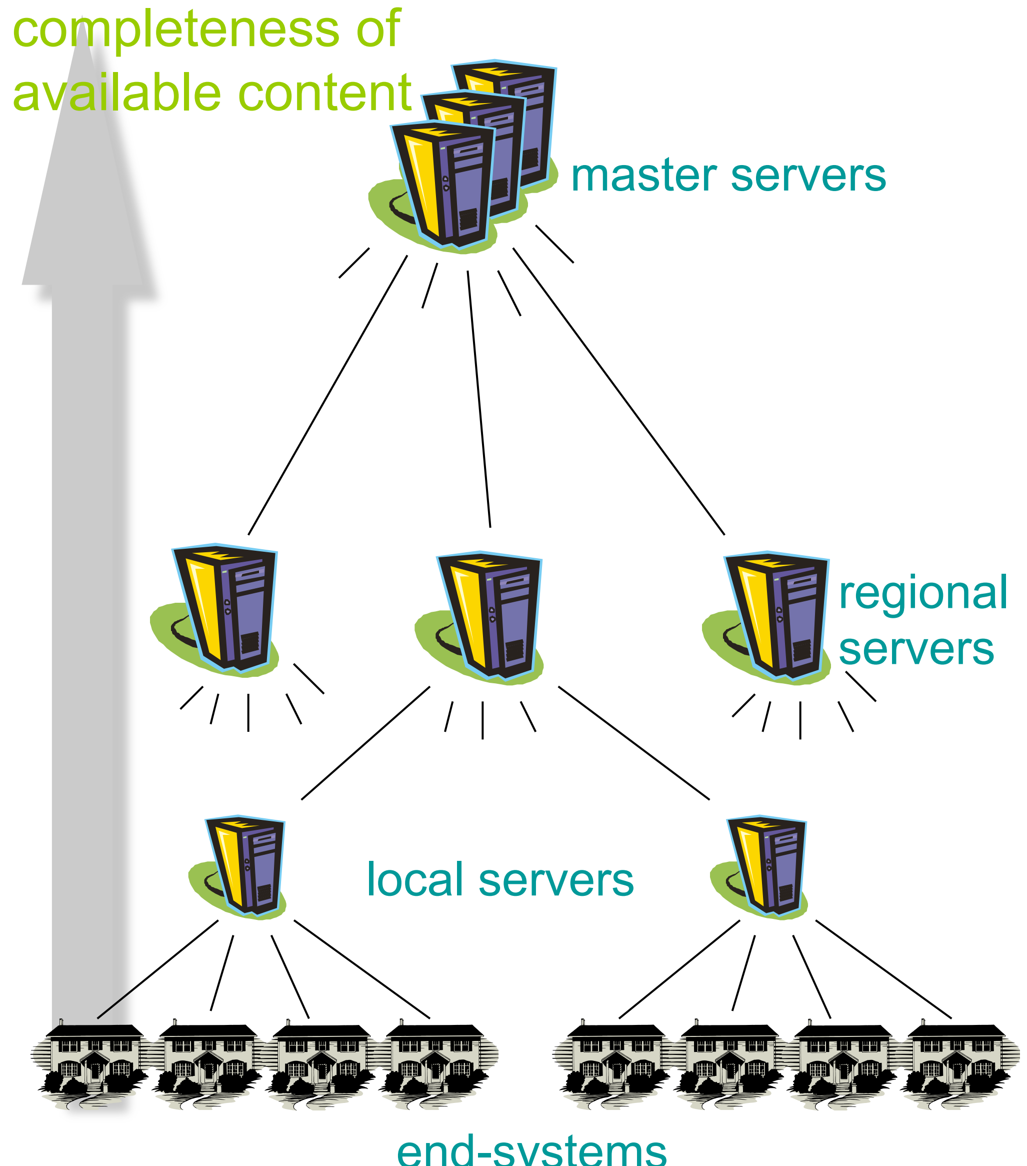
- Multimedia Servers
- Analysis of the YouTube streaming system
- Improving performance
  - Caching
  - Prefetching
  - Recommendation systems

Server Hierarchy

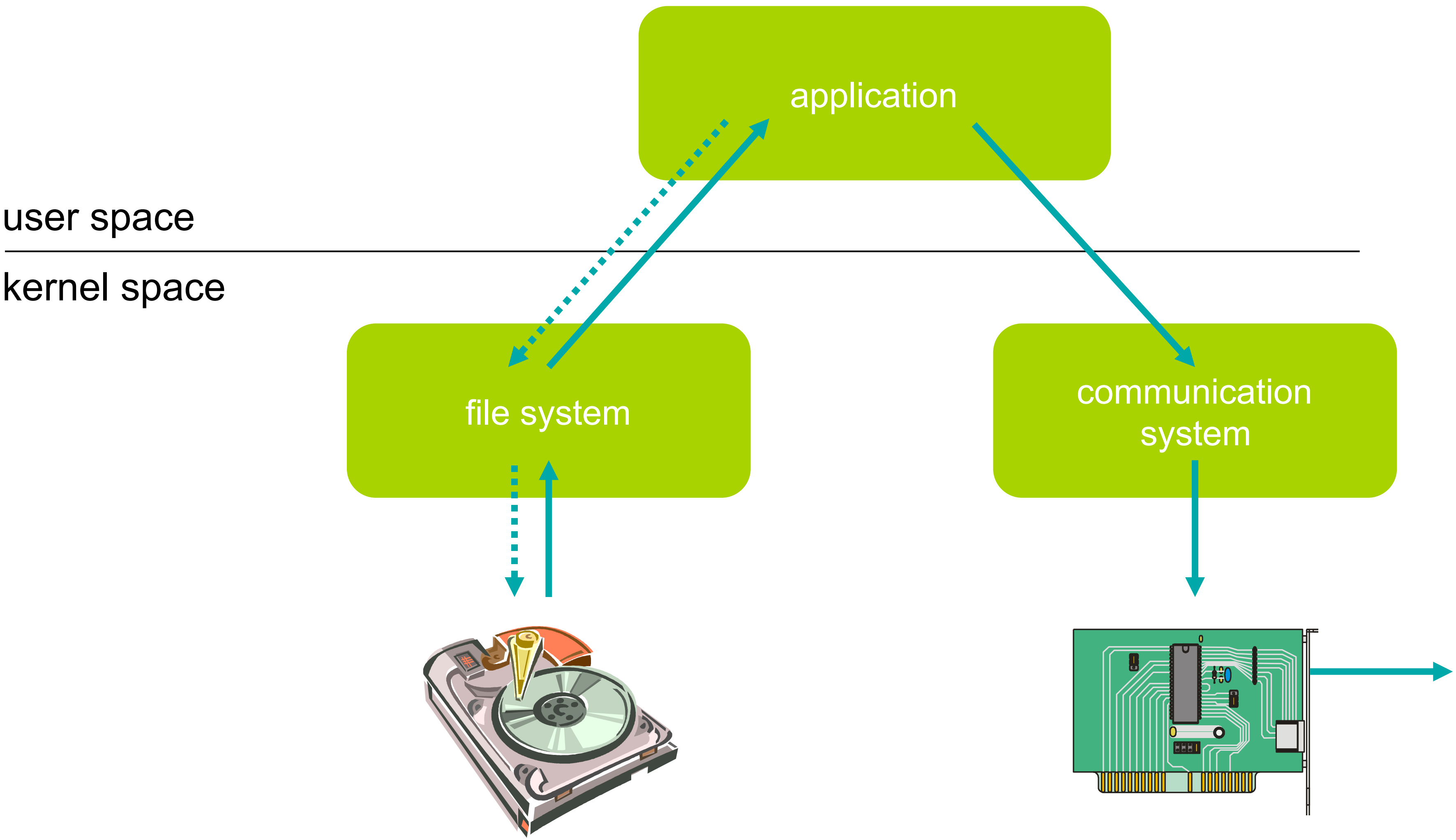
- Intermediate nodes or proxy servers may offload the main master server

- Popularity of data:
  not all are equally popular – most request directed to only a few

- Straight forward hierarchy:
  - popular data replicated and kept close to clients
  - locality vs. communication vs. node costs



General OS Structure and Retrieval Data Path



Server Internals Challenges

- *Data retrieval from disk and push to network for many users*

- Important resources:
  - memory
  - busses
  - CPU
  - storage (disk) system
  - communication (NIC) system

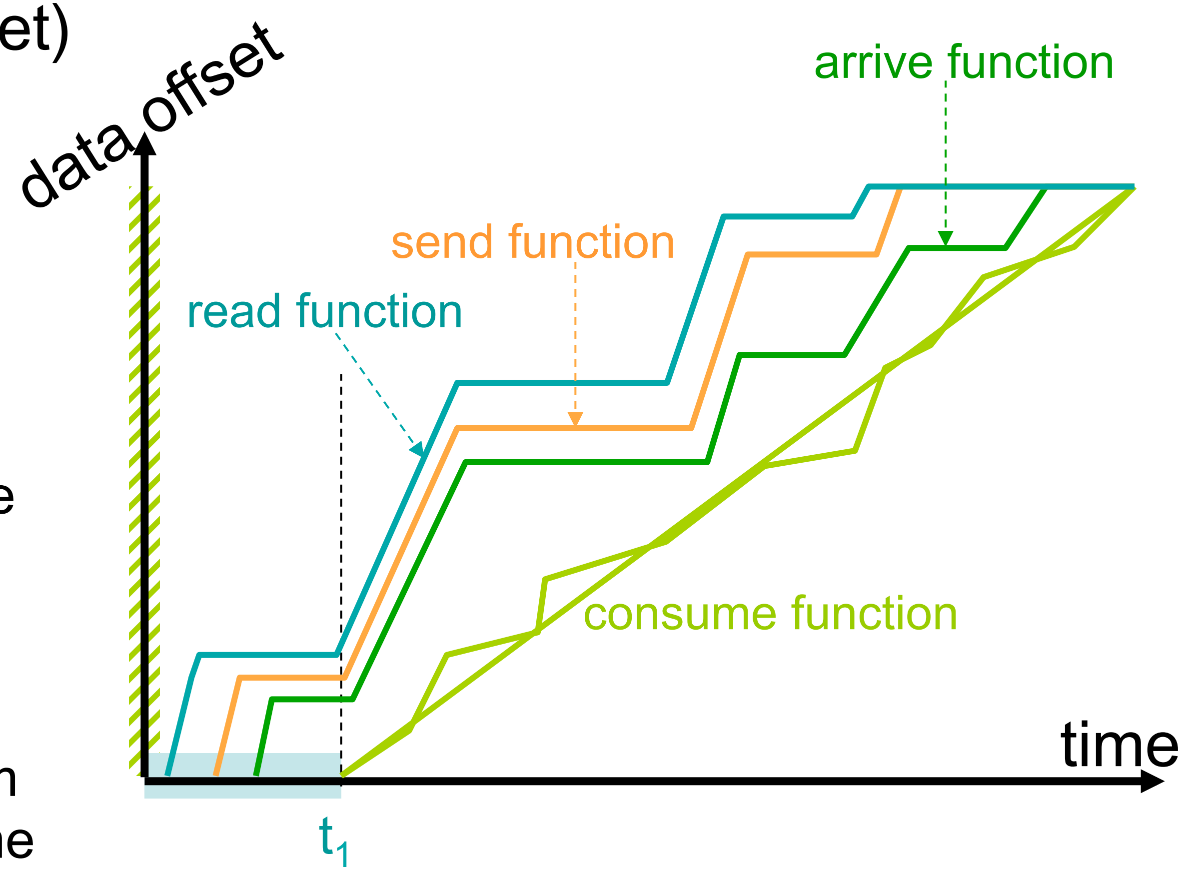
- Much can be done to **optimize resource utilization**, e.g., scheduling, placement, caching/prefetching, admission control, merging concurrent users, …

Timeliness: Streaming

- Start presenting data (e.g., video playout) at t_1

- Consumed bytes (offset)
  - variable rate
  - constant rate

- Must start retrieving data earlier
  - Data must arrive before consumption time
  - Data must be sent before arrival time
  - Data must be read from disk before sending time



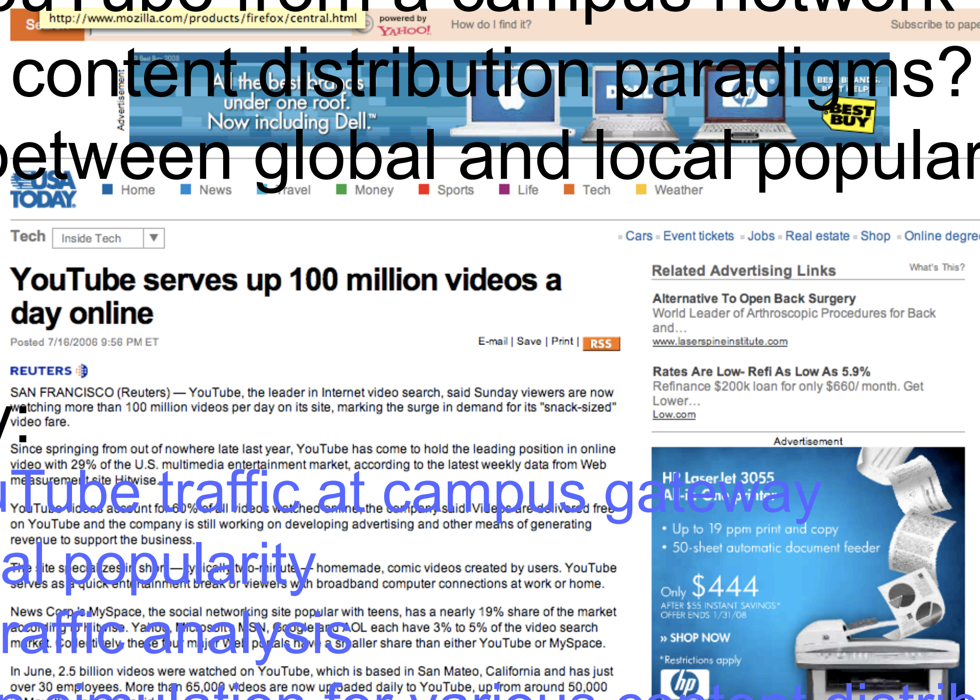
Watch Global, Cache Local: YouTube Network Traffic at a Campus Network – Measurements and Implications

Overview

- Motivation
- Measurement
  - How YouTube Works
  - Monitoring YouTube Traffic
  - Measurement Results
- Distribution Infrastructures
  - Peer-to-Peer
  - Proxy Caching
- Conclusions & Future Work

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?



YouTube serves up 100 million videos a day online

Posted 7/16/2006 9:56 PM ET

E-mail | Save | Print | RSS

REUTERS

SAN FRANCISCO (Reuters) — YouTube, the leader in Internet video search, said Sunday viewers are now watching more than 100 million videos per day on its site, marking the surge in demand for its "snack-sized" video fare.

Since springing from out of nowhere late last year, YouTube has come to hold the leading position in online video with 29% of the U.S. multimedia entertainment market, according to the latest weekly data from Web measurement site Hitwise.

YouTube videos account for 60% of all videos watched online, the company said. Videos are delivered free on YouTube and the company is still working on developing advertising and other means of generating revenue to support the business.

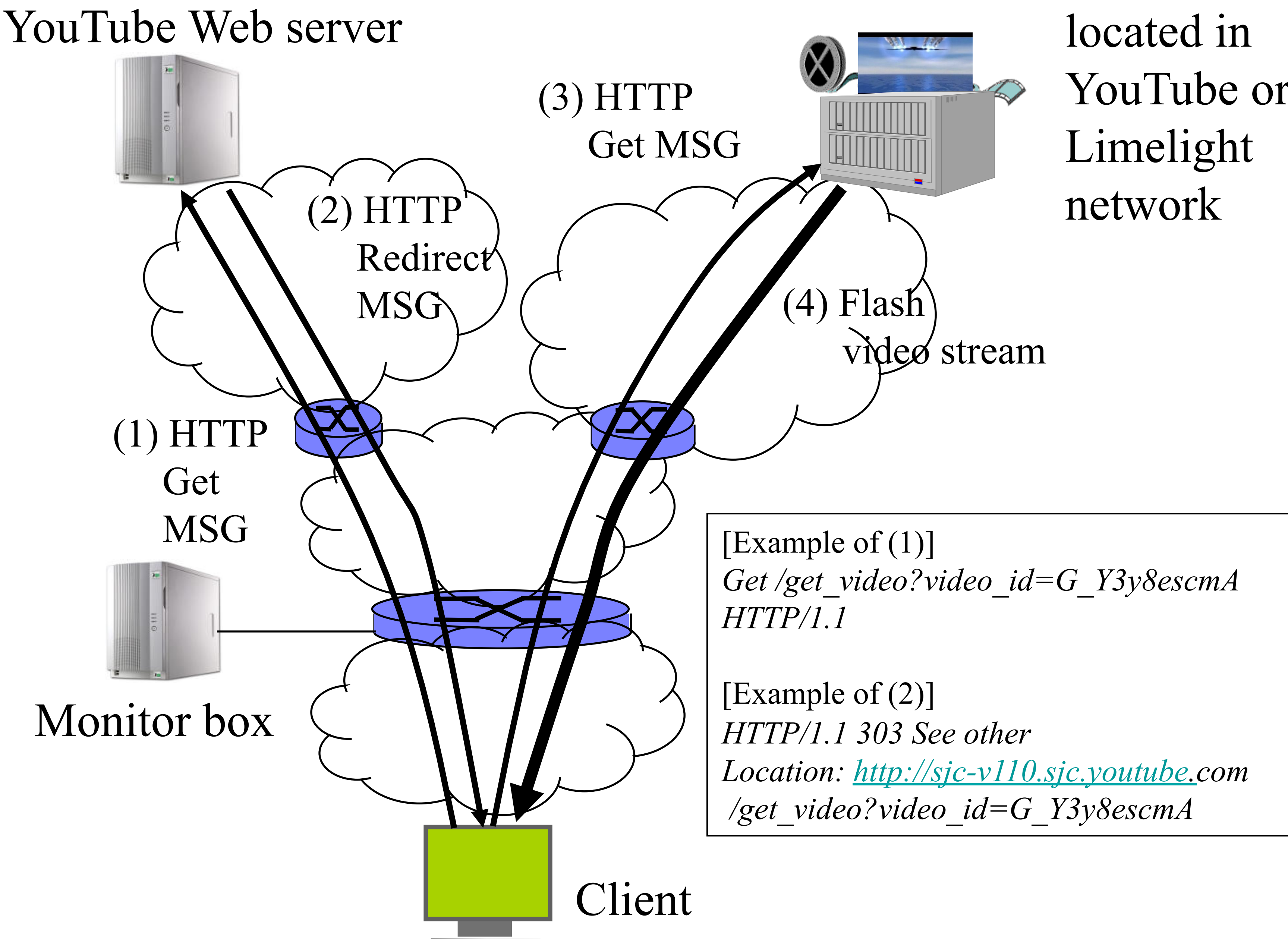
The site specializes in short — typically two-minute — homemade, comic videos created by users. YouTube serves as a quick entertainment break for viewers with broadband computer connections at work or home.

News Corp.'s MySpace, the social networking site popular with teens, has a nearly 19% share of the market according to Hitwise. Yahoo, Microsoft MSN, Google and AOL each have 3% to 5% of the video search market. Collectively, these four major Web portals have a smaller share than either YouTube or MySpace.

In June, 2.5 billion videos were watched on YouTube, which is based in San Mateo, California and has just over 30 employees. More than 65,000 videos are now uploaded daily to YouTube, up from around 50,000 in May, the company said.

- Methodology:
  - Monitor YouTube traffic at campus gateway
  - Obtain global popularity
  - Video Clip traffic analysis
  - Trace-driven simulation for various content distribution approaches

How YouTube Works!



[Example of (1)]
Get /get_video?video_id=G_Y3y8escmA HTTP/1.1

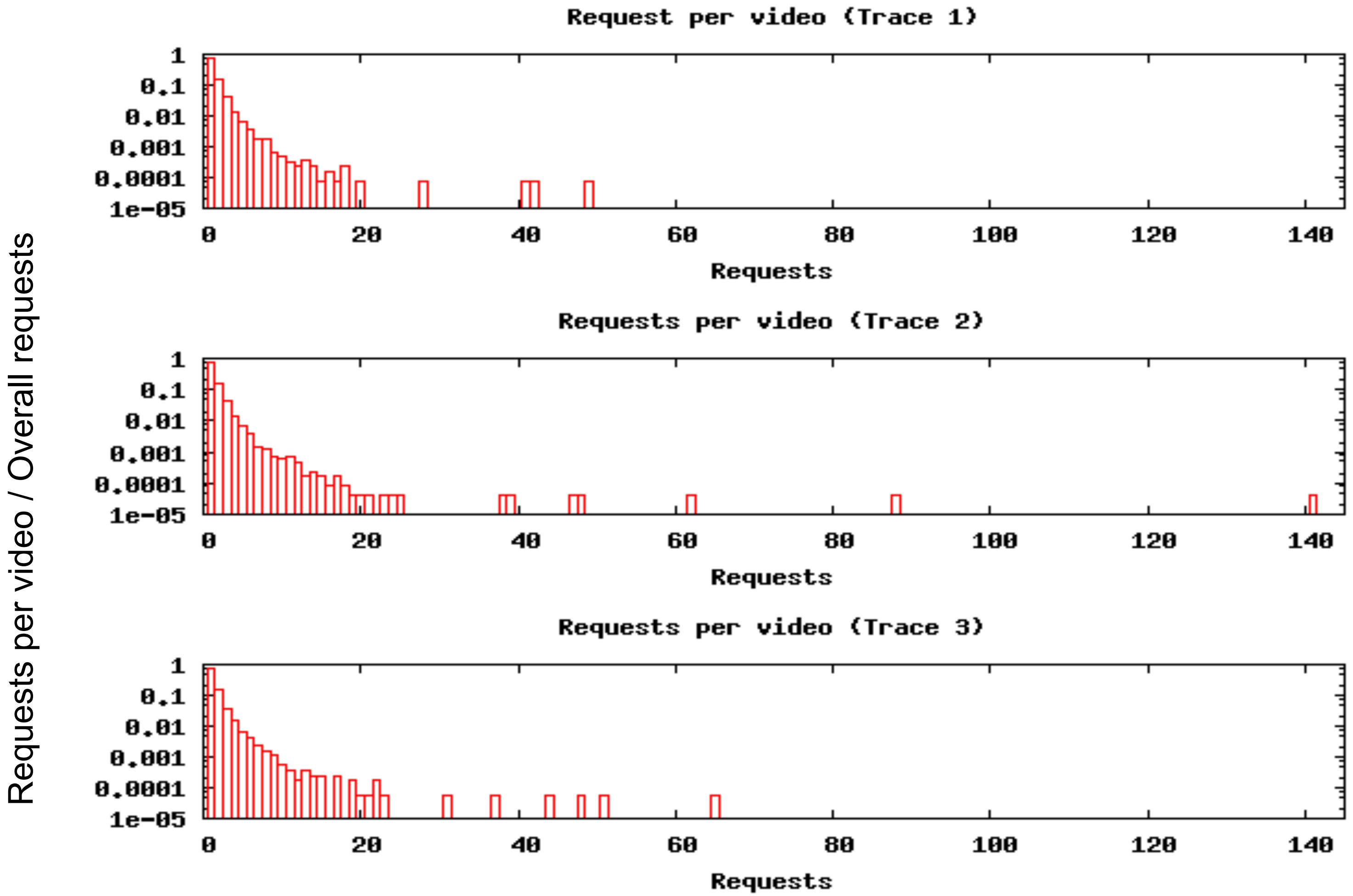
[Example of (2)]
HTTP/1.1 303 See other
Location: http://sjc-v110.sjc.youtube.com /get_video?video_id=G_Y3y8escmA

Monitoring YouTube Traffic

- Monitor web server access
  - Destination or source IP of YouTube web server pool
  - Analyze HTTP GET and HTTP 303 See Other messages
- Monitoring Video Stream
  - WWW access information to identify video stream
  - Construct flow to obtain:
    - Duration of streaming session
    - Average data rate
    - Amount of transferred payload data

Trace	Date	Length (Hours)	# of Unique Clients	Per Video Stats		
				Total	Single	Multi
1	05/08- 05/09	12	2127	12955	77%	23%
2	05/22-05/25	72	2480	23515	77%	23%
3	06/03-06/07	108	1547	17183	77%	23%

Measurement Results: Video Popularity



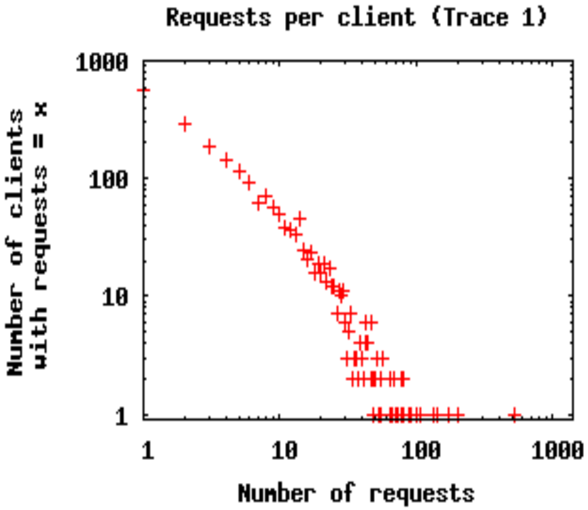
Measurement Results: Observations

- No strong correlation between local and global popularity observed: 0.04 (Trace1), 0.06 (Trace2), 0.06 (Trace3)
- Neither length of measurement nor # of clients observed seems to affect local popularity distribution
- Video clips of local interest have a high local popularity

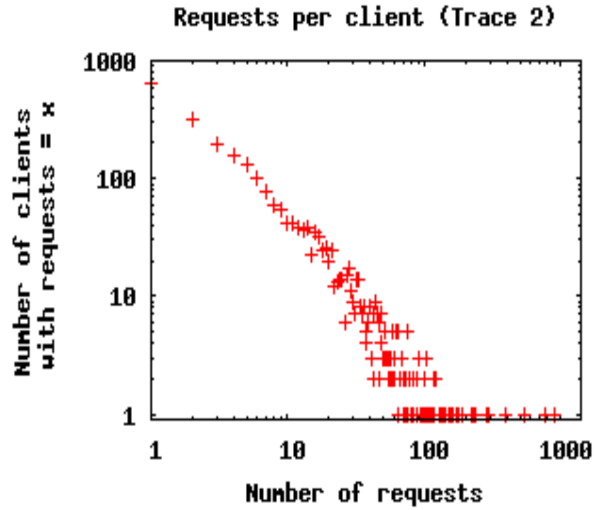
http://www.youtube.com/watch?v=dp4MYii7MqA

Measurement Results: Requests per Client

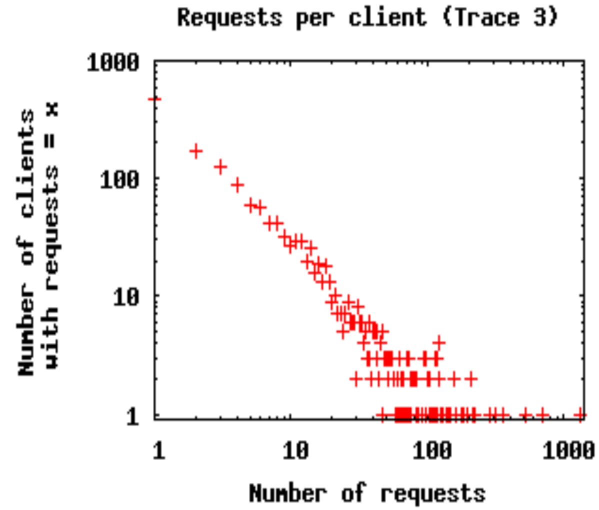
Client in here means IP address (NAT, DHCP)



(a)



(b)



(c)

Trace	Video clips with multiple requests from same client	Total number of requests	Max. number of requests per client
1	2149	3100	17
2	3899	5869	25
3	3170	4893	47

Overview

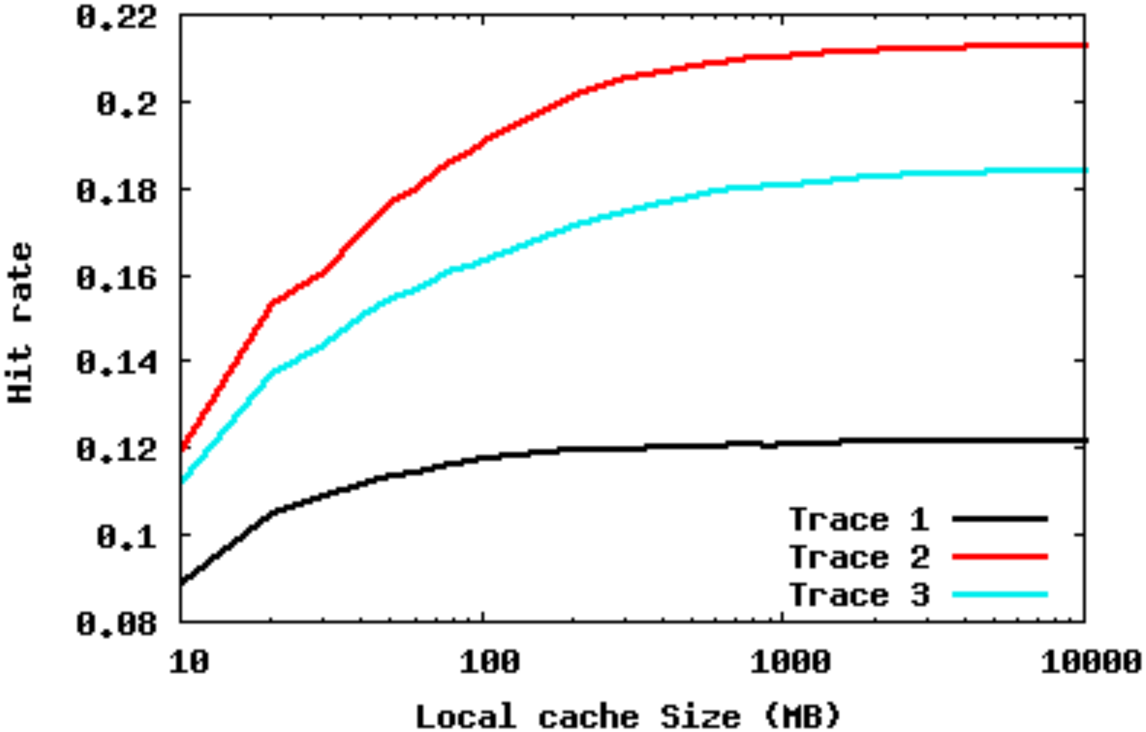
- Motivation
- Measurement
  - How YouTube Works
  - Monitoring YouTube Traffic
  - Measurement Results
- Distribution Infrastructures
  - Peer-to-Peer
  - Proxy Caching
- Conclusions & Future Work

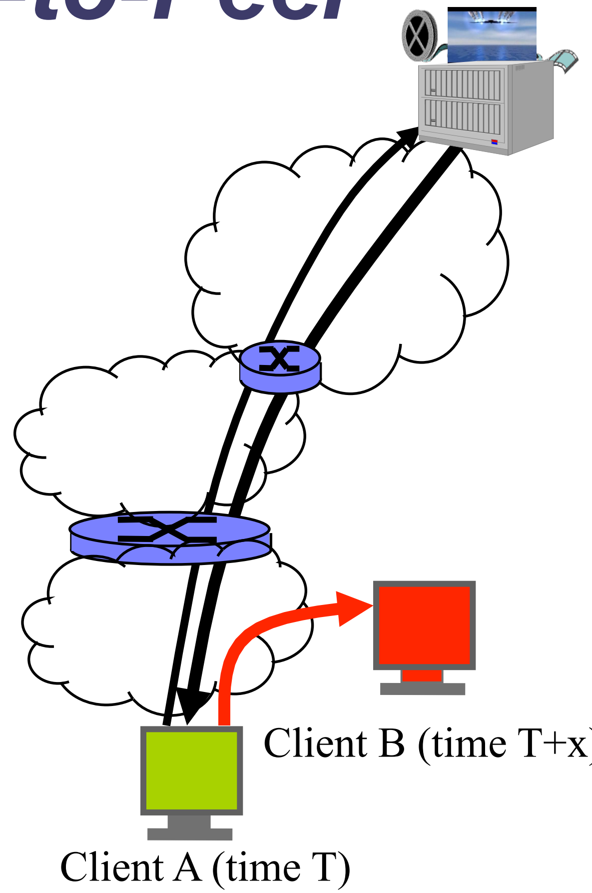
Distribution Infrastructures

- Trace-driven simulation based on traces 1, 2, and 3
- Create sequential list of requests
- Make use of results from stream flow analysis

Trace	Duration (sec) (Length of viewing)			Packets			Payload Size (bytes)			Rate (Kbps)		
	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min
1	99.62	4421.00	0.04	5202	149098	2	7.5×10^6	2.15×10^8	484	632	5450	0.54
2	95.81	2359.83	0.53	4478	89350	76	6.4×10^6	1.30×10^8	95760	646	8633	6.74
3	81.34	16956.28	0.04	4431	97452	2	6.3×10^6	1.42×10^8	452	908	10582	0.19

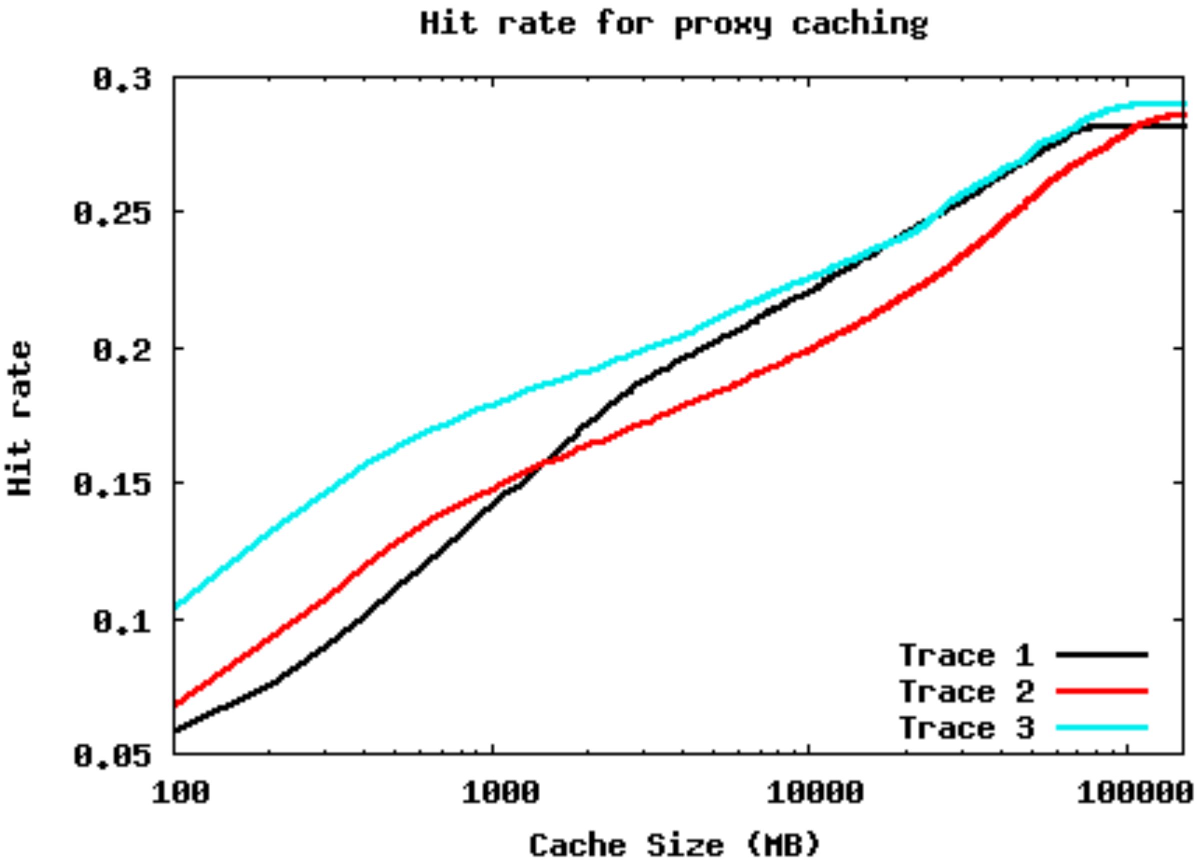
Simulation: Peer-to-Peer

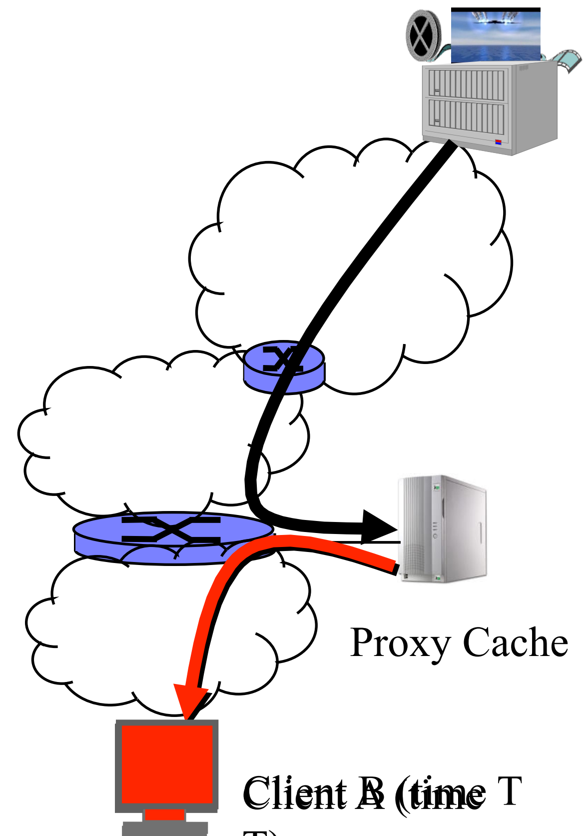




- Peer availability based on flow trace file information
- Window-based availability approach
- Client availability influences hit rate

Simulation: Proxy Caching





- 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

Related Work

Parallel work to ours:

- Cha et al. (IMC 2007):
  - Only information from YouTube server is analyzed
  - No information about benefits of using caching in access networks
- Gill et al. (IMC 2007):
  - Similar motivation to ours
  - Only predefined set of content servers could be monitored
  - General trend between their and our results observable

No simulative study on different distribution architectures

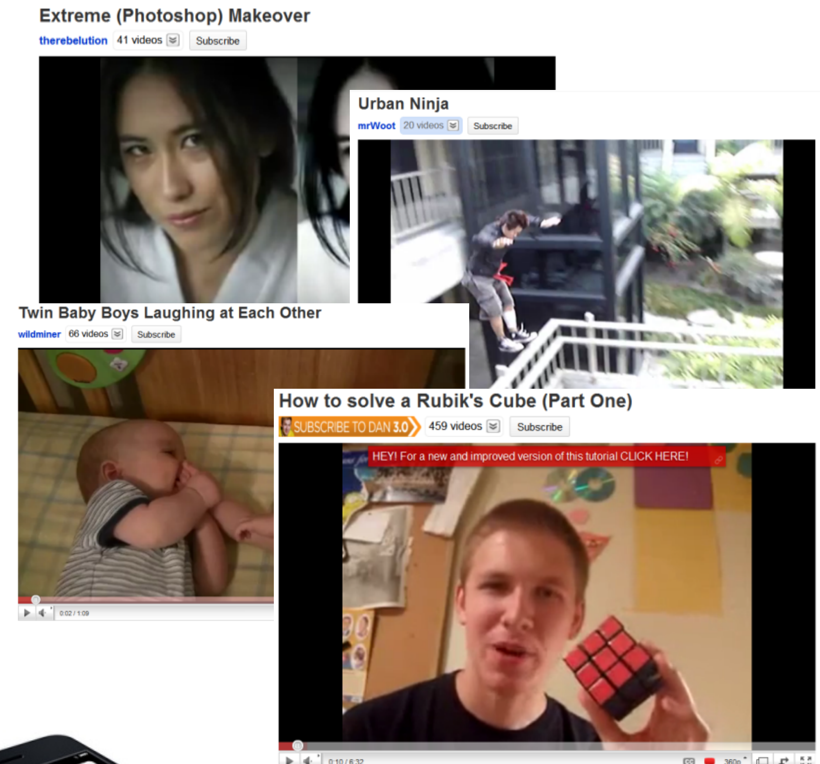
Conclusions

- No strong correlation between local and global popularity observed
- Neither length of measurement nor # 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

Watching User Generated Videos with Prefetching

User Generated Videos

- Professional Produced Videos
  - Netflix
  - Hulu
- User Generated Videos
  - YouTube, Youku, Tudou
  - Hundreds of millions of short video clips
  - Wide ranges of topics
- Growing user generated videos
  - Readily available device
  - Production cycle is short



Motivation

- User experience in watching videos is not satisfactory
  - Slow startup time
  - Many pauses during playback

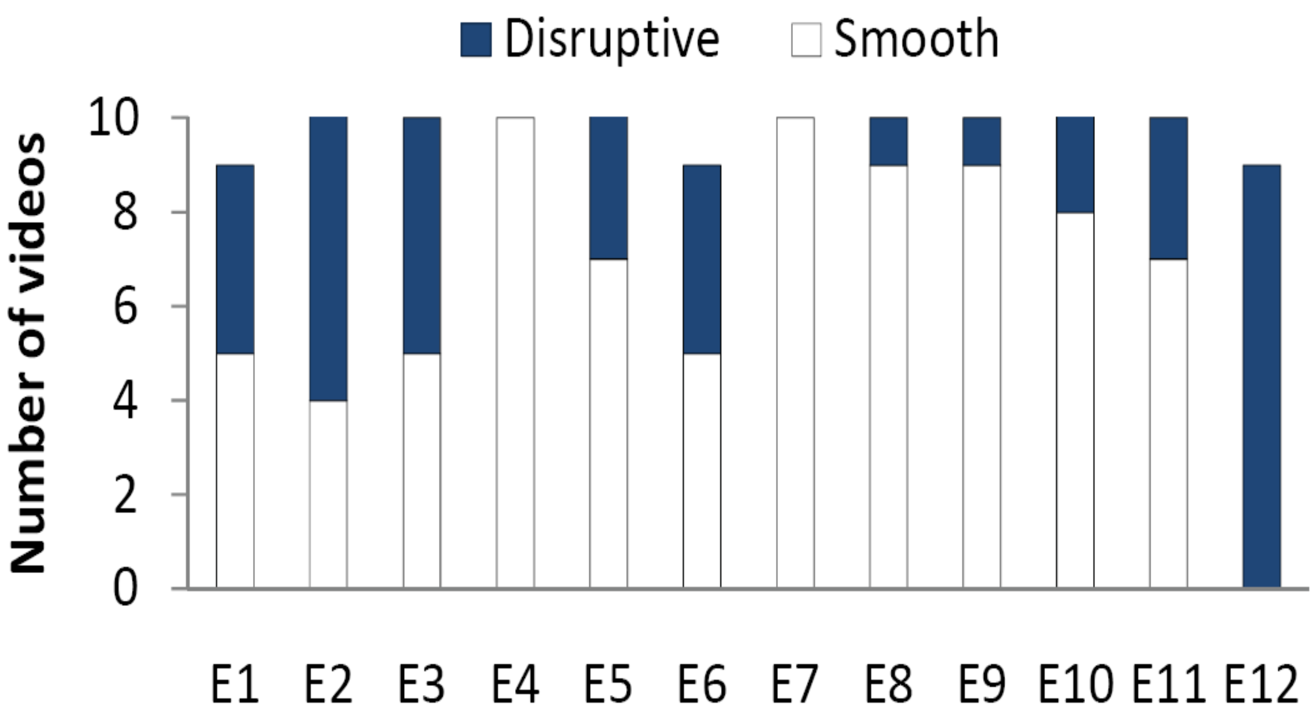
Measuring User Experiences Watching YouTube

V*ideo download traces* from various environments

Environment	Location	Network Technology
E1	University 1	Campus WLAN
E2	Company 1	DSL
E3	Home 1	DSL
E4	Apartment 1	Cable Internet
E5	Dormitory 1	Campus LAN
E6	Dormitory 2	Campus LAN
E7	Apartment 2	Cable Internet
E8	Town Library	Wireless Network
E9	Coffee shop	Wireless Network
E10	University 2	Campus WLAN
E11	Home 2	DSL
E12	Hotel	Wireless Network

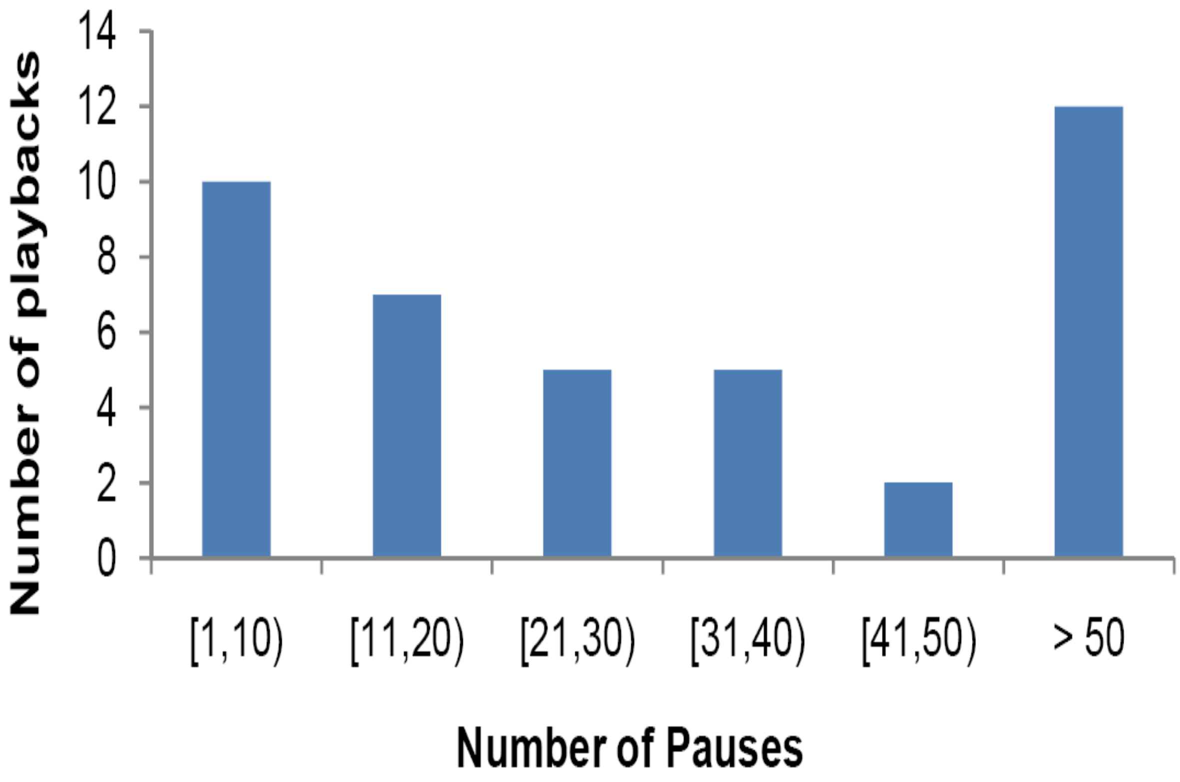
Likelihood of Experiencing Pauses

- 10 out of 12 environments contain playbacks with pauses
- 41 out of 117 playbacks (35%) contain pauses



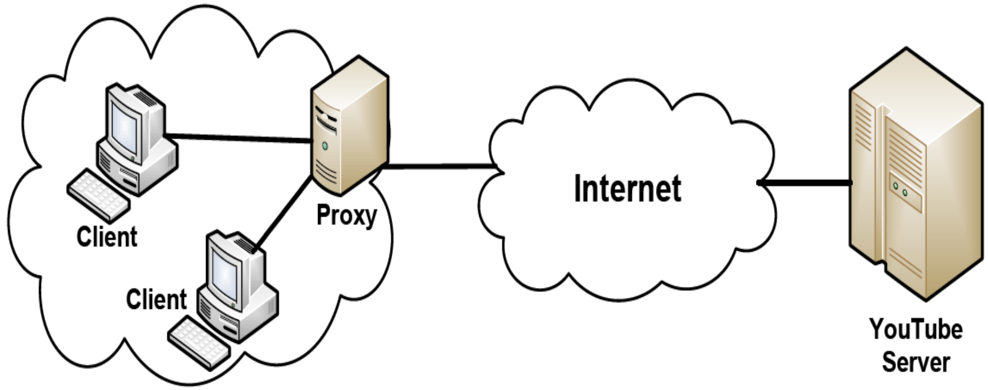
Number of Pauses

- 31 out of 117 playouts (22.6%) contain more than 10 pauses



How to improve user experiences?

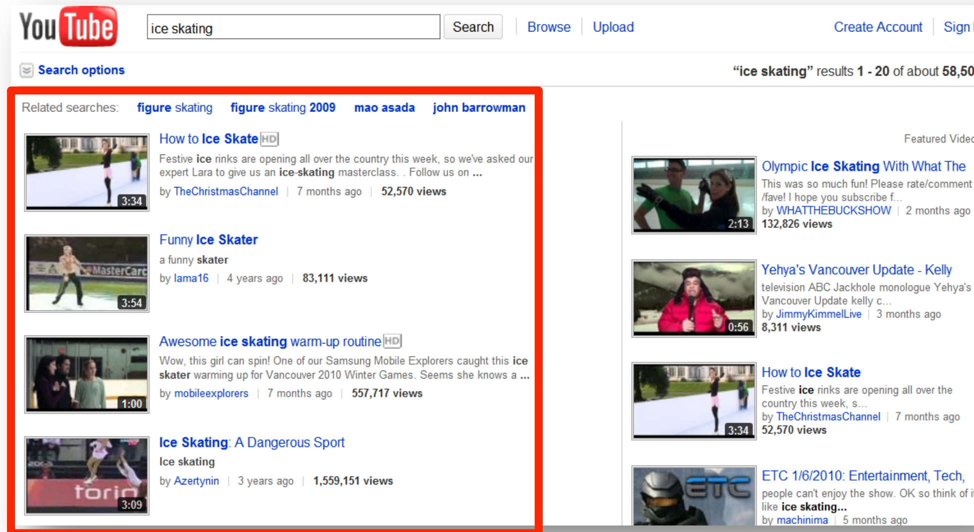
Video Prefetching Scheme

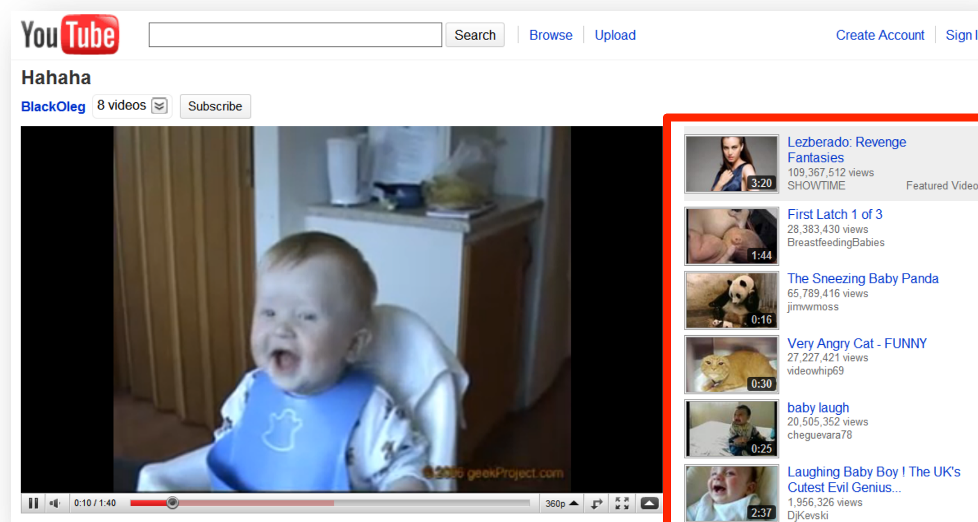


- Prefetching Agent (PA)
  - Select videos to be prefetched and retrieve their prefixes
  - Store prefixes of prefetched videos
  - At clients (PF-Client) or proxy (PF-Proxy)
- Predict videos that are most likely to be watched
  - PA determines videos to prefetch from incoming requests

How to select videos to prefetch?

- PA predicts a set of videos to be requested
- Two main sources of video requests
  - Search Result lists
  - Related Video lists
- Use top N videos from these lists
- Advantages
  - Simple
  - Require no additional data
  - Effectiveness?





Datasets for Evaluation

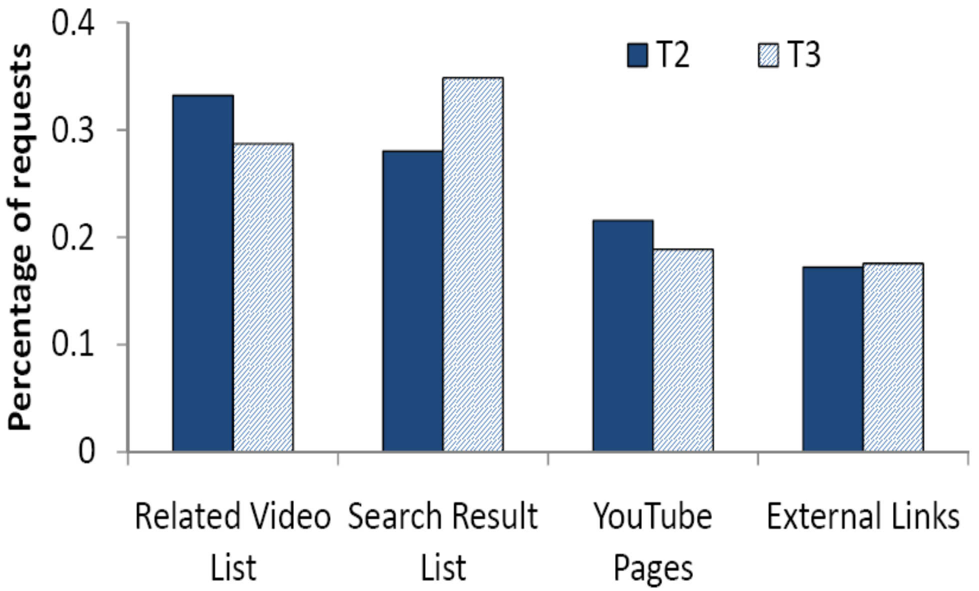
- Traces of data traffic between a campus network and YouTube servers

Trace File	T1	T2	T3
Duration	1 day	3 days	7 days
Start Date	20-Oct-09	8-Jan-10	28-Jan-10
# Request	71,282	7,562	257,098
# Unique Clients	7,914	607	10,511
# Unique Videos	48,978	5,887	154,363

- Retrieve Search Result lists and Related video lists via YouTube data API

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

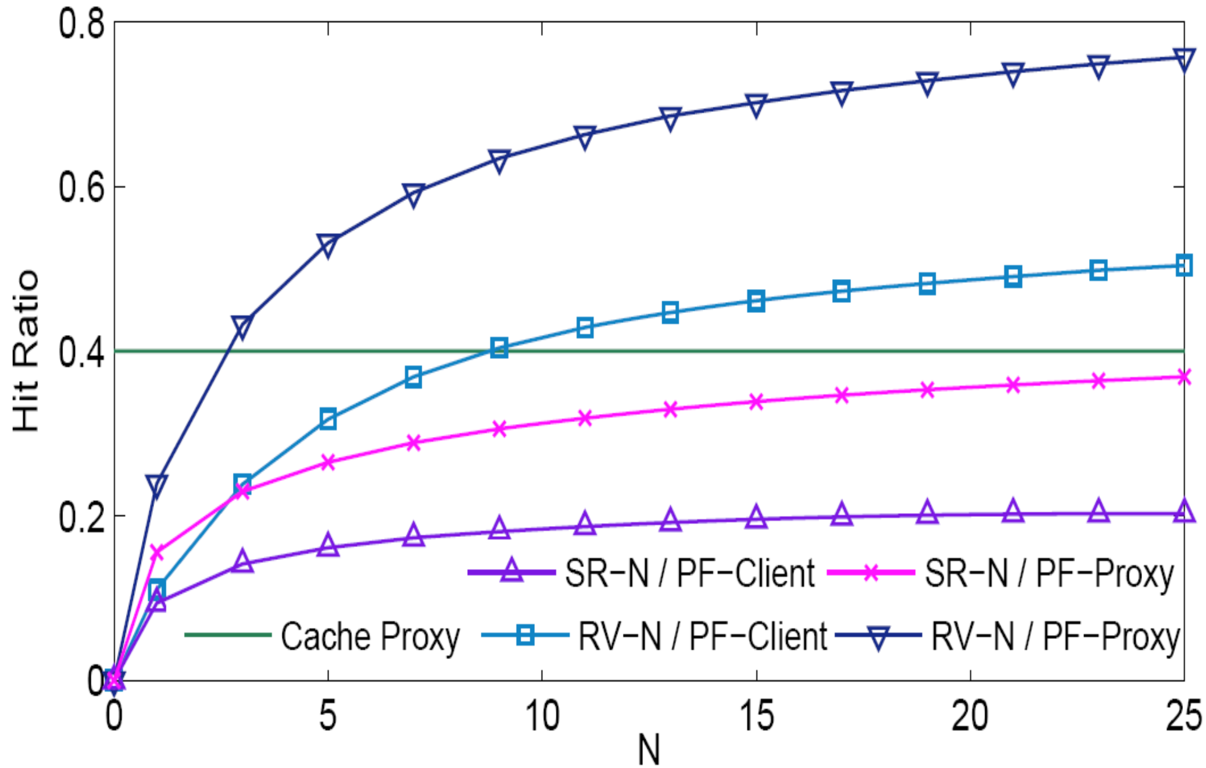


Evaluation Methodology

- Issue the requests based on real user request traces
- Keep track of the videos in PA's storage
- Evaluation metric
  - Hit ratio: How many requests we can serve from the PA's storage?

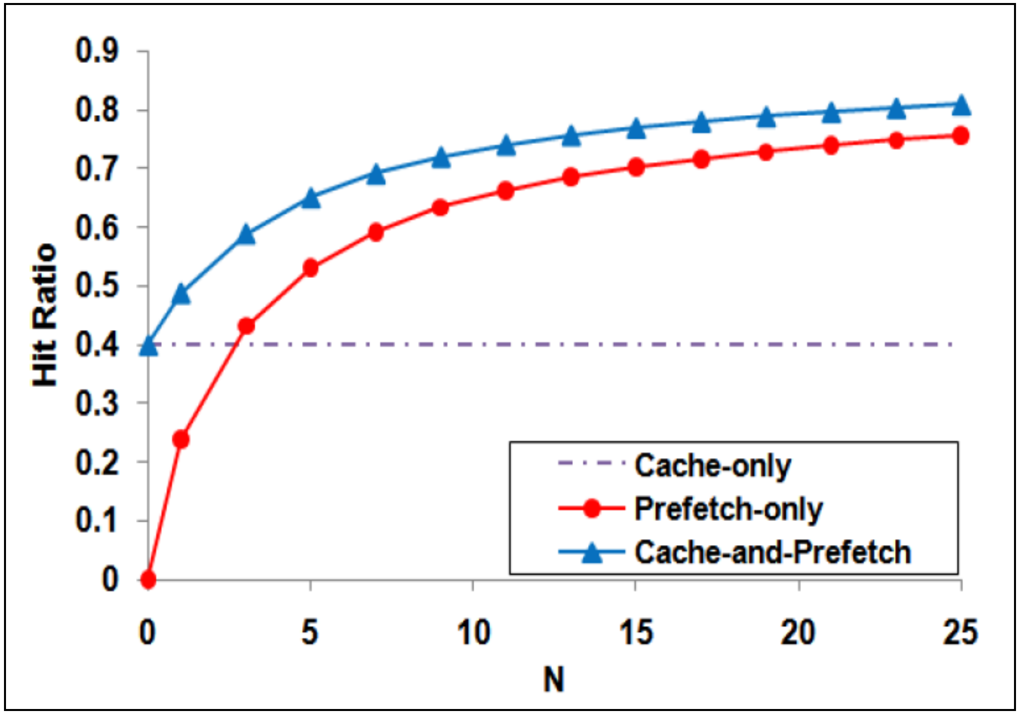
$$\text{Hit ratio} = \frac{\text{Hit requests}}{\text{All requests}}$$

Effectiveness of various scheme combinations



- Videos from a Related Video list of a user are watched by other users
- Best combination is using RV-N algorithm with PF-Proxy setting

Combining Caching with Prefetching

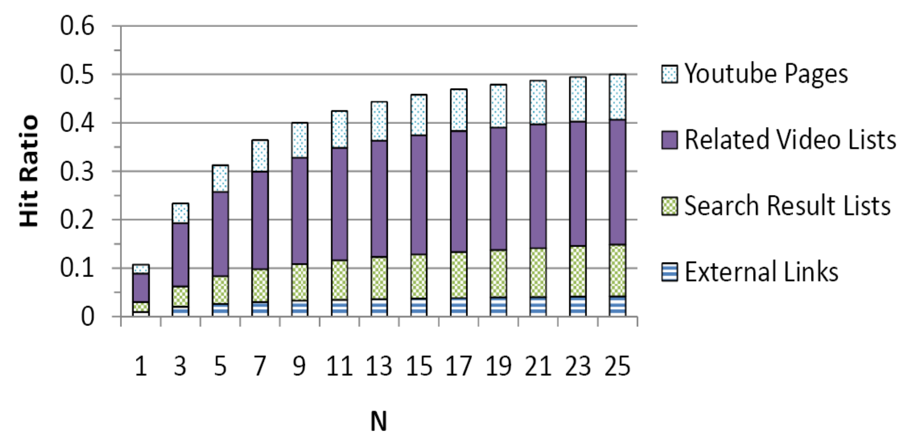


- Cache-and-Prefetch can reach up to 81% of hit ratio
- Improvement is smaller as N increases due to larger overlapping between prefetched videos and cached videos

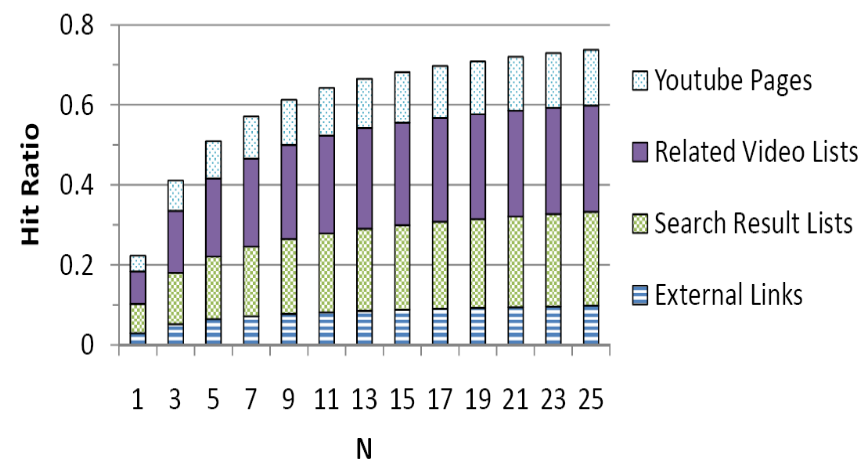
Analyzing Hit Ratios

- 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

PF-Client



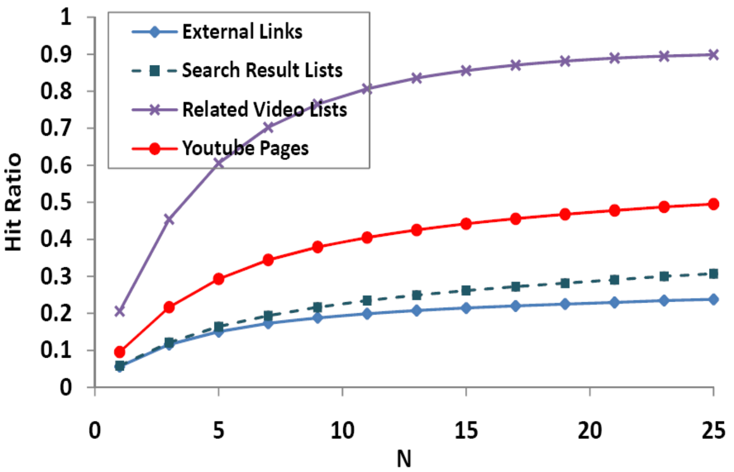
PF-Proxy



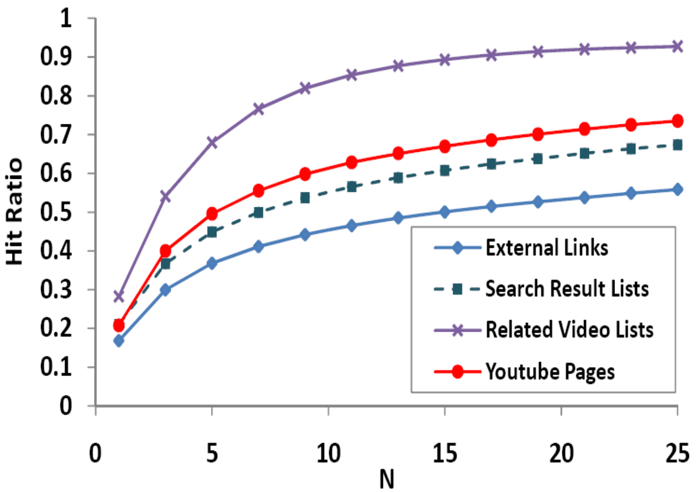
Analyzing the High Hit Ratios

- RV lists overlap with the video requests generated from other sources (esp. in PF-Proxy) up to 70%

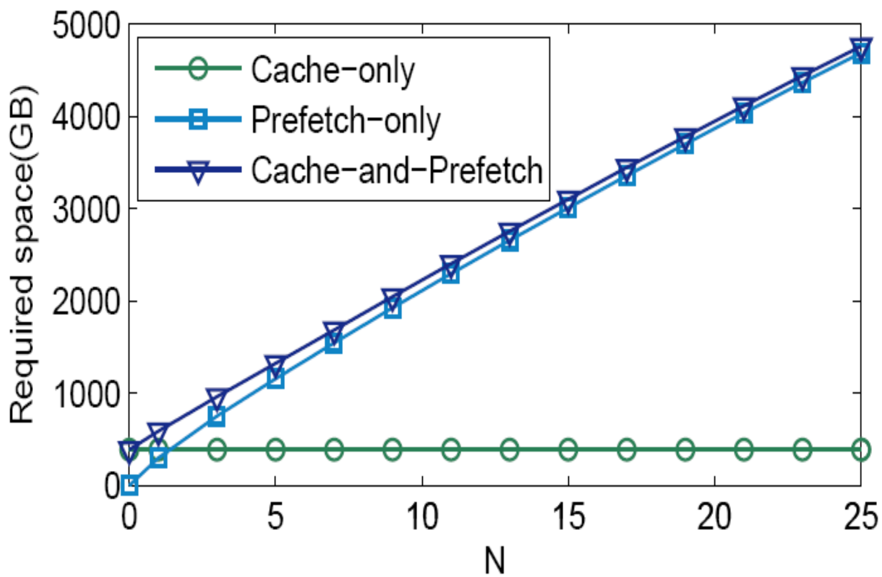
PF-Client



PF-Proxy

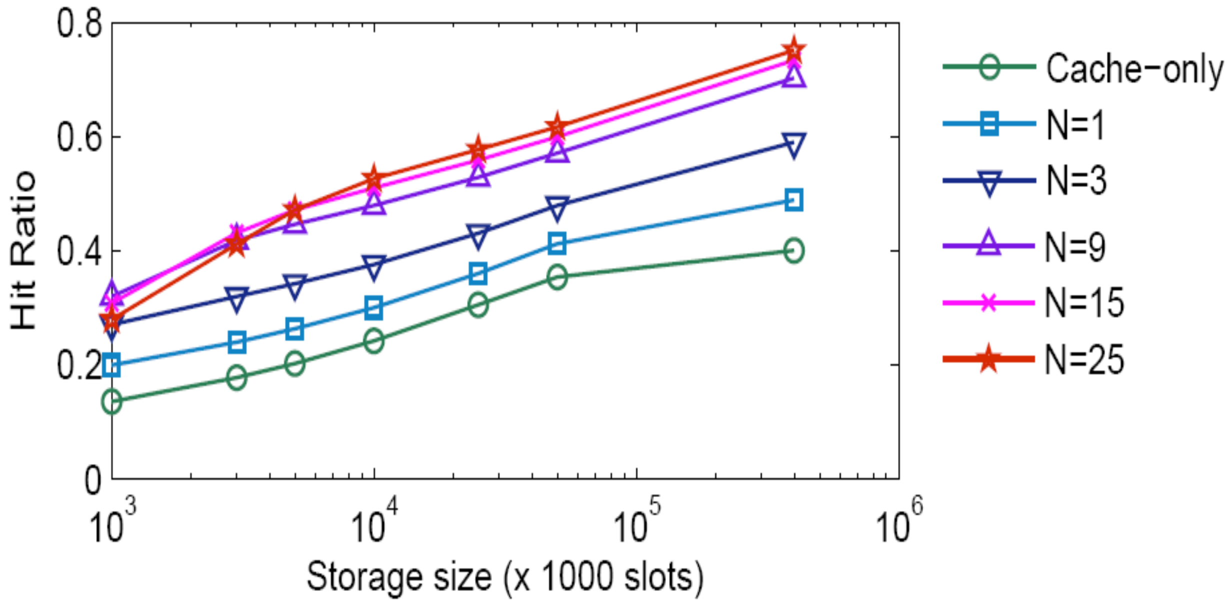


Storage Requirement



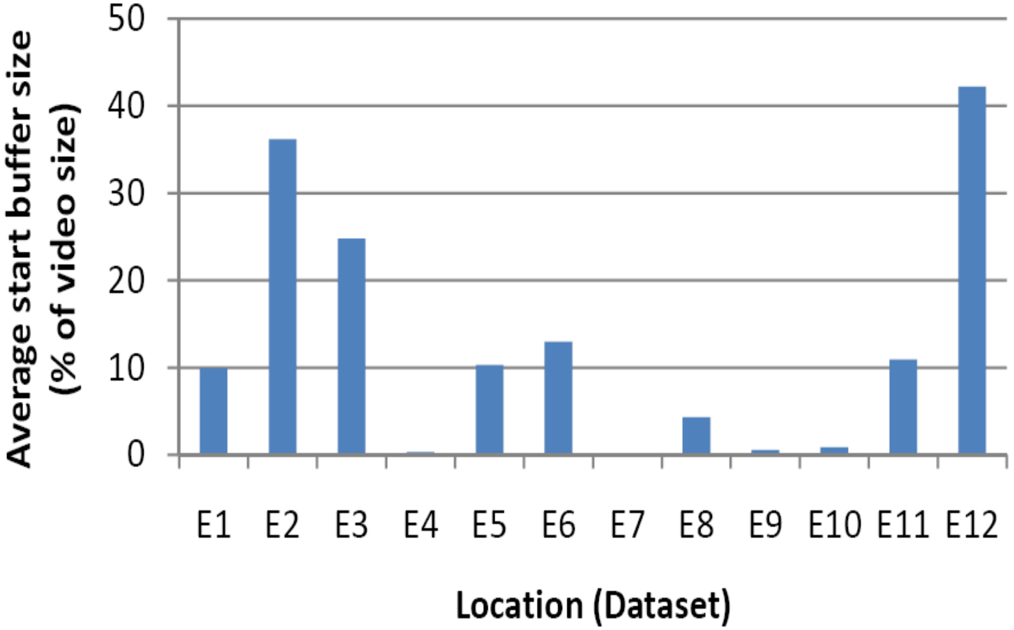
- Measured in slots – a slot holds one prefix of a video
- One slot = 2.5 MB (for prefix size of 30% and average video size of 8.4 MB)
- Require only 5 TB to reach 81% of hit ratio (at N=25)

Impact of Storage space



- Hit ratio decreases with the storage space size
- Still can achieve hit ratio of around 60% with 125 GB (50k slots)
- Compared to caching, cache-and-prefetch always performs better

Do we need to prefetch the whole video?



- Prefetching the whole videos is not necessary
- From analysis of video download traces, each location and each video requires different prefix size

Feasibility – Traffic Overhead

- Suppose prefix size = 15%, N = 11 and caching whole videos

Scheme	Hit Ratio	Normalized load
No scheme	0%	1.00
Cache-only	40%	0.60
Prefetch-only	66%	1.44
Cache-and-Prefetch	74%	1.02

- Caching helps reduce the traffic
- Pure prefetching yields higher hit ratio while increase traffic by 44%
- Combining the two results in highest hit ratio and only introduce 2% additional traffic

Conclusion

- Watching videos with prefix prefetching
  - Delay and Pauses are often
  - Prefix prefetching is feasible during browsing
  - Related videos are good interest predictors
  - Prefetching can reach hit ratio over 81% while caching can reach hit ratio of 40%

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

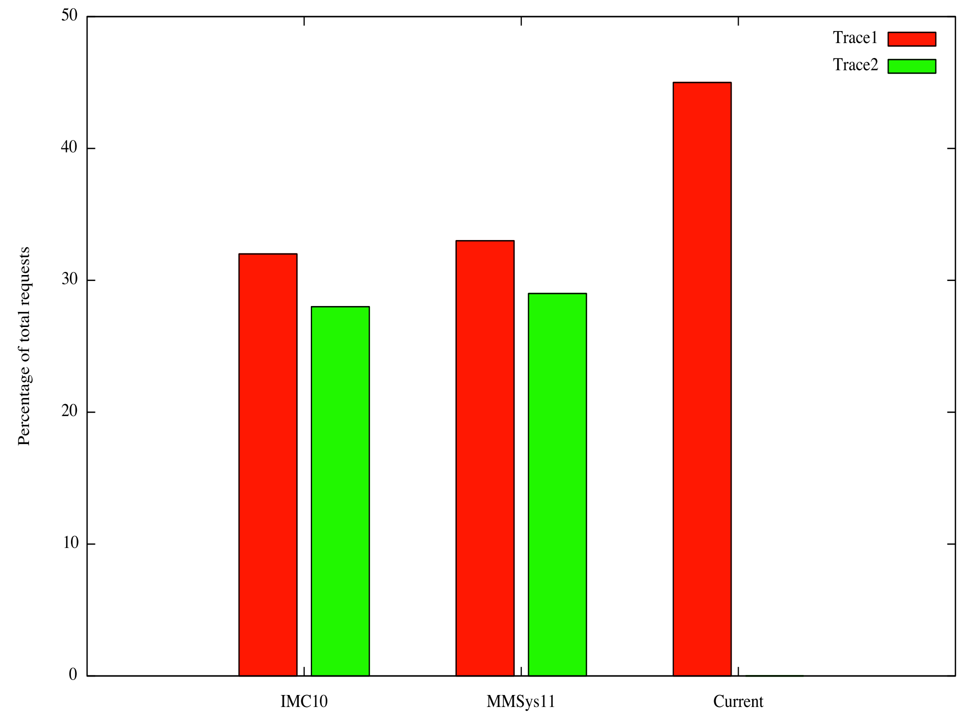
Outline

- Motivation
- Approach
- Chain Analysis
- Cache Latency
- Related List Reordering
- Discussion
- Conclusion

Motivation

- YouTube is most popular user generated video service.
- Billions of videos with unequal popularity leads 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.

Motivation (Contd.)

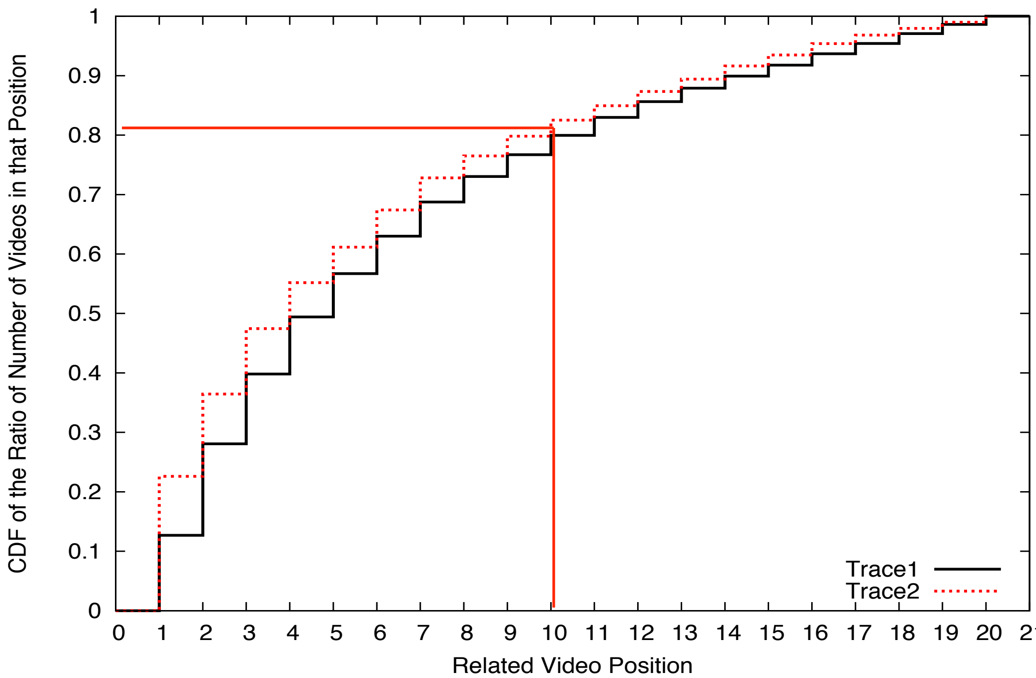


Approach

- 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.

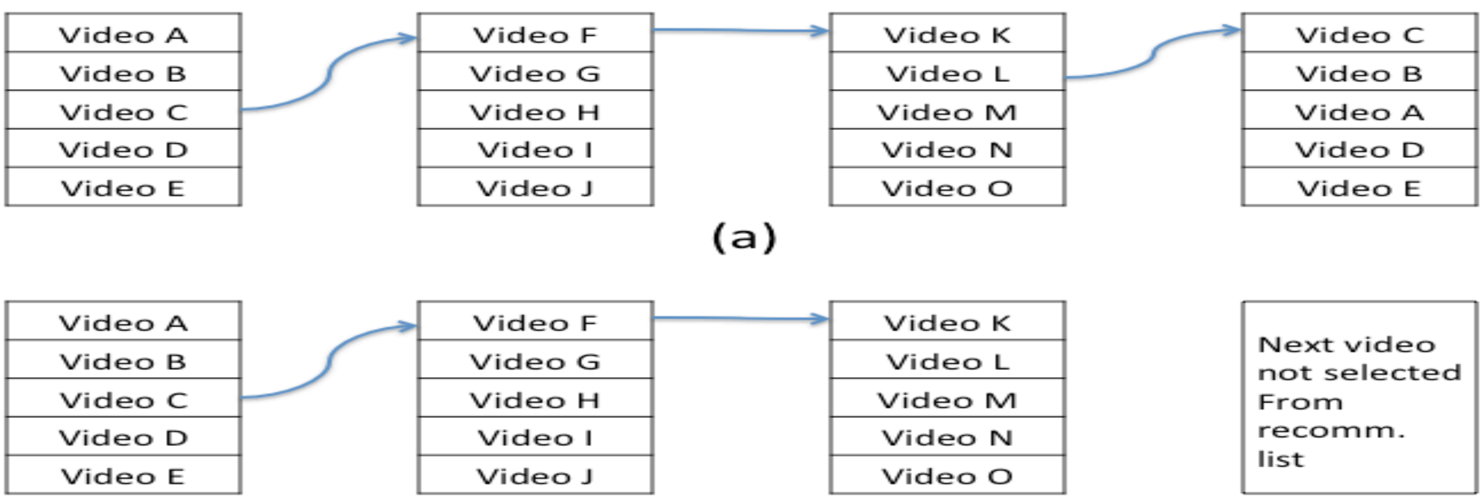
Trace Details

Trace File	T1	T2
Duration	3 Days	3 Days
Start Date	Feb 6th 2012	Jan 8th 2010
#Requests	105339	7562
#Related Videos	47986	2495



Chain Analysis

- Loop Count – Video selection ending in loop.
- Chain Count – Video selection from related list until the last video selected by other means.



(a)

(b)

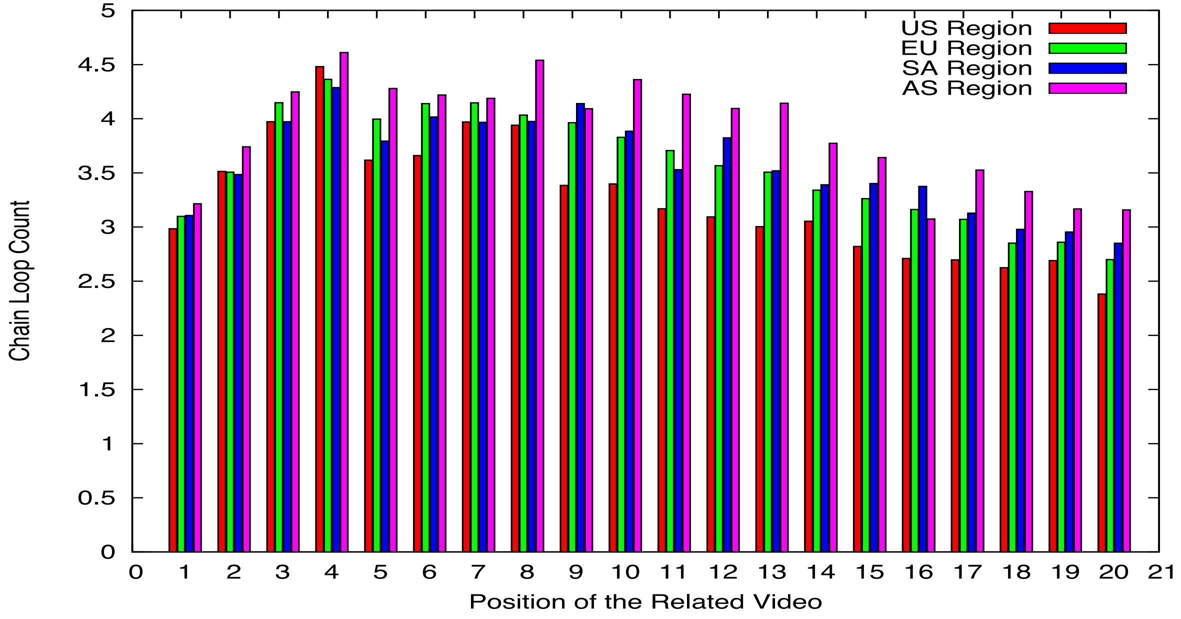
Chain Count

- Trace T1 – 84.76% chain count of 1 and 15.24% chain count of at least 2.
- Trace T2 – 48.2% chain count of 1 and 51.8% chain count of at least 2.

Chain Count	Trace T1	Trace T2
Average	1.195	2.304
Maximum	8	21

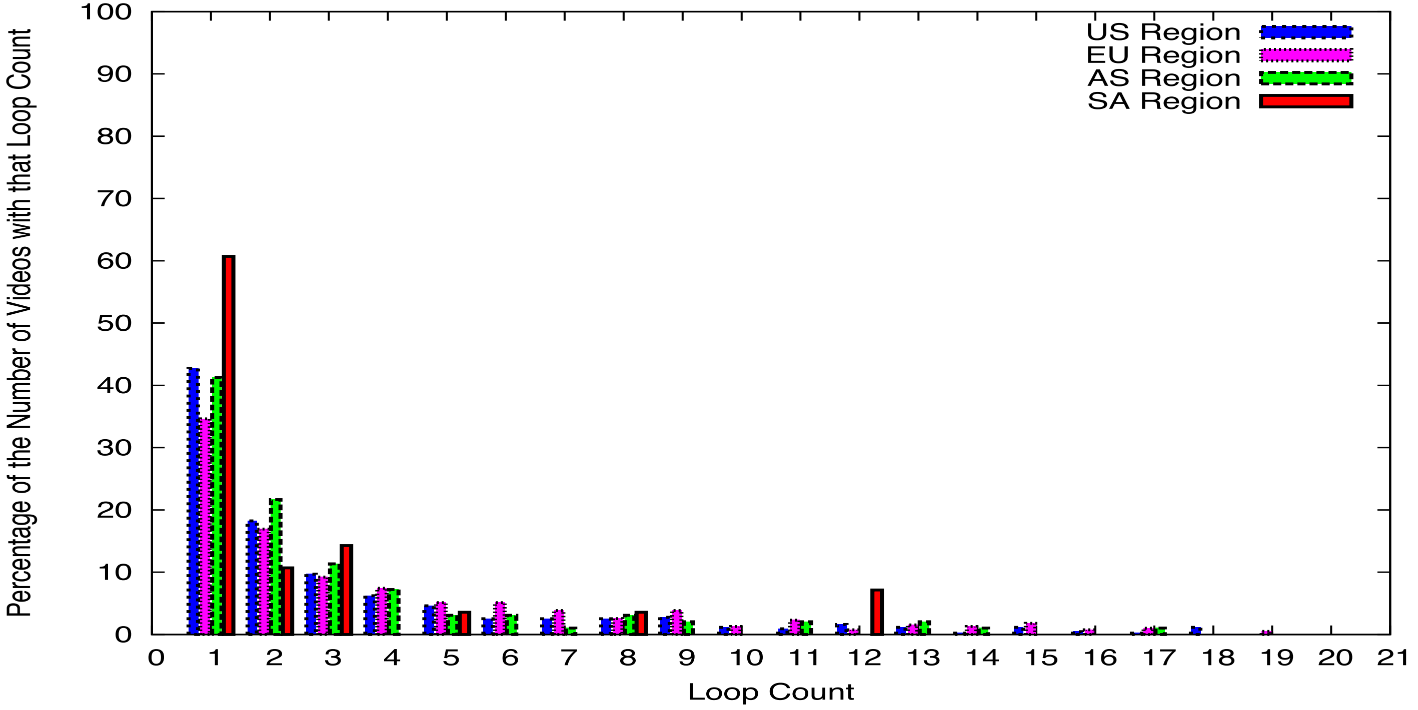
Loop Count

- Global analysis using PlanetLab.
- Loop length at fixed related video positions for 100 video requests.



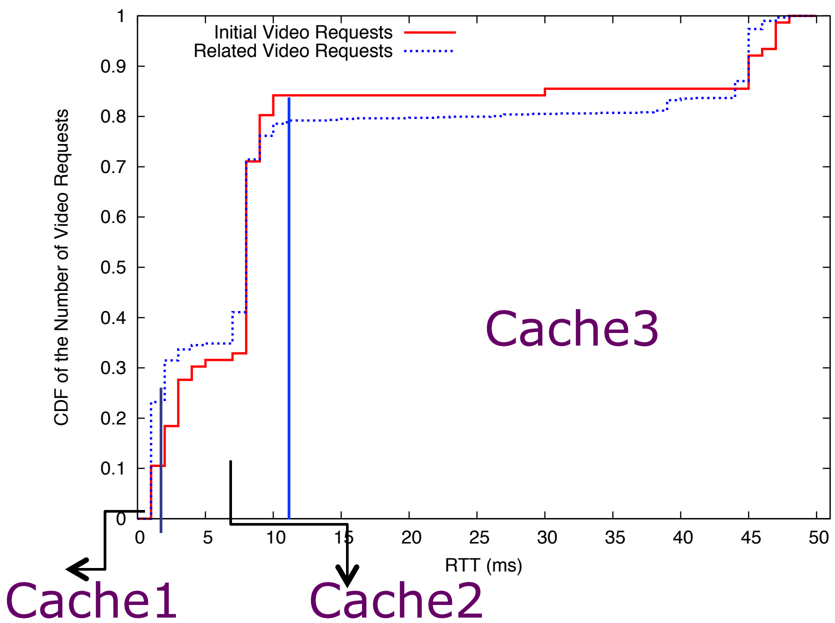
Loop Count (Contd.)

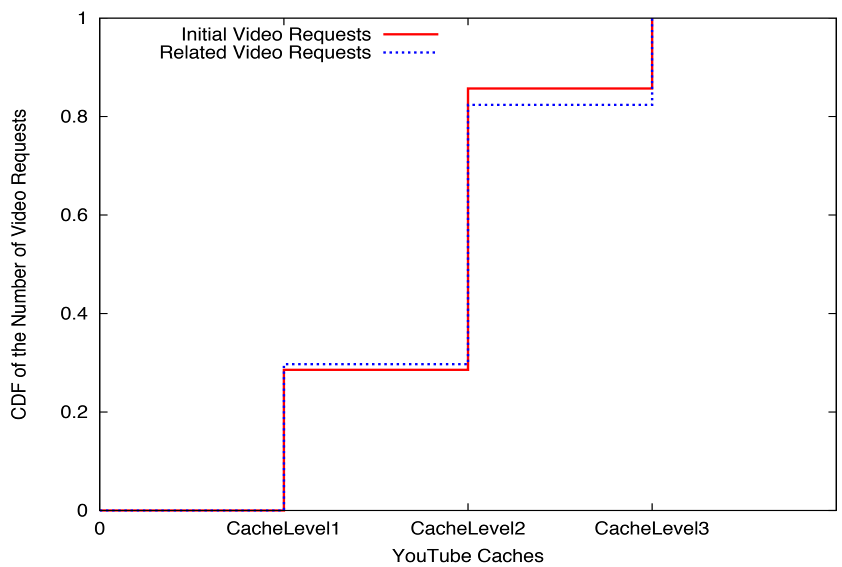
- Loop length using random selections from the related list.
- Repeated 50 times for to obtain loop length.



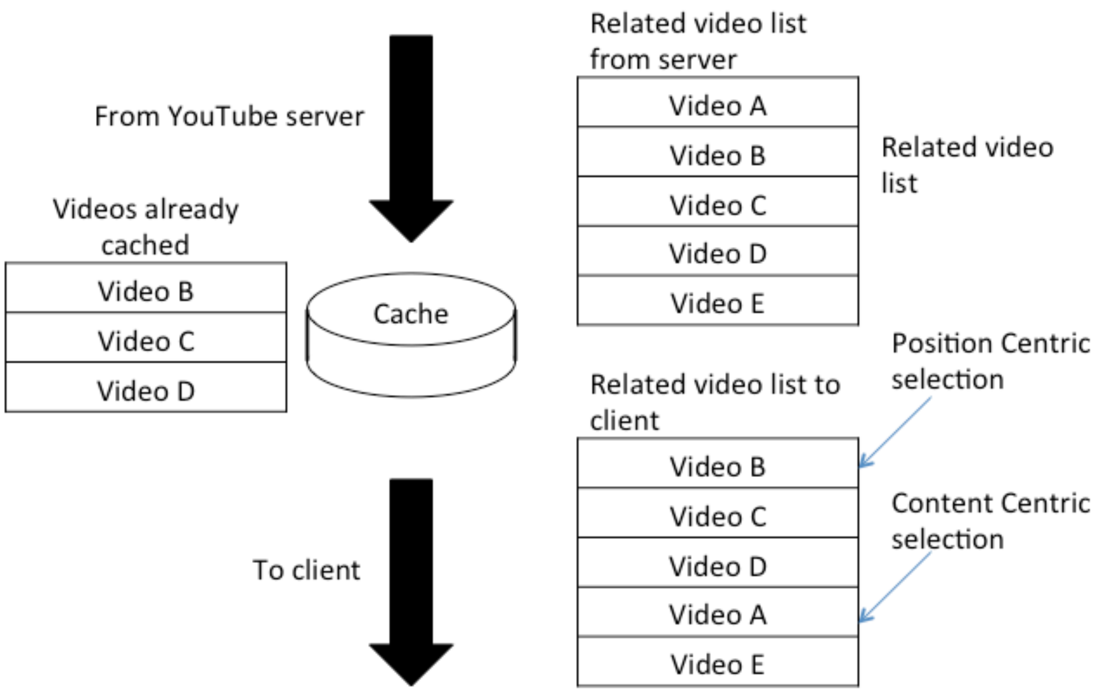
Video Origin

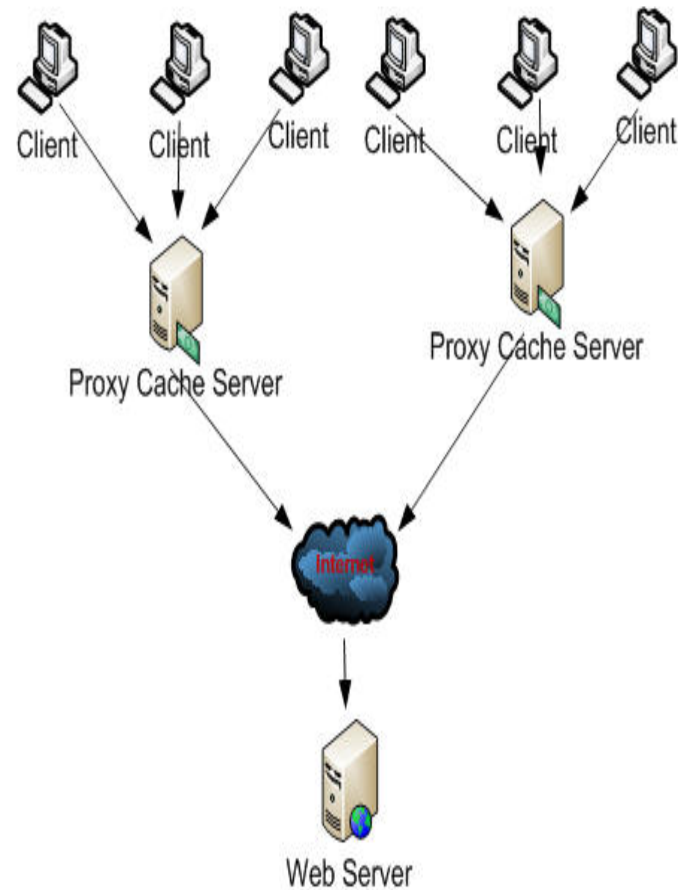
- Requested 100 videos from Trace T1 and their related videos.
- Calculated RTT for the data session in the captured trace.





Related List Reordering

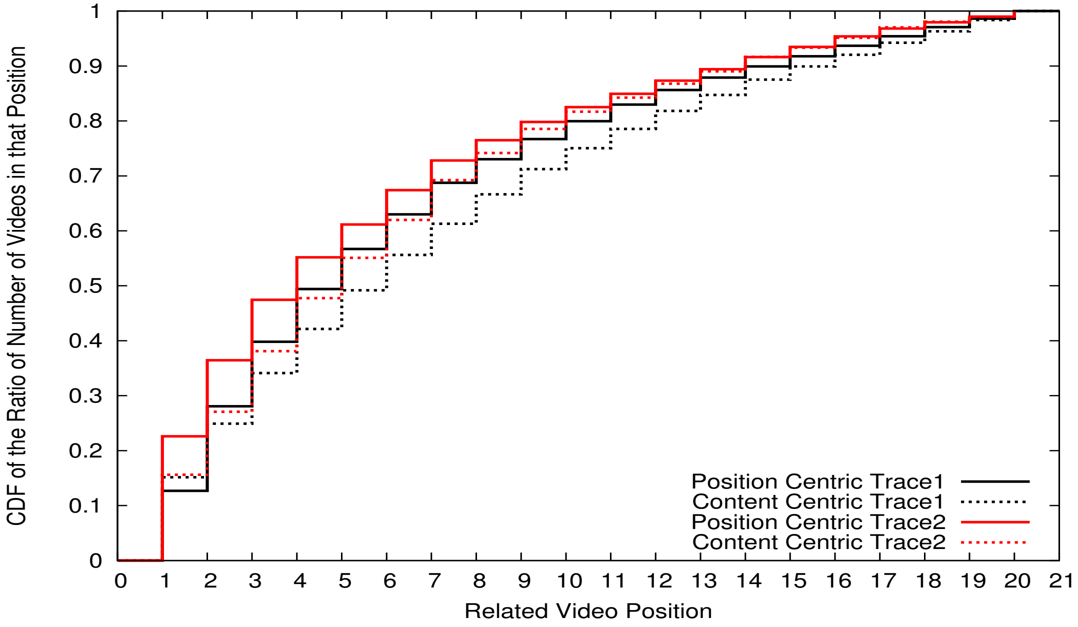




Reordering Approaches

- 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.

Reordering Results



Trace	No Reordering	Content Centric	Position Centric
T1	6.71%	6.71%	11.83%
T2	4.71%	4.71%	22.90%

Discussion

- Cost of Recommendation List Reordering.
  - Cost of cache depends on the cache structure and its size.
  - Using a plain hash table, worst case look up time will be O(n).
  - Reordering comes with little extra cost but hit rate is more substantial.

- Reduction in Server Load.
  - Trace T1 cache hit rate increase from 6.71% to 11.83%, load reduction from 93.29% to 88.17%.
  - Trace T2 hit rate increase from 4.71% to 22.9%, load reduction of 18.19%.

Discussion (Contd..)

- Popularity based sorting of related list.
  - Reordering of related list is performed without taking into consideration of the popularity of videos in the cache.
  - Only significant differences in popularity would render the approach feasible.

- Adaptive video streaming.
  - Bandwidth adaptive video streaming contains different formats of same video.
  - Each format is a different file and caching them is not considered.

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.