PP Caching: Proxy Caching Mechanism for YouTube Videos in Campus Network

Varangpa Suranuntakul and Chutimet Srinilta

***Abstract*—Nowadays, YouTube has become a very popular website offering video sharing service. YouTube accounts for a significant portion of Internet traffic in global and local networks. Many techniques have been implemented to deliver videos smoothly to every user. Streaming proxy servers are put into action to reduce user-perceived latency as well as network resource utilization. "What is viewed next" can be guessed from relationships of YouTube videos. This paper proposes PP caching mechanism that benefits from prefix caching and prefetching technique. Certain parts of selected videos are prefetched and kept at the proxy cache. The experimental results show that startup delay decreases and cache hit rate increases when the proposed mechanism is tested against actual YouTube traces from two campuses.**

***Index Terms*—prefetching mechanism, prefix caching, proxy caching, YouTube**

I. Introduction

Nature of streaming media differs significantly from that of other types of media; e.g., text and image. Streaming media are usually large in size. They require high I/O and network bandwidths in delivery process. A number of storage and retrieval techniques have been proposed specifically for streaming media applications. Such applications include distance learning, audio and video on-demand.

Social media is a relatively new form of media. They are media for social interaction where users in community generate, publish, share, consume and discuss media content. User-generated content are accessible by anyone at anytime. Unique characteristics of social media and user community must be considered in order to achieve a better service.

YouTube is a successful social media website. People post their videos, watch and give comments to videos posted by others. YouTube is a very active community. According to YouTube fact sheet, hundreds of thousands of videos are uploaded and two billion videos are view each day [1]. As for Internet traffic contribution, Alexa's statistics report that YouTube traffic ranks third among all the websites on the Internet (as of December 2, 2010) [2].

Videos on YouTube are tagged with keywords, classified into certain categories and ranked by their popularity. YouTube lists most popular videos, in each category, on its first page. YouTube provides a personalized video recommendation. YouTube also gives a list of videos related to the video being viewed.

There have been several recent studies focusing on characteristics of YouTube and other similar user-generated content systems. Video popularity life-cycle at server side was studied in [3] and [4]. Relationship between video age and its statistical properties was also explored in the study. In addition, various cache designs for user-generated content system were discussed.

YouTube traffic and user access pattern in campus network were studied in [5] and [6]. Statistical analysis was performed to characterize traffic at both YouTube and campus sides. The design of a cache for local YouTube usage was also studied in [6].

There exist three types of relationships in YouTube system: (1) relationship among YouTube videos, (2) relationship among YouTube users, and (3) relationship between YouTube videos and YouTube users. These relationships should be considered in video recommendation, video storage, and video retrieval processes.

Our work focuses on short video caching mechanism in campus network. We propose PP caching mechanism that prefetches certain videos before they are actually requested. Video relationships are taken into account in prefetching process.

The rest of the paper is organized as follows. In section II, we briefly describe streaming media caching and proxy caching mechanisms for multimedia data streams. Section III presents characteristics of YouTube videos and their metadata. YouTube traces and video popularity are discussed in section IV. Section V explains the proposed PP caching mechanism. Important parameters and performance metric used in experiment are discussed in Section VI. Section VII presents experimental results. Finally, section VIII concludes the paper.

II. Streaming Media Caching

High latency and loss rates in the Internet make it difficult to stream audio and video without a long playback startup delay. Sen *et al.* propose new caching strategy called Prefix Caching where only "prefix" parts of popular streaming media are kept in proxy cache [7]. A "prefix" part contains only initial frames of a media stream. Since the initial part of the stream is stored locally, stream playback can begin immediately upon request. The rest of the stream is then fetched from the server while the initial part is being played. Size of the prefix part depends on performance properties of

Manuscript received January 23, 2011.
Varangpa Suranuntakul is a master's degree student in School of Computer Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Chalongkrung Rd., Ladkrabang, Bangkok, Thailand, Postal code:10520, (e-mail: varangpa@gmail.com).
Asst. Prof. Dr. Chutimet Srinilta is with School of Computer Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Chalongkrung Rd., Ladkrabang, Bangkok, Thailand, Postal code:10520 Phone number +66(0) 2329 8000-99; (e-mail: chutimet@yahoo.com, kschutim@kmitl.ac.th).

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

server-to-proxy path. In addition, "work ahead smoothing" is performed using client buffer. Client then experiences a better level of Quality of Service at the end.

Wu *et al.* propose a technique called "Segment-Based Proxy Caching" [8]. A media file is divided into multiple equal size blocks. These blocks are the smallest unit of transfer. Blocks are grouped into segments of different sizes. Less popular media are partially cached while the most popular media are fully cached.

Recent research works concentrate more on data sharing. Li *et al*. propose optimal prefix caching and data sharing (OPC-DS) which combines prefix caching and interval caching [9]. Cheng *et al*. propose Peer-to-Peer sharing called NetTube [10]. NetTube includes a bi-layer overlay, an efficient indexing scheme and a prefetching strategy leveraging social networks.

Because size of proxy cache is limited, challenge lies in deciding whether to store certain data in the cache and whether cache replacement is necessary.

III. YouTube Videos

YouTube was established in early 2005 as a video sharing website. YouTube has been gaining acceptance worldwide since then.

Each video on YouTube is given a unique ID. User can share link to a certain video in email or web page in a form of hyperlink. YouTube video can also be embedded on web page and application.

Users can "tag" videos they upload with keywords or phrases that best describe video content. These tags together with viewing history are used by YouTube to recommend related videos. From the study in [10], view count of a video and view count of its related videos are comparable.

YouTube uses metadata to describe a video. Metadata contains video ID, title, description, tags, geo-location, uploader, published date, recorded on, category, duration, view count, rating, comment count and list of related videos. Metadata of a YouTube video is explained in Table I.

TABLE I
YouTube Video Metadata

Content	Description
Video ID	A URN that uniquely and permanently identifies a video
Uploader	The video owner
Published	The time that a video was created (Times are specified in UTC)
Recorded on	The date that a video was recorded
Title	A human-readable title for a video
Description	A summary or description of a video (sentence-based)
Tag	The uploader's keywords for a video
Category	The category of a video
Duration	The length of a video
Geo-location	A descriptive text about the location where the video was taken
View Count	The number of times that the video has been viewed
Rating	The rating that users are assigning to a video (liked and disliked a video)
Comment	A list of comments for a video
Related Videos	A list of videos that are related (videos having similar title, description or tags)

IV. Traces and Video Popularity

A. YouTube Traces

There are two sets of YouTube traces being used in our experiment. The first set is obtained from proxy server log at King Mongkut's Institute of Technology Ladkrabang (KMITL). The second set comes from trace repository of University of Massachusetts Amherst (UMASS) [11].

Each trace entry contains following information: client IP address, server IP address, timestamp and video identifier. We are interested only in entries that send HTTP GET to YouTube IP server (63.22.65.xx). Such entries represent requests to view YouTube videos. YouTube request may look similar to

GET http://www.youtube.com/v/dMH0bHeiRNg/...

With this particular request, a YouTube video by the ID of "dMH0bHeiRNg" is being requested by a user.

KMITL traces were collected during September 8-14, 2009. They consist of 36,803 YouTube requests, out of which 17,158 requests are for unique videos. UMASS traces were collected on March 11, 2008. They consist of 25,415 YouTube requests, out of which 20,223 requests are for unique videos.

B. Video Popularity

YouTube provides Data API [12] to allow a program to access video's metadata. We obtain video's view count via this Data API.

Number of views of a video is plotted against its rank in a log-log graph. Popularity of videos in KMITL and UMASS traces are shown in Figures 1 and 2, respectively. It can be seen from both figures that video popularity distribution follows Zipf's Law.

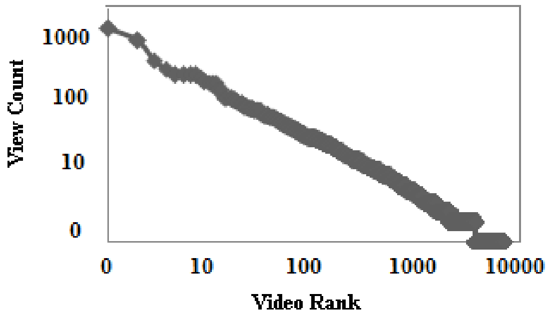

Fig. 1. Video Popularity in KMITL Traces

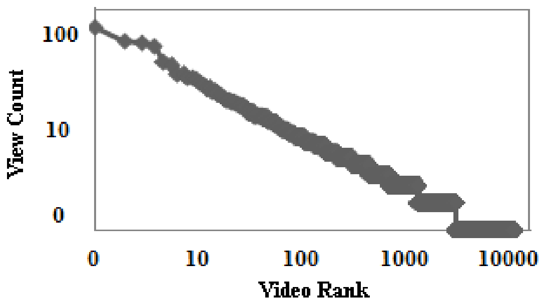

Fig. 2. Video Popularity in UMASS Traces

Zipf's Law [13] states that if objects are ranked according to the frequency of occurrence then the frequency of occurrence (F) is related to the rank of the object (R) according to the following relation:

$$F \sim R^{-\beta},$$

where $0 < \beta < 1$.

As for our data, β is 0.6 for KMITL traces and β is 0.5 for UMASS traces.

Both KMITL and UMASS traces comply with Zipf's Law which means that only a few videos are viewed at very high rates while most of the videos are viewed at low rates. Only a few videos are highly popular.

V. PP CACHING

We propose a PP caching mechanism that puts together Prefix caching and Prefetching techniques.

A. Prefix caching and YouTube videos

As discussed earlier that the beginning part of video is kept in the cache in prefix caching in order to begin playback immediately. Prefix size should depend on video quality and playback rate requirement.

YouTube accommodates videos of five different qualities. Details are shown in Table II.

TABLE II
VIDEO QUALITIES IN YOUTUBE

Quality	Resolution	Container	Playback rate (Mbps)
HD 1080P	1920 x 1080	MP4	3.5 - 5
HD 720P	1280 x 720	MP4	2
LARGE	854 x 480	FLV	0.8 - 1
MEDIUM	640 x 360	FLV	0.5
SMALL	400 x 226	FLV	0.25

Prefix size can be calculated from the following formula.

$$Prefix\ size \geq (d \times r) + w - (s \times r),$$

where d is a distance (in seconds) from YouTube server to proxy server, s is a distance (in seconds) from proxy server to client, w is a smoothing window (in Mbits) and r is a playback rate of video (in Mbps).

A high resolution video requires a larger prefix space when compare to a low resolution video. Prefix caching helps shorten startup delay in video playback.

In a normal playback session, user is watching only a certain part of a video at a given time, from the beginning to the end. Therefore, it is not necessary to store the entire video in the cache or memory. Video content newly retrieved from server can simply replace the space occupied by the content that has already been viewed. Space required for any given playback session can be determined from the prefix size equation.

B. Prefetching technique

The prefetching technique is based on the assumption that there is a higher chance that a user chooses to view a video that is related to the one being viewed over those un-related ones. If content of a video is prefetched and ready to be played, the playback can begin immediately after user request to view that video.

There exists relationship among videos. One kind of relationship occurs when video duration is longer than what YouTube limits (fifteen minutes, at present). The video must be split into many less-than-fifteen-minute pieces. Users tend to view these video pieces one after another, in the same order. The other kind of relationship is found when videos are about the same thing or similar things; e.g., Lady Gaga, Toyota Prius and funny dogs. When users are interested in a certain thing, they tend to watch more than one video about that thing.

While a user is watching a video, YouTube proposes a list of suggested videos on a side. Most people choose to view video from this list after they finish viewing the current one. Videos at the top of the list have higher chance to be viewed next.

Viewing history can indicate video relationship as well.

In summary, if videos are related, they tend to be requested consecutively. Therefore, prefetching such videos making them ready to be played upon request will certainly reduce playback startup delay.

C. PP Caching

PP caching takes the best of both worlds. In short, prefix parts of some related videos are prefetched from server. In case the prefetched part is requested later, it can be played immediately.

We define two types of space holding video data in order to facilitate the proposed caching mechanism. The first type is called *normal* space; the second type is called *prefetch* space. *Normal* space functions as a normal proxy cache space. It stores only content that has been requested. *Prefetch* space is reserved for content that are prefetched from server waiting to be requested at later time.

Figures 3-5 show how a video request is handled in three different scenarios. Figure 3 describes the scenario when the prefix part is found in *normal* space. Figure 4 describes the scenario when the prefix part is not found in *normal* space but is found in *prefetch* space. Figure 5 describes the scenario when the prefix part is not found in neither *normal* space nor *prefetch* space.

Suppose user requests to view video A and video B is related to video A.

Proxy server receives a request for video A from a user. The proxy checks to see whether prefix part of video A is stored in its *normal* space. If so (Figure 3), the proxy server sends the prefix part of video A to the user while requesting the rest of video A and prefix part of video B from YouTube server. Video A's content retrieved from YouTube is placed in *normal* space while prefix part of video B goes to *prefetch* space. Video A is delivered to user from *normal* space throughout the entire playback duration.

If video A is not found in *normal* space, the proxy checks *prefetch* space to see if video A has previously been prefetched. If so (Figure 4), the prefix part of video A is moved to *normal* space and then sent to user. The playback starts right away. Similar to the previous scenario, at the same time the prefix part is sent to user, two requests are sent to YouTube server. One request is for the rest of video A. The other request is for the prefix part of video B. From this point onward, both videos are handled in the same manner as in the previous scenario.

The last scenario happens when video A is not found in neither *normal* space nor *prefetch* space (Figure 5). In this scenario, video playback cannot begin immediately. Two requests are sent to YouTube server. One is for the entire video A, the other is for the prefix part of video of B. The rest of the process is similar to that of the other two scenarios.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

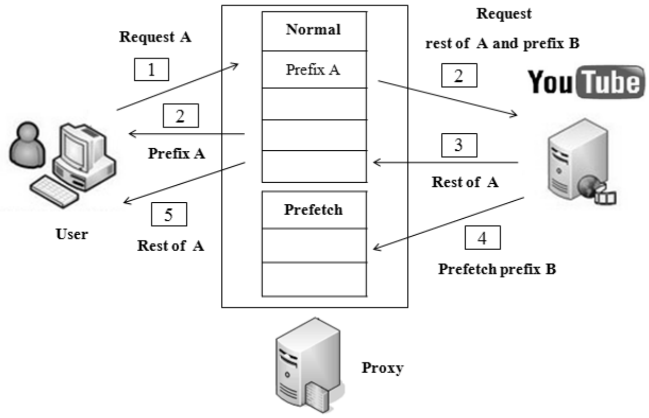

Fig. 3. Scenario 1: video A is found in *normal* space

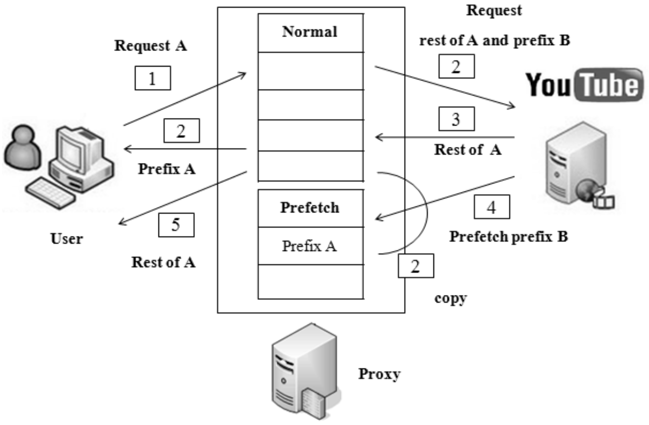

Fig. 4. Scenario 2: video A is found in *prefetch* space

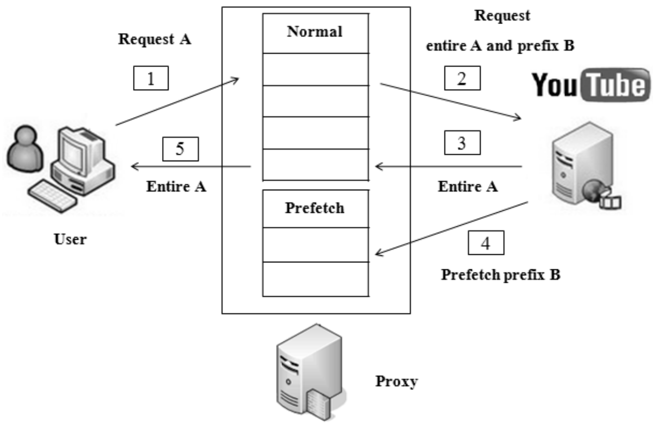

Fig. 5. Scenario 3: video A is not found at proxy server

Parameters to be considered in the proposed cache implementation are as follows:

- Size of *normal* space (S_n)
  Number of videos that can be stored in *normal* cache space is proportional to S_n.
- Size of *prefetch* space (S_p).
  Number of videos that can be prefetched from server is proportional to S_p.
- Consecutiveness threshold (C)
  This parameter is used to decide whether two requests are consecutive requests.
  If the difference in time when any two related videos are requested is less than C, those two videos are said to be requested consecutively. They will be treated as two separate requests, otherwise.
- Consecutiveness count ($CC_{(A,B)}$)
  $CC_{(A,B)}$ represents number of times when a pair of videos (video A and video B) are requested consecutively. Value of $CC_{(A,B)}$ is added by 1 every time video A and video B are requested within C time units apart.
- Prefetching threshold (P)
  This parameter determines whether or not to prefetch a video.
  If video A is requested by a user and $CC_{(A,B)}$ is greater than P, video B will be prefetched to the *prefetch* space.

Note that video A and video B satisfy both thresholds explained right above in all scenarios described in Figures 3 – 5.

VI. Experiment

Traces used in our experiment are actual YouTube traces collected from KMTIL and UMASS proxy server as explained in Section IV.

Performance matrices:

1. Hit rate difference (*diff*) is the difference of proxy cache hit rates when PP caching is in action and when it is not.

$$diff = \frac{HitRate_{PP} - HitRate_{noPP}}{HitRate_{noPP}} \times 100,$$

$$HitRate = \frac{Req_{np}}{Req_{total}} \times 100,$$

where Req_{np} is number of user requests satisfied from *normal* and *prefetch* spaces and Req_{total} is total number of user requests.

2. Usability of prefetched videos (in percentage). This metric indicates how well prefetching technique is.

$$usability = \frac{Req_p}{V_p} \times 100,$$

where Req_p is number of user requests satisfied from *prefetch* space and V_p is total number videos prefetched from YouTube.

Parameters:

C is set to 15 minutes which is the limit of video length at YouTube, S_p is equal to 30.

P is varied from 0 to 25 for KMITL traces and 0 to10 for UMASS traces. S_n is set to 50, 100 and 150.

Cache Replacement Policy: Least Recently Used (LRU)

VII. Results

Hit rate differences in KMITL and UMASS traces are plotted against prefetching thresholds in Figures 6 and 7, respectively. Hit rate difference drops when prefetching threshold increases at all values of S_n in both sets of traces. Large S_n results in lower hit rate difference at any given P. The hit rate difference drops quite fast at small P and high S_n. The hit rate difference tends to be more stable as P increases.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

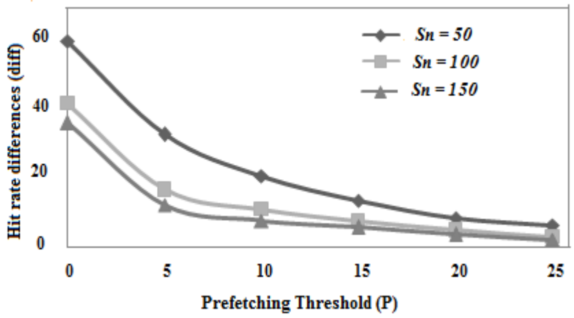

Fig. 6. Hit rate differences (KMITL traces)

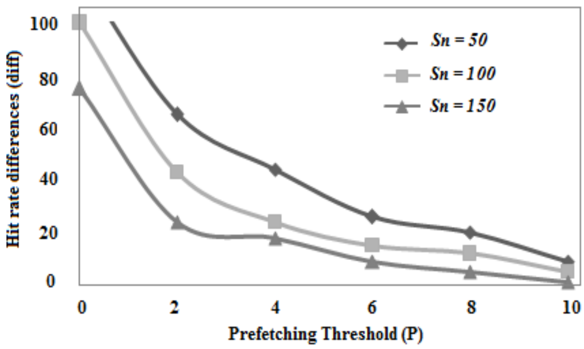

Fig. 7. Hit rate differences (UMASS traces)

Usability of prefetched videos is shown in Figures 8 and 9. Usability increases when ***P*** increases. It is approaching 100% at the end of the curve which means that almost all prefetched videos satisfy user requests. Thus, PP caching can help shorten playback startup delay.

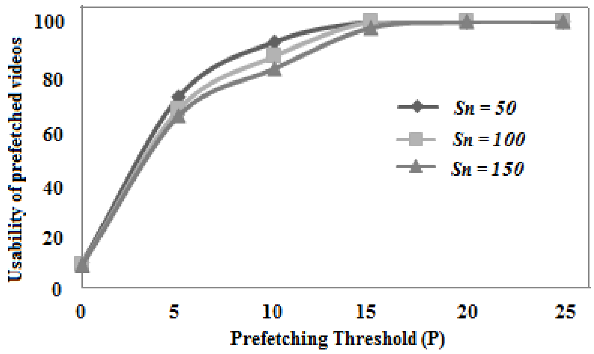

Fig. 8. Usability of prefetched videos (KMITL traces)

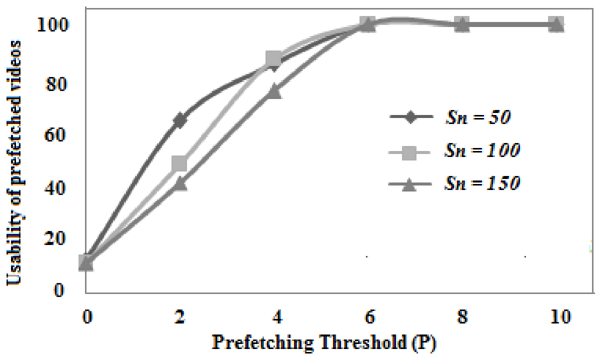

Fig. 9. Usability of prefetched videos (UMASS traces)

VIII. CONCLUSION

This paper is about proxy caching mechanism for YouTube videos in campus network. We propose PP caching mechanism that augments prefetching feature to prefix caching. PP caching uses relationship between videos in its prefetching process. Results when run against actual YouTube traces collected from KMITL and UMASS show that PP caching can improve cache hit rate and reduce startup delay.

REFERENCES

[1] "YouTube Fact Sheet", http://www.youtube.com/t/fact_sheet.
[2] "Alexa", http://www.alexa.com.
[3] M.Cha, H.Kwak, P. Rodriguez, Y-Y. Ahn, and S. Moon, "I Tube, YouTube, Everyday Tubes: Analyzing the World's Largest User Generated Content Video System," in *ACM Internet Measurement* Conference, October 2007
[4] X. Cheng, C. Dale and J. Liu. " Statistics and Social Network of YouTube Videos", In *Proceedings of the 16th International Workshop on Quality of Service*, 2008
[5] P. Gill M. Arlitt, Z. Li and A. Mahanti, "YouTube Traffic Characterization: A View from the Edge," in *IMC '07*: *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement***,** 2007.
[6] M. Zink, K. Suh, Y. Gu, and J. Kurose, "Watch Global, Cache Local: YouTube Network Traffic at a Campus Network: measurements and implications", In *Proc. SPIE (2008)*, vol. 6818.
[7] S. Sen, J. Rexford, and D. Towsley, "Proxy prefix caching for multimedia streams," In *Proc. IEEE INFOCOM*, April 1999.
[8] K. Wu, P.S. Yu, J.L. Wolf, "Segment-based proxy caching of multimedia streams", in: *Proceedings of the 10th International WWW Conference*, May 2001
[9] K.Li, C. Xu, Y. Zhang and Z. Wu, " Optimal Prefix Caching and Data Sharing Strategy", in *Multimedia and Expo, 2008 IEEE International Conference* , June 23 2008-April 26 2008
[10] X. Cheng and J. Liu, "NetTube: Exploring Social Networks for Peer-to-Peer Short Video Sharing" In Proceedings of *INFOCOM* 2009 (April 2009).
[11] "UMASS Trace Repository: YouTube Traces from the Campus Network", http://traces.cs.umass.edu/index.php/Network.
[12] Reference Guide, http://code.google.com/apis/youtube/1.0/reference.html.
[13] "Zipf's Law (Wikipedia)", http://en.wikipedia.org/wiki/Zipf%27s_law.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)